State-of-the-art of related technologies to Alfanet

Project Deliverable Report

Deliverable D12 – State-of-the-art

Workpackage	WP1. User Requirements & Specifications			
Task	T12. State-of-the-art of related technologies			
Date of delivery	**Contractual**	31 07 2002	**Actual**	25 10 2002
Code name	ALFANET_D12_v1.doc		**Version** 1.0	draft ☐ final ☑
Type of deliverable	Report			
Security (distribution level)	Public			
Contributors	OUNL, UNED, SAGE			
Authors (Partner)	Marcel de Croock, Frans Mofers, Maarten van Veen, Peter van Rosmalen, Francis Brouns (OUNL); Jesús G. Boticario, Carmen Barrera, Olga Santos, Antonio Ayala, Elena Gaudioso, Félix Hernández (UNED); Cristina Arana, Irma Trueba, Carlos Fuentes, Mª del Mar Rodrigo (SAGE)			
Contact Person	Marcel de Croock Open University of the Netherlands - OTEC Valkenburgerweg 177 6401 DL. Heerlen		Tel: + 31-45 5762735 Fax: + 31-45 5762802 Email: marcel.deCroock@ou.nl	
WP/Task responsible	OUNL			
EC Project Officer	Elena Coello			
Abstract (for dissemination)	This report describes the results of the study into the State-of-the-art of technologies related to the Alfanet project. The aim of the study was to identify existing systems that could be used as components of the Alfanet system, technologies that could be used for the design of the system and peripheral systems and/or standards that the system should interface with or adhere to. 14 topics were studied. Each topic is described in detail in an appendix to the document. A summary of the results of each topic and conclusions for the development of the Alfanet system are included in the document.			
Keywords List	State-of-the-art; E-learning platforms, Learning management systems; Computer supported collaborative learning; Knowledge management tools; Adaptive educational systems; Instructional design models; Web mining; User modelling; Machine learning; Intelligent agents; Multi agent architectures; Web access and services; EML; XML; Security concepts; Educational ontologies.			

Alfanet Project Coordination at: Software AG España, S.A. Ronda de la Luna, 4
Tel: +34 91 8079411 - fax: +34 91 8079447 - email: carana@softwareag.es Tres Cantos, E-28760 Madrid

Executive Summary

Introduction

This report describes the results of the study into the State-of-the-art of technologies related to the Alfanet project. The aim of the study was to identify existing systems that could be used as components of the Alfanet system, technologies that could be incorporated in the design of the system and peripheral systems and/or standards that the system should interface with or comply to.

Chapter 1 first provides a brief introduction to the Alfanet project and describes the scope of the project.

Chapter 2 provides an overview of the preliminary architecture for the Alfanet e-learning system and allocates the technologies which will be studied.

Chapter 3 summarizes the outcomes of the evaluations of the different technologies.

Chapter 4 draws some preliminary conclusions for the further development of the Alfanet system.

Attached to this document are 14 appendices in which each technology is described in more detail.

Description of conclusions/results

The study resulted in the identification of existing systems that could be used as components for the Alfanet system, several technologies that could be incorporated in the design of the system and peripheral systems and/or standards that the system might have to interact with. No definite conclusions were formulated but several principles were stated that served as starting points for discussions and decisions about the design of the architecture for the Alfanet system. The most important principles that were stated are:

1) The Alfanet system should incorporate the existing E-learning platforms: aLF and Edubox, and facilitate integration with existing knowledge management tools.
2) The system should adhere to standards for privacy and security, XML and related standards, and the EML standard for describing the structure and process of learning. In addition, several principles were stated about requirements and approaches for designing the architecture of the system.

Table of Contents

1. Introduction

1.1 Situation

The present document describes the results of activity T12: State–of-the-art-technologies, the purpose of which was to study and describe technological and educational developments that could be relevant for reaching the objectives of the Alfanet project. The main goal of the Alfanet project is to build an e-learning system that will take advantage of the new internet related techniques, human interaction, and machine learning, to allow:

a) organisation personnel to have interactive, adaptive and personalised e-learning experiences bringing them the opportunity to learn and experiment on matters that are relevant for their work.
b) organisations to control and efficiently manage intellectual capital, promoting the evolution of employees in specialised & multidisciplinary areas for their work.

In addition the project aims to contribute to the educational standardisation efforts for adaptive education and to define a business model for e-learning.

1.2 Purpose

The purpose of the document is to provide an overview of technologies and approaches that are relevant for the development of the Alfanet system. At the start of the project the following main requirements, constraints and basic assumptions were set out for the Alfanet system:

- The system will present effective and adaptive instruction to learners in a web-based learning environment.
- The system will allow learning content providers and educational centres to provide learning contents in such a way that these contents can be adapted to the personal needs.
- The system will allow collaboration between users (learners, teachers). It will facilitate the development of virtual communities which can manage workgroups on the web.
- The system will be secure, meaning that all personal information of users of the system will remain confidential, safeguards will be installed to avoid unauthorised access and virus infections to secure the integrity of the system and measures will be taken to secure the copyrights of the learning materials.

Technically the solution is anticipated as:

- The system will be based on a multi-agent architecture.
- Adaptation of instruction to learners will be inferred from user models that are acquired from available learner data and the learner's interaction with the system. The models are build by applying a set of machine learning techniques.
- The system will be based on advanced pedagogical models (active learning, collaborative learning, …).
- The system will build upon and contribute to existing standards for describing and publishing learning materials, including standards for adaptive instruction.

In the light of these requirements the main objectives for task T12 were to:

1. Identify existing systems that could be readily used or adapted to fulfil the basic functional requirements of the Alfanet system. In order to avoid reinventing the wheel and having to develop system functionalities from scratch, already available systems on the market that could possibly be incorporated into the Alfanet system are identified and described.
2. Identify technologies and approaches that could form the basis for the design of the Alfanet system in order to fulfil the main functional requirements for the system.

3. Identify peripheral systems and/or standards that the Alfanet system should adhere to or should be able to interface or integrate with.

This document is intended to be read by designers and developers of the Alfanet system. The results of achieving objective 3 provides input to Deliverable 1.3 “Functional specifications”. Results of achieving objectives 1 and 2 can be as input for further designing (Work package 2) the Alfanet system and/or to implement the final requirements for the Alfanet system (Work packages 3 and 4).

1.3 Overview

In order to achieve the objectives for Task T12 as stated in the previous section all project participants identified possible relevant topics for study. In several sessions these topics were sorted, ordered, combined and/or split up until a list of topics evolved that could be expected to provide all necessary relevant information. Per objective the list is as follows:

To identify usable components for the Alfanet system, a review of existing learning systems in the open market was made. The following types of systems were studied:

- E-learning platforms
- CSCL environments and approaches

To identify relevant technologies and approaches, the latest developments in the following domains were studied:

- Adaptive educational systems and approaches
- Advanced instructional design models
- Web mining
- User modelling
- Educational ontologies
- Machine learning
- Intelligent agents and multi-agent architectures
- Web access and services

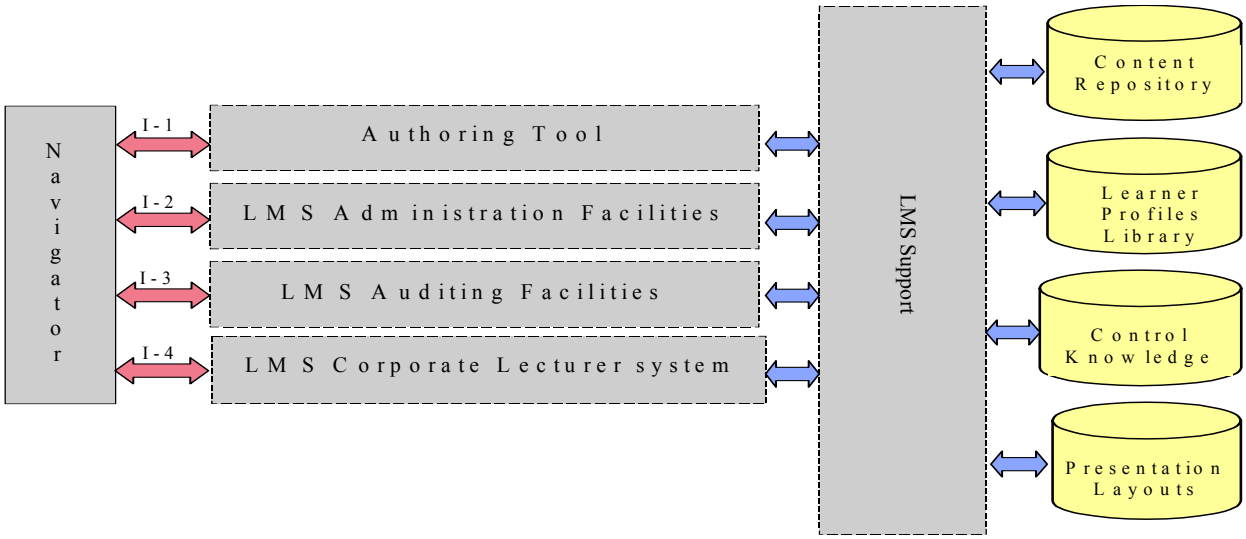
To identify relevant peripheral systems and/or standards, the following domains were studied:

- EML – Learning standard
- XML and related technologies
- Security Concepts
- Knowledge management tools

In the next section we will provide an overview of the Alfanet e-learning system and allocate the technologies which will be studied. In the third section the outcomes of the evaluations of the different technologies are summarized. Finally in the fourth section preliminary conclusions are drawn for the development of the Alfanet system.

2. Overview

The following figure provides a possible overview of the basic components for a learning management system (LMS).



All interaction (I-1 to I-4) with the LMS takes places via a web-browser or navigator. There are principally four different goals that different users can have to interact with the system. These goals and the different user roles are:

I-1. Content Management: an **author** or professor defines or refines contents by means of the Author Tool. The contents are stored in the Content Repository.

I-2. Learning Administration Management: an **administrator** manages the learning trajectories of each individual learner or groups of learners. Functions that can be performed are for example the enrolment of learners, the assignment of learners to courses, authorising levels of information access to different learners, monitoring of learners progress, etc.

I-3. Competence and Knowledge Management: **Tutors and managers** assess and report on learners' competencies in order to maintain an overview on a companies' human resources and maintain a level of services the company as whole has to be able to provide to customers.

I-4. Learning: **Learners** interact with the system and through the system with other learners or professors in order to acquires new knowledge or learn new skills. The system adapts the contents, the presentation and the available services to the learners' needs and preferences, and advices the learner on additional information sources and communication channels.

The next sections describe the main components of the LMS architecture and how they relate to the topics that are studied and described in this document.

2.1 Authoring Tool

The authoring tool supports the author in the definition of *Content Material*: the main input of the *LMS Corporate Lecturer System*. With contents the actual texts, graphics and assignments that are presented to the learner are meant. These contents are specified according to a certain methodology that describes when, how and in what format the contents should be presented. The contents that an author adds are

stored in the content repository. The tool also provides facilities for importing and exporting content and for connecting to external content sources.

Several topics of the study are relevant for the development of this tool. First, the topics E-learning platforms and CSCL Environments and Approaches are relevant for identifying existing market authoring tools (for editing contents, image, video composition) that could be incorporated in the system. Second, the topics Adaptive educational systems and approaches and Advanced instructional design models are relevant. The Alfanet system will be able to present and adapt the instruction that is provided to the learner according to different pedagogical models and adaptive approaches. Therefore it must be ensured that the authoring tool will allow an author to enter the contents in a format that is consistent with the chosen models and approaches. Third and finally the topic EML – Learning standard is relevant in order to investigate how the contents can be entered, imported and exported consistent with the EML standard.

2.2 LMS Administration Facilities

The Administration Facilities component provides functionality for the general management of assigning learners to courses and facilities. It covers the entire life-cycle of a course's administration from registration and follow-up to termination. Some facilities are: user inscription, course registration, assignment of users to courses, authorisation of access to certain contents or special private information, workgroups configuration, and in general all types of events that can occur in a LMS.

The most important topic of the study that is relevant for the development of this tool is: E-learning platforms. Administrative functions are the most important (and often only) features of most of the many LMS's. An overview of E-learning platforms in the existing market is given and those that could be incorporated into the Alfanet system will be identified. In the design of the Administration Facilities component functionalities will have to be included to ensure learner privacy and to manage copyrights of third party instructional materials. In the topic: Security Concepts, approaches for meeting these requirements are studied.

2.3 LMS Auditing Facilities

The purpose of the LMS Auditing Facilities component is to gather data about the effectiveness of the tool. The following aspects are taken into account: 1) assessment of changes in learners' knowledge and skill development, 2) assessment of the efficiency, effectiveness and employment of (parts of) the courses, and 3) identification of unusual or undesired events. Examples of the last aspects can be for example: critical situations in that learners do not or too slowly reach a required level of performance, or situations in which collaborative group processes are ineffective or even detrimental to the learning of the group members.

The chosen approach for gathering these data is to analyse the traces of the learning process that learners leave while interacting with the system. That is, all information about interactions of learners with the system (e.g., access of information pages, performance on assignments, interactions with other learners, etc.) is stored and analysed by the Auditing Facilities component. Results of the studies into topics: Knowledge management tools, Web mining, are expected to provide information that is of relevance for the design of this component.

2.4 LMS Corporate Lecturing System

The LMS Corporate Lecturing System is the central component of the Alfanet system. This component takes care of managing the interaction between the learners and the Alfanet system. It's key function is to dynamically adapt the instructional interactions to a learners' needs, wishes, learning performances and/or characteristics. In order to do this the component decides on: 1) what instructional contents are to be delivered to a learner, 2) in what format these contents are to be delivered, 3) what advice are to be given to a learner (e.g., to use certain services, to join certain workgroups, to contact other users, to use certain tools, to investigate sources of information, etc.).

The main technique that will be used to design the functionality for this component is a dynamic multi-agent architecture in combination with different user modelling techniques. Agents are specialised, independent software components that can collect information about the interaction of a user with a system and based on that information build a model about certain aspects or characteristics of that user. Different agents are used to build different models of different aspects of a users interaction with the Alfanet system. Heterogeneous agents will be developed to combine all the information from the separate agents in order to adapt and personalise the instructional interventions of the system with the learners. The topic: Intelligent agents and

multi-agent architectures, further analyses these techniques and will provide information to further develop the Corporate Lecturing System component. Studying the topic: User Modelling, will provide more information of how the agents can build models about the different users aspects. The topic: Web Mining, further investigates approaches for analysing the data that is gathered while the user interacts with the system in order to provide information that allows the creation of user models. Studying the topics: Adaptive Educational Systems and Approaches, Advanced Instructional Design Models, CSCL Environment and Approaches, Web Access and Services Personalisation and Machine learning will provide information about how on the basis of user models the instruction can be adapted to and personalized for a learner.

2.5 LMS support

The LMS support component manages four knowledge repositories that are used by the rest of the system. The Content Repository (1) is the main repository. It includes all the components of the course to be lectured by the Alfanet system. The repository is fed by the Authoring Tool and used by the LMS Corporate Lecturer component and it can contain multiple courses. The Learner Profiles Library (2) contains the characteristics of the learners represented in a model or profile. The contents of these models are continuously updated as the learners interact with the Alfanet system. The information in the model is used by the LMS Corporate Lecturer component to adapt the instruction to individual learners. The Control Knowledge Repository (3) contains information about the interactions of the user with the system during the training; it keeps all the user's key actions and all the user session information. It is used to update the Learner Profiles Library and also provides information to allow adaptation of the instruction to individual learners. Finally, the Presentation Layouts Repository (4) includes a set of templates or layouts that can be customised to different user profiles and that will provide a different presentation style adapted to users preferences.

The structure of the Content Repository is strongly related to standardisation efforts in the field of e-learning. The topic: EML – Learning standard will provide information for structuring this repository. In the topic: User modelling, current techniques to build user models will be studied to provide information for structuring the Learner Profiles Library. The topics: Knowledge Management, Advanced instructional design models and Adaptive educational systems and approaches will provide information to assess what types of presentation layouts should be stored in the Presentation Layouts Repository. The topics: XML and related technologies and Security Concepts will provide more general information about techniques for structuring and maintaining the repositories and for making sure that the information is securely stored in the repositories.

3. Outcomes of the evaluation of the related technologies

In the following sections the outcomes of the evaluations of the different technologies are summarised.

3.1 appendix 1 E-learning platforms

3.1.1 Overview

This section describes the functionalities of e-learning platforms on the market. E-learning platforms or learning management systems (LMS) are tools that offer possibilities to (re-)organize learning, teaching and education. They allow for flexibility in delivery, but also in learning, in time and place. They should also make it easier to differentiate didactical models and scenarios depending on users' preferences. There are many definitions of e-learning platforms and thus many functionalities incorporated and covered by the systems. In the appendix several definitions are described and expanded on. The platforms can have many features, including distribution of content, tutoring, assessment, communication and administrative tasks. In the appendix differences between tools are indicated and the most common functionalities are described and discussed. In the context of Alfanet only those platforms are relevant that consist of at least content delivery and tutoring facilities via internet technologies, combined with communication facilities and organizational and administrative functions.

3.1.2 Conclusions for Alfanet

The usability of the described system depends on users' expectations and needs. As most universities and higher education institutes are only starting in the field, most current systems seem to be ‘value for money’ for the specific area they market. Many of the systems incorporate the base functionalities and features needed for the Alfanet LMS. None of the systems, except Edubox, model the didactics as well as the content. However, none of the systems meets all criteria for Alfanet. Only the more complex e-learning systems might be a starting point for the Alfanet LMS, but they would require extensive customisation. Customisation is always risky. It can be very time consuming and expensive. It also makes the product very dependent on specific versions of applications. When the Alfanet LMS is going to build on the basis of an existing system, it might be wise to choose a system of which one or more of the partners of Alfanet has extensive knowledge and in-house capabilities.

3.2 appendix 2 CSCL environments and approaches

3.2.1 Overview

Current e-learning platforms support collaborative learning tasks, but they are not CSCL environments. Pure CSCL environments exist mainly in research scenarios and are applied to a particular domain, where collaboration dialogues are completely structured to a specific audience. E-learning platforms promote collaboration by providing collaborative tools and a more open interaction to their users.

However, the CSCL field can be of help in specifying objectives and tasks regarding collaboration in broader frameworks like web environments for e-learning. To clarify objectives and tasks a general framework for designing CSCL environments, developed by OUNL, is introduced.

In this appendix the main features of CSCL systems and some systems already implemented in research scenarios are described. To complement this view, e-learning systems that support collaboration are also mentioned, such as BSCW and FLE3. In particular, a platform called ALF (Active Learning Framework), which being developed at UNED, is described in more detail. This platform provides facilities for collaborative learning and is related to issues that are the focus of Alfanet project, such as Adaptation and Multiagent Systems.

3.2.2 Conclusions for Alfanet

CSCL is a vast field of research. Moreover, it can be used for other purposes than supporting collaboration, such as for transmitting and delivering knowledge.

The aim of the Alfanet Project is not to develop a CSCL platform, but to build a system that provides some features related to CSCL. Therefore, ideas from CSCL can be taken into account when designing the system for Alfanet. Whereas CSCL environments have a very ambitious approach regarding collaborative issues, the Alfanet project has a broad approach which is intended to provide an interactive, adaptive and personalised e-learning framework.

In this appendix we have mentioned some key points to be considered in the Alfanet design.

- To include synchronous and asynchronous tools to support communication, and shared workspaces to promote collaboration
- To include some method to structure the communication
- To collect learner interaction data and to create a model of interaction
- To provide collaborators with indicators that represent the state of the interaction
- To advise collaborators by recommending actions to improve their interaction

CSCL research field can be of help in specifying objectives and tasks regarding collaboration in broader frameworks, like Web environments for e-learning. This technology is intended to solve the challenges of collaborative scenarios, but the Web environment adds complexity to a traditional CSCL system.

In this sense, it can be very useful to consider OUNL methodology for designing CSCL environments. This framework provides a model to define and describe the CSCL key elements that Alfanet should support. It can be used as a starting-point for developing functionality for the Alfanet system, which is intended to support a dynamic and adaptive collaboration framework. Therefore, it is desirable to start the development of Alfanet taking as a basis an existing system that provides collaboration facilities.

3.3 appendix 3 Adaptive educational systems and approaches

3.3.1 Overview

The purpose of educational systems in general is to provide learners with information and practice opportunities to help them to reach certain learning goals, that is, to help learners to learn certain skills and/or to increase their knowledge about a certain topic. Different learners might have different needs, characteristics, prior knowledge, etc., which might require the presentation of different information to different learners and/or in a different format. It is the purpose of adaptive educational systems to take these aspects of individual learners into account when presenting information and/or practise opportunities in order to make the learning process as effective, efficient, and motivating as possible. In this appendix a brief historical overview is given of the research that eventually led to the development of adaptive educational systems. Three examples of state of the art solutions for making educational systems adaptive are described.

3.3.2 Conclusions for Alfanet

From the viewpoint of developing Alfanet the most important requirements for an adaptive educational approach are that it will allow for an adaptive presentation of instructional materials 1) that is effective, 2) that is domain independent, 3) that allows for collaborative learning, and 4) that is web-based. Regarding the first point no information is available about the effectiveness of adaptive educational systems compared to traditional non-adaptive systems. Regarding point two it can be concluded that the approaches of all systems are generally applicable in different domains. However, for TASKi and ELM ART II courses will have to be specifically developed in great detail. Only WEB-DL has functionality that allows for the adaptation of contents that are not part of a purposely, pre-specified instructional design. With regard to point three, none of the systems disallows collaborative learning. WEB-DL is the only system that specifically addresses collaborative learning by providing advice on the use of collaborative services. With respect to point four, ELM ART II and WEB-DL are purposely developed for usage on the internet. Whether the TASKi module can be used on the internet, depends on the implementation of the learning environment that TASKi controls. No clear conclusions can be made for the use of the described approaches in Alfanet. None of the approaches appears to have characteristics that would make it problematic to include it in Alfanet. Each of the approaches have their specific advantages and disadvantages.

3.4 appendix 4 Advanced instructional design models

3.4.1 Overview

The main goal of any instructional design process is to construct a learning environment in order to provide learners with the conditions that support desired learning processes. In this appendix four ID-models are described that focus on the analysis of a to-be-trained skill in a process of job and task analysis and the conversion into a training strategy, or the design of a learning environment (often taking the form of some kind of blueprint) that is ready for production.

The models are compared against Merrill's (2002) *First Principles of Instruction*, which are design oriented, fundamental instructional principles that apply under all conditions of learning but can be implemented by a wide variety of programs and practices. Learning from a given instructional program will be facilitated in direct proportion to the explicit implementation of first principles of instruction. Based on studies into the effectiveness of many instructional programs, Merrill suggests that the most effective learning environments are problem based and involve the students in four distinct phases of learning: (1) activation of prior experiences, (2) demonstration of skills, (3) application of skills, and (4) integration of these skill into real-world activities.

3.4.2 Conclusions for Alfanet

From the viewpoint of developing Alfanet the most important requirements for an ID model are that it will allow for the creation of courses that 1) are effective, 2) are adaptable to learners, 3) allow for collaborative learning, and 4) provide clear, complete and broadly applicable instructional design guidelines.

According to Merrill (2002), learning environments are effective if they are problem based and address the first principles of instruction for each phase of the activation-demonstration-application-integration learning cycle. Four ID models were discussed that incorporate many but not all first principles of instruction. A combination and fusion of the: 4C/ID, GBS, and CPS models might be the basis for a highly effective set of ID guidelines.

With respect to adaptability to learners, neither of the models appears to have aspects that would disallow an adaptive instructional process. Of the four models only the 4C/ID model provides guidelines for how an instructional program designed according to the model could be adapted to individual learners.

Collaborative learning is not a first principle of instruction and should therefore not be a starting point for instructional design. There is a need for guidelines that state the conditions when collaborative learning is useful and when it is not. Of the four models CPS provides the clearest guidelines for the design of collaborative learning environments.

The 4C/ID model is a comprehensive instructional design model that provides clear guidelines and involves all of Merril's first principles. However, the model primarily focuses on the design of training for learning complex cognitive skills, it does not address instructional design for learning a body of knowledge. A strong point is that an authoring tool for developing 4C/ID training designs is being developed. Schank's GBS primarily focuses on the design of learning by doing simulations. Nelson's CPS solely focuses on the design of collaborative learning environments. For the Alfanet system a combination and fusion of the models: 4C/ID, GBS and CPS appears to be a good basis for specifying instructional design guidelines that covers all of the requirements of the Alfanet system with respect to instructional design.

3.5 appendix 5 Web mining

3.5.1 Overview

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services. Web-mining tasks can be decomposed into resource finding (retrieving intended web documents), information selection and pre-processing (from retrieved web resources), generalization (discovering general patterns at web sites) and analysis (validation and/or interpretation of mined patterns). Three Web mining categories can be distinguished, Web content mining, Web structure mining and Web usage mining.

One of the key points of the system to be developed in the Alfanet Project is adaptation in the three dimensions: Content Level Adaptation, Navigation Level Adaptation and Collaboration Level Adaptation. To

achieve adaptation, as much information as possible about the user activity and behaviour has to be collected and analysed by the system. Three web mining techniques can be distinguished: (1) Web Content Mining, the process of extracting knowledge from the content of documents or their descriptions; (2) Web Structure Mining, the process of inferring knowledge from the WorldWide Web organization and links between references and referents in the Web; and (3) Web Usage Mining, the process of extracting interesting patterns in web access logs.

3.5.2 Conclusions for Alfanet

Web mining techniques can be very useful for achieving several of the adaptation tasks proposed to be done by the system:

- Free-text or semi-structured analysis can be very useful in order to retrieve information from news, forum messages, etc. that can be used to learn the interests of each individual user, which will be used for content and collaboration adaptation
- Link mining can give support for navigation adaptation tasks
- Information extracted from interactions of the users with the Web is also useful to construct the user model used for adaptation.

Therefore, web content mining, web structure mining and web usage mining are techniques that could be used in the Alfanet project, although a more detailed analysis is necessary to confirm its viability and utility in the system.

Finally, it is important to know that the success depends on what and how much valid and reliable knowledge one can discover from the large raw log data, and that for an effective web usage mining, an important cleaning and data transformation step before analysis has to be done.

3.6 appendix 6 Machine learning

3.6.1 Overview

Machine learning is concerned with developing computational theories of learning processes and building machines that learn. To achieve dynamic adaptation in the system to be develop in Alfanet Project, machine learning technologies have to be used since they can provide the dynamic adaptation desired, specially in User Modeling and Multiagent Systems.

3.6.2 Conclusions for Alfanet

To achieve adaptation in the system to be developed it seems necessary to have dynamic user models that adapt to the user's interaction with the system [Garrido and Gea, 2001; Hoppe, 1995]. These adapted models allow the system to provide the user with an adapted response to these models. Machine learning techniques can be used for user modelling, and therefore, can be used in the development of aLFanet.

Nevertheless, the machine learning field is very rich in research and lots of theories exists. Time and effort will be needed to spend in order to select the best technique to be used for each particular case.

The system to be developed will be used by wide and heterogeneous users, whose needs will be completely different, and it is impossible to know beforehand which is the appropriate task in each case. The solution is using machine learning techniques to combine different strategies.

Some learning tasks have already been implemented in the aLF platform, and can serve as the basis for others to be performed by the system to be developed in aLFanet Project.

3.7 appendix 7 User modelling

3.7.1 Overview

Adaptation is essential in any e-learning environment because its users have a wide variety of backgrounds, interests, level of experience on the use of resources, etc. and they demand an environment that adapts to each individual user needs. To carry out this adaptation a user model representing the user's knowledge state, preferences and goals has to be constructed. The adaptation is done through user model acquisition

from the user data available and from traces of interaction of that user with the system. Different kinds of data have to be considered in order to build the user model: data about the user, data about the computer usage and data about the user's hardware, software and physical environment. Different techniques to acquire these data have to be employed, depending on the nature of each data. Two approaches can be used in user modelling representation and inference, one based on knowledge representation that uses a representation formalism and can perform reasoning tasks and the other one based on machine learning techniques.

3.7.2 Conclusions for Alfanet

An overview of the two basic techniques that can be used in user modelling: knowledge representation and machine learning is given. Both approaches lack important features and therefore a combined approach seems to be the best choice. For the Alfanet system a hybrid approach is recommended for defining the user model. This proposal has to be further investigated in order to confirm its viability, but anyway, we will point out now some features that will have to be analysed:

- Explicit knowledge representation in the system to allow direct access to the user model and the use by different learning processes.
- No domain-dependent knowledge in the system, only generic acquisition rules and some reasoning based on the already acquired data to extend user model (the objective of aLFanet Project is self-learning for work in any subject; therefore, no domain-dependent knowledge can be used).
- Observations about users' behaviour should be coded in the available representation formalism and used to form training examples.
- Incremental acquisition, taken into account past interactions using different weights depending on the antiquity of each interaction.
- Learning components used to learn from interaction tasks so the user model holds the changing needs.
- The learning results obtained by running the learning algorithms on the training examples should be transformed into explicit assumptions.
- Heuristic acquisition methods for quick and dirty acquisition from a small number of observations and short usage periods should be used.
- Model contents should be both behaviour-related and mentalistic
- Decision should be supported based on learning results and on the user model contents themselves
- All user models should be kept in the system, and machine learning techniques should be used for group modeling by clustering user models into user group models. Collaborative-filtering can be done to reduce the amount of time to construct a reliable model of user interests. Perhaps an initial stereotyping could be done to provide some adaptation at the beginning, when the user is not known yet.

3.8 appendix 8 Intelligent agents and multiagent architectures

3.8.1 Overview

Appendix 8, Intelligent agents and multi-agent architectures, gives an overview of what software agents and multi-agents systems are, focusing on their application to education, mainly in terms of Intelligent Tutoring Systems.

Multi-agent architectures have a great potential that can be used to solve many of the current problems of web education oriented systems.

3.8.2 Conclusions for Alfanet

Multi-agent systems can be used to reach the adaptation and personalization in the platform.

In order to achieve good personalised and adaptive e-learning, the system could make use of heterogeneous agents that combine the solutions learned with different biases corresponding to different

machine learning methods. It is thus possible to apply different approaches to different tasks and clarify which option is the most appropriate for each instance.

The flexibility and robustness obtained from multi-agent systems makes them suitable to be used in Web environments. Multi-agent systems oriented to Web will be able to learn using machine learning tools and could simulate an expert when making pedagogical tasks.

Based on the above considerations, multi-agent systems improve the system by doing it viable in opened knowledge domains, where ITS were earlier applied with less success.

The use of machine learning tools is needed since the environment will evolve both with time and users. The system will have to personalize the interactions with the users using these techniques.

3.9 appendix 9 Web access and services personalization

3.9.1 Overview

Web Personalization is the technology of dynamically altering the presentation of a web site according to the preferences of the user in order to build user loyalty and typically is used by web sites to filter or recommend the content and navigation choices displayed to each user.

Personalization of Web Services is marketing oriented. In the educational environment creating customer relations is not the goal, but efficient learning and tutoring. However, concepts and techniques described for web services can be applied to an educational environment, since in both fields personalization involves the process of gathering user information during interaction with the user which is then used to deliver appropriate content and services. A set of rules for a useful personalization system design are given in the appendix, which can be helpful when designing the Alfanet system.

3.9.2 Conclusions for Alfanet

Currently, most web services provide customization instead of personalization, with contents tailored for a predefined audience. In a marketing environment customers do not need to be individually defined and it is enough to determine target groups and what contents are appropriate for those groups. However, the system we are to develop in aLFanet should adapt individually to each user, by real personalization.

The rules for designing useful web personalization systems given in this appendix should be taken into account when designing the aLFanet system, specially the ones that refer to not annoying the user while collecting the data and to give the user the last word when using the system. Privacy and information protection, which is an issue of great importance in web access and service personalization, has to be also a key point in aLFanet system. Moreover, the principle of share-and-connect is also applicable.

Mobile devices demand the use of an effective adaptive personalization due to their intrinsic constraints. Requirements fulfilled by adaptive personalization in mobile devices should also be considered in the aLFanet system.

Having in mind the rules for good designing of web personalization systems and the requirements for adaptive personalization, the aLFanet system should, at least, take into account the following topics:

- Provide a good initial non-personalized experience and learn quickly, so users check the benefits of adaptive personalization soon.
- Adapt quickly to changing needs of the users
- Collect and study all data that can be used to individually adapt the system to each user
- Do not require hand tagging on content with category labels
- Do not place resistance in front of personalization by forcing actions which are costly (in effort) to users
- Allow the user to prove wrong system decisions
- Be able to recover from user's wrong actions
- Never leave the user waiting for a personalized response
- Respect individual's privacy

3.10 appendix 10 EML – Learning standard

3.10.1 Overview

Current learning technology specifications allow only for some simple ordering and sequencing of resources used in e-learning (e.g. SCORM, IMS Content Packaging, IMS Simple Sequencing). EML adds to this the ability to integrate learning designs ('instructional designs') to enable more advanced e-learning applications, e.g. to model competency based education, portfolio's, collaborative learning and personalisation.

EML is a semantic specification, based on a pedagogical meta-model, which describes the structure and processes in a 'unit of learning'. It aggregates learning objects with learning objectives, prerequisites, learning activities, teaching activities and learning services in a workflow (or better 'learning flow'), which itself is modelled according to a certain learning design.

The objective of Alfanet, to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods, requires the capability to model both structure and process. EML (including Edubox c.f. section 3.1) can offer this capability.

3.10.2 Conclusions for Alfanet

The objective of Alfanet to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods requires the capability to model both structure and process. EML (including Edubox) offers this capability and equally important in depth knowledge of EML is available and directly accessible, so a quick start can easily be made. In other words EML is a logical candidate for inclusion in Alfanet.

To assure a successful inclusion a number of points have to be taken care of:

- All partners of Alfanet have to acquire the appropriate level of knowledge of EML. This ranges from the basic knowledge of the language, to the design process of ‘units of learning’, to the technical consequences to be able to fit EML in the proposed multi agent architecture.
- The relation of EML with other standards to be included have to be investigated, in particular in the area of knowledge management and multi agent architectures, and the technical consequences have to be documented.

Finally, it is important to notice that Alfanet, given its unique requirements and approach, can by using EML contribute to the further development of standards by validating the current scope and/or proposing extension(s) necessary for a multi-agent, adaptive approach. The required work to achieve this fits in WP3 ‘Standards’.

3.11 appendix 11 Educational ontologies

3.11.1 Overview

In the past decade research efforts have been directed to the problem to unambiguously model educational structure, content and processes.

However educational processes are now rapidly extending to open environments where materials, actors and processes cannot be modelled completely in advance.

For example in cases where competencies have to be trained in realistic environments, based on constructivist learning principles, more open learning structures are often needed.

This means that support is needed for a higher level of semantic typing of relevant objects in the study processes.

Educational ontologies should facilitate the meta-modelling of the relevant entities in the educational processes to:

- facilitate broader reuse;
- give access to content that is emerging in the actual study process;
- anticipate unexpected and unanticipated use;

- adapt to changing learning-needs of the student;
- adapt assignments to changing conditions by defining assignments on a meta-level;
- give access to various data (personal and non-personal) for evaluation purposes to analyse learning activities on a meta-level;
- facilitate automatic processing of separately developed materials;
- facilitate integrated analysis of separately modelled processes.

To fulfil these demands shared educational ontologies can help to create 'networks of meaning'.

3.11.2 Conclusions for Alfanet

Educational ontologies are a very interesting field to study within the scope of the Alfanet project.

To build the intelligent tutoring systems of the future, knowledge at a meta-level of the different actors and resources has to be available and intelligent agents are necessary to interpret meta-information by using various inference mechanisms.

So:

- incorporate ontologies to describe materials, people and processes at a meta-level;
- differentiate when considering the incorporation of educational ontologies; do not consider ontologies as a central design principle, but adopt ontologies in cases where the advantages are obvious;
- develop software agents with various inference mechanisms that use meta-information to help people and systems make use of all the relevant available information, both inside and outside the educational system.

A critical point has to be taken into account as well: an open learning environment is a prerequisite for the meaningful use of ontologies.

3.12 appendix 12 XML and related technologies

3.12.1 Overview

The evolution of information representation and use is currently based on XML standards. XML is an innovative way to allow different environments to communicate and treat information in a standard and structured way. XML technology also provides tools and languages to treat information and to show it. Due to this, it is important to consider this technology as the base platform to develop any software that has information organisation, treatment and communication as its targets.

On the other hand, it is important to consider data storage. For this reason it is necessary to study the XML storage products and technologies.

3.12.2 Conclusions for Alfanet

The use of **XML** in Alfanet will allow to separate data content from presentation. The separation of both aspects is very important. In those web applications where data and presentation are linked, the maintenance of them is very complex. Any variation in the aspect of the application imply to recompile the whole application and to modify lots of graphic elements.

In order to isolate the information structure and content from the visual presentation of these data we propose the use of XML in Alfanet project. Besides, we will need XSLT transformation patterns that will be applied to the data structure defined in XML. The output will be automatically generated in HTML, WML, … and sent to the client. With the use of these standards we will get more flexibility in our application, besides more traceability, maintainability, change resistance, and provide to the whole application a homogeneous aspect for all elements of the same type.

As a consequence of these data and presentation separation, it will be possible to reuse the modules in charge of the data generation. Using different presentation templates for the same data set, Alfanet will be able to:

- Present the same content with different aspects, depending on the context where it will be presented or depending on the user profile or preferences
- Provide service from the same application to different types of client devices (PC, telephone, …). In order to do that, it is necessary to include some routines in the server that will decide depending on the type of connected client, and will apply different specialized templates.
- Provide compatibility and interchangability by the use of approved XML standards.

All these features guide us to the use of XML in Alfanet, and it would be very important to store the data in a XML database. The **XML database** will avoid the transformation from raw data to XML and vice versa when we have to access or store to the database.

In any case, it will be in a more advanced stage when this decision will be taken, but at the moment all works are focused on this idea.

3.13 appendix 13 Knowledge management tools

3.13.1 Overview

E-learning systems consider as an important part of them knowledge management (knowledge tracking, courses contents…). Alfanet is taking into account the existing standards about this field (KM), the common features that the most important related tools provide and the possibility of future integration with these ones.

3.13.2 Conclusions for Alfanet

Some of the features of Knowledge Management are considered in the initial proposal of the Alfanet toolkit in the Technical Annex. Collaboration is included as one of the main features of the Alfanet e-learning platform. So the Alfanet consortium should evaluate the possibility of using some KM collaboration tools within Alfanet system instead of building a new collaborative environment.

Another important issue to take into account is that it will be very useful that e-learning contents will be stored in such a format, compliant with standards, that could be used by KM tools for content management. In this way both e-learning platforms and KM tools could use the same contents.

The integration of the Alfanet platform with KM tools could provide the users with information and contents available in distributed repositories outside the e-learning application. Other possible relationship of KM with authoring tools is to help to capture tacit knowledge into explicit knowledge.

The integration of the Alfanet platform with HR tools which cover different aspects of the human resources management, such as administration, succession planning, leadership development, competency management and organizational charting, should also be considered as an Alfanet integration. This integration could be achieved by XML communication.

3.14 appendix 14 Security concepts

3.14.1 Overview

It is very important, within any system that manages and stores any kind of information about people, to control the access to the data. Personal data, preferences, knowledge, etc. are information that Alfanet uses for working. On this way it is necessary to study the different technologies about security, and more exactly, Internet security, considering within this field the data transmission, storage and privacy.

3.14.2 Conclusions for Alfanet

The findings about privacy reviewed here can contribute to the Alfanet project in a number of ways:

- The tradeoffs on interface design will be noted, and implementations of ‘unobtrusive’ interfaces will be accessed and absorbed, where appropriate, into the prototype.
- User modelling privacy in Alfanet will benefit from compliance with P3P, and the design team may also wish to develop a more detailed specification along the lines described within Cobricks section.

The technical requirements of Alfanet system security are anticipated high, since trust is a central aspect between users as well as in the system. The state of the art in security technologies is sufficient for the implementation of a security scheme for the Alfanet.

As technologies applicable to Alfanet, the following could be considered:

- Data Encryption (SSL – Secure Socket Layer, XML ciphered).
- User Authentication (Single sign-on for Alfanet users across all interfaces during runtime; Username/Password protection for content at URL level)
- Authorisation of exchanges of information (ACL – Access Control List).

In an initial approach, these technologies are used to form different levels of security on profiles or parts thereof for defined users or groups are envisioned. This means that special customisable rules could be applicable for each profile to ensure privacy.

The consortium should take into account security and privacy aspects for achieving an open design that will assure the future exploitation of the Alfanet toolkit.

4. Final Conclusions

In this section the conclusions from the different appendices are integrated to provide answers to the original questions that were set out at the beginning of this study, namely: 1) what are the existing components that can be readily used within the Alfanet system, 2) what are the relevant technologies and approaches that can be applied for the design of the Alfanet system, and 3) what are the relevant peripheral systems that the Alfanet system should be able to interface with and/or standards that it should comply to. Possible answers will be provided for each question by stating principles that the design of the system should adhere to. These principles can serve as a starting point for discussions and decisions about the design of the architecture for the Alfanet system. Please note that they should not be seen as final design decisions. The principles will be listed for each of the three questions. For each principle it will be specified from which appendix it was derived. For question 2 the principles will also be related to the main functional components of the basic LMS architecture as described in chapter 2.

Existing components

- The Alfanet system incorporates and further builds on the Edubox E-learning platform (appendix 1)
- The Alfanet system incorporates and further builds on the aLF E-learning platform (appendix 2)

Relevant technologies

Authoring tool

- The authoring tool allows the development of content for problem based learning environments taking into account the First principles of instruction (appendix 4).
    - A fusion of the 4C/ID, GBS, and CPS instructional design approaches is used for the design and specification of content.
- The general framework for designing CSCL environments developed by OUNL will be used as a starting point for the development of functionality for the Alfanet system (appendix 2)
- As part of the authoring tool, facilities are available to mark resources for web-mining (appendix 5).
- As part of the authoring tool, facilities are available to mark resources for web-personalization (appendix 9)
- As part of the authoring tool, facilities will be available to model resources and processes on a meta–level in terms of DAML+OIL (appendix 11).
- The authoring tool allows the development of content according the EML standard for modelling the structure and process of learning (appendix 10).

LMS Administration Facilities

- Models for the LMS administration facilities are described in meta-models before making the actual data models and process descriptions, especially for data from interactions with objects outside the LMS (appendix 11).

LMS Auditing Facilities

- Web mining tools are part of the LMS auditing facilities (appendix 5).
- Intelligent agents provide specific information for auditing, inferred form the user-model LMS support component (appendix 8).
- Auditing facilities that gather information from outside the LMS are modelled on a meta-level before modelling the actual interface between the LMS and the external systems (appendix 11).

LMS Corporate Lecturing System

- Intelligent multi-agents provide specific information according to user-needs and guidance inferred form the user-model (appendix 8).
- Intelligent multi-agents are central components of the LMS corporate Lecturing system to provide specific information according to user-needs and guidance inferred form the user-model (appendix 8).
- User models will be modelled in XML.
- The LMS corporate lecturing system provides facilities for web-personalization (appendix 9).
- The user-model and the content from outside the LMS are modelled on a meta-level (appendix 11).
- Agents are available that can relate resources within the LMS to resources from outside, based on meta-level descriptions (appendix 11).
- An adaptive approach is used that adapts to a learners a) prior knowledge and skill, b) learning capabilities, c) performance level/ knowledge state, d) interests, e) personal circumstance and f) motivation (appendix 3)
    - Aspects from various existing approaches will be combined to achieve this adaptive functionality
- The adaptive approach is based on a problem based learning environment (appendix 4).
- Machine learning techniques are used to model a users learning process (appendix 6)
    - The best machine learning based, modelling technique will be chosen in the architecture phase of Alfanet.
- The Alfanet system uses a hybrid approach (a combination of knowledge modelling and machine learning) for defining user models (appendix 7).
- The system's reasoning with user model data for inferring adaptive measures will be done using: deductive reasoning, inductive reasoning and analogical reasoning (appendix 7).

LMS Support

- The repository of the LMS support component is easily accessible for web content mining, structure mining and usage mining (appendix 5)
- The repository of the LMS support component contains facilities to represent knowledge on a meta-level (appendix 11).

Peripheral systems and standards

- The Alfanet system should be opened to the integration with Knowledge Management Tools following the current standards and techniques (appendix 13)
- The Alfanet system adheres to XML and related technologies (appendix 12)
    - In the design of Alfanet content and presentation are inherently separated.
    - XML is used as the standard tool to describe content, XSL to express style sheets and XSLT to describe the transformation patterns to the different outputs.
    - XML content is stored in a XML database.
- The Alfanet system adheres to Privacy and Security Concepts (appendix 14)
    - Taking care of privacy of all the actors that use the LSM is an inherent design parameter, based on a basic privacy Charta for Alfanet that will reflect the basic privacy demands and regulations within the various cultures participating.
    - The privacy commitments of Alfanet could be expressed using the P3P protocol.
    - SSL is used as standard for encrypting data traffic over the Internet in the application points where it is considered necessary.

- Access control lists are inherently modelled in Alfanet to authorize access to resources.
- Users are placed in groups that reflect common security concerns.

- The Alfanet system adheres to the EML standard for modelling the structure and process of learning (appendix 10).

Appendix 1 E-learning platforms

A 1.1 Overview

The Internet offers possibilities to re-organize learning, teaching and education. It allows for flexibility in delivery, but also in learning, in time and place. It should also be easier to differentiate didactical models and scenarios depending on users' preferences. Many tools arose, claiming to fulfil these needs both in learning as tutoring, as distance learning. These tools are generally referred to as electronic learning environments/platforms or learning management systems (LMS). Many terms are used: virtual class room, virtual learning environment (VLE), digital learning environment, electronic learning environment, computer-based training (CBT), learning management system (LMS), course management system (CMS), content management system (CMS), learning content management system (LCMS), etc. These platforms can have many features, including distribution of content, tutoring, assessment, communication and administrative tasks. It is a fast evolving market that sees an evolution towards more flexible systems (interchangeable components) and towards integration of international standards like IMS. There is a lot of variation in the definition of electronic/digital learning platforms. Sometimes merely the use of educational software is considered to be an electronic learning platform. In the context of ALFANET only those platforms are relevant that consist of at least content delivery and tutoring facilities via internet technologies, combined with communication facilities and organizational and administrative functions.

Some tools arose following the changed vision on learning, which arose over the past few years. New insights resulted in new pedagogics, like individual learning paths; constructive learning or competency based learning, while differences in various styles of learning are acknowledged. Also developments in society demand changes in education, which should be more flexible, use competency and be of constant high quality. This has consequences for the learning process. The teacher will become more of a coach; learning tools should be able to be used by students with different styles and at different levels. Actual time of face-to-face education will decrease, therefore obtaining knowledge, building portfolio and cooperation will become more important.

A distinction should be made between integrated systems and electronic learning environments, which support only parts of the integrated system. The integrated system should provide in electronic/digital tools for creation, management and distribution of learning material, learning design, tasks and assessments. It should facilitate communication between students and between student and tutor. The system should also at least be able to enroll students, to monitor progress and manage results.

There should be a clear definition of LMS. LMS, Learning Management System, used to refer to administrative and organizational aspects. Nowadays a LMS tends to be an integrated system, which could also provide course delivery. Sometimes there is made a distinction between LMS and de e-learning environment.

ALFANET refers to a LMS as an integrated solution combining a learning environment with administrative and content management functionalities.

Other definitions consider e-learning services to consist of infrastructure (communication, authentication, registration, etc), and application architecture. The latter contains an e-learning environment geared to the learning process and a learning management system for the administration. The content management system, in which content is created, developed and maintained is not part of this.

Some definitions make a distinction between LCMS (learning content management system) for development and management of content, including authoring tools, project management, repositories and version control, electronic learning environment (also called VLE, virtual learning environment) for the student, and LMS (learning management system) for administration of student data and progress, including portfolio and competence management. Examples of administrative learning management systems are Docent, Saba, Click2Learn, electronic learning environment cover Blackboard, WebCT, Fronter, Lotus LearningSpace, examples of CMS are Netschool, Viviance, HarvestRoad. Most applications profile themselves in one of these areas, but mostly combine aspects of other areas. Combination systems mainly combine the administrative and e-learning system, or the e-learning system and the content management system. Saba for example combines functionalities of all 3 areas, while Blackboard and WebCT are pure elearning environments, although Blackboard is moving towards content management.

Another term encountered is ILS (integrated learning system). An ILS is a LMS with additional functionality like authoring tools, content management, and knowledge management. In the Netherlands the term

'teleleerplatform' is more and more used. The term is chosen to indicate that the integrated learning environment should be an integrated system, consisting of the technical facilities (hardware, software, infrastructure) needed to facilitate the interaction between the learning process, the communication needed for the learning, and the organization of learning (Droste). This definition applies that a suitable system should incorporate functionalities of all 3 major areas: LMS, LCMS and electronic learning environment. Examples of systems that are used in the Netherlands are WebCT, TopClass, Lotus Learning Space, Constructor, Koepel, Virtual Campus, Blackboard, Holo-E, TeleTop, and Edubox. Others are Docent, Firstclass, Edubox, Ingenium, and N@TSCHOOL.

Not all systems support more traditional forms of education, so it might be important to consider this. This is also referred to as ILT instructor led training.

Integrated systems like WebCT, TopClass, and Lotus Learning Space combine functionalities of content, communication and organization in one system. However, there seems to be a trend to create system of components, e.g. external test/assessment system, content databases, communication/collaboration facilities, administrative systems etc are combined with the elearning platform (Docent, Ingenium, Edubox, HOLO-E, TeleTOP).

Some platforms take the course as basis, others the organization. That is reflected in the roles that are distinguished in the system and their rights. Most platforms that take the course as starting point do not distinguish between tutor and course developer, thereby allowing the tutor much flexibility in layout, but also assuming that the tutor will create material; e.g. WebCT, TopClass, Koepel, Constructor, TeleTop. Systems that take the organization as basis are usually component-based (like Docent.com, Ingenium, Edubox, Holo-E, and TeleTop). They have clearly defined and distinct roles. It is possible that content is developed outside the system (Edubox, Docent, Ingenium). A third group is a hybrid of both functions (e.g. Holo-E, Blackboard, and Learning Space). They have a more distinct role definition; provisions to work with larger numbers (material and students) and have a database allowing content to be reused.

Most systems tend to lean more and more to the corporate market. Thus decreasing support for the higher education market and increasing prices.

There is a wide range in functionalities offered by the different systems. Most systems used in universities and higher education are relatively simple. They are used by teachers to complement classroom teaching. These systems are Internet based, provide content over the web and provide communication facilities. However, more and more the systems tend to evolve to larger, more complex ones. The systems seen in large corporations and businesses tend to be more complex. These systems belong more to the 'knowledge management tools' and tend to provide skill assessment. These systems are mainly used to enhance skills of employees.

As Zeiberg states there can be varying business needs for a LMS. This can vary from simply automating the training administration to enterprise wide deployment of learning. There are functional and non-functional requirements of an LMS. Functional requirements cover: content management, curriculum management, class scheduling, course delivery, competency management, assessment, testing, evaluation, compliance, certification, accreditation, resource management, inventory management, finance management, content authoring, system administration, reporting, definition of user role in LMS, user profiles, communication, collaboration, etc. Non-functional requirements can be performance, user interface, business domains, global access and functionality, application and database management, user and system documentation and training, security and audit, flexibility and scalability. Then there are architecture and system requirements like networking, hardware, software, operating system, and client workstation specifications. Requirements of interoperability to comply with standards like SCORM, AICC, IEEE, IMS, and technical interoperability (MAPI, OLE, TCP/IP, ODBC etc). Integration with other systems, e.g. ERP system, finances, communication (e-mail, conferencing, news, listservers, etc) and other applications.

Most systems advertise their products as webbased servers that provide content, communication, delivery and student enrolment. Some focus on additional areas like assignments, assessments, student performance assessment, tracking and analysis, gradebooks, portal, re-use of content.

Systems that provide content authoring and management can include drag-and-drop features, sharable media library, interactive course map, html editor, spell check, equation editor, search capabilities, batch upload/download for authors, import and export of content either in the system's format or to other formats, use or convert existing content.

Additional features mentioned are library and information access, annotation/markup, glossary help, study skill building, access to newsgroups/listservers, automated table of contents and indexing, course search engine, whiteboard, chat, conferencing, web search engine, timed and repeated quizzes and exercises, chat

archiving, bookmarking, content templates, personal home pages, file sharing, student tracking, archiving, backing up and replication of courses, course revision, online help, FAQ.

WebCT for example provide instructors with the possibilities to access, track, and analyze student performance, in order to improve courses and seem to use datamining for this.

Personalized delivery is implemented in various ways, ranging from a customizable interface to education based on portfolios. HOLO-E is developed from the vision to enable recent development in education like individual learning paths, learning styles, competency-based learning, where the basis is the portfolio and the student who controls his learning. Personalization is also applied by providing course schedules, assignment tracking, course communications, grades, course specific news and support materials and content delivery based on demonstrated and assessed proficiency.

Some systems are able to deliver the content to multiple media. This is mostly restricted to website, and webpages on CD-ROM and DVD.

A customizable user-interface is important to enable student learning styles and enhance accessibility.

Another important features are reporting, assessment of student progression, grading and maintaining this information in digital gradebooks or portfolios.

Possibility to create and assign roles is important not only for security reasons, but also to facilitate certain pedagogical models.

Re-use of content is very important to allow efficient course building. To enable re-use the material should be easily located, by adding metadata and use of a searchable repository. The content should be stored in context-independent manner or should be easily adaptable to other contexts. This is referred to as re-usable learning object by several systems (TKM Generation21, Cisco Virtuoso).

Architectures for the more complex systems are usually n-tier architectures, often based on Java. Most claim to use industry standards, thereby ensuring interoperability and open platforms. Of course this is no guarantee for interoperability. WebCT for example is a four-tier web application architecture, J2EE compliant, using Oracle at the database layer, providing system server clustering, session fail-over and database clustering. This optimizes performance and availability. Java API, IMS API, LDAP and Kerberos support provide extensibility and security mechanisms.

In particular the American systems advertise their compliance with Section 508 of the Americans with Disabilities Act, but accessibility is an important factor for all systems.

Use of and compliance with standards like XML, IMS, IEEE, AICC, SCORM may be important to ensure interchangeability of content and exchange between organizations. Support for these standards still varies among systems, but is increasing. Blackboard and WebCT participate in IMS.

Other features include a modular and open architecture, providing both a LCMS, LMS, and elearning environment, collaboration, personalized delivery, information management, performance management, simulation editor, auditing, support of industry standards, interoperation with other standards-compliant learning products (AICC, SCORM, XML), integration with existing systems, improvement of business performance, collaborative authoring, adaptive and personalized delivery, delivery in multiple formats, reuse and re-package of content, webbased, multiple language support, customizable user-interface, role-based security, AICC, SCORM, XML compliant, assess and enhance skill sets, blended learning, manage performance, competencies, streaming audio and video, live interactive sessions, shared whiteboard, instant messaging, personalized and adaptive delivery, adaptive learning based on pre-testing, dynamic studyplans, post-assessment, personalized feedback, identify and communicate objectives, activities, performance, create learning plans, link people to the content they need, track skills, collaborate, measure, competency management, global commerce, student registration and tracking, records management, reporting for any blend of delivery environment (self-paced, web, live interactive web, traditional courses and documentation), multiple languages, individual and 360° assessment, competency levels, personalize learning plans, web-based, real-time communication, collaboration, creating, managing, assembling delivering learning content, subject matter experts, context-driven personalized delivery, learner feedback, assessments, contributions, establish and communicate critical corporate goals, measure performance, common network platform, internet technologies, XML, Java, centrally manage learning content on the internet, allows users to assemble all types of digital media into a single course and publish into html, automate all business functions: set and track business goals, assess performance, competence, plan, feedback, automate processes in managing human capital resources, model current and desired state, key resources, key competency gaps, exchange with providers, n-tier internet architecture, enterprise Java beans, Java server pages, XML, XML.

A 1.2 Solutions

A short description of some solutions is provided here. A more detailed description is given in the deliverable of task 7.1 Initial market studies.

Saba: combines aspects of LMS in administrative sense, e-learning environment and content management system.

Holo-E, based on Lotus Notes, custom interface, portfolio module

N@tschool, complies to Dutch market, geared towards content production and management, publishers' niche

Saba, Docent, Smartforce, focus on administrative LMS, corporate e-learning

Blackboard, WebCT, focus on e-learning environment, higher education

Lotus Learning Space, higher education and corporate market

Topclass, initially administrative learning management and e-learning environment for higher education, now also corporate and content management

Blackboard, leading market

A 1.2.1 Edubox

Developer: Open Universiteit Nederland

URL: http://www.edubox.nl

General overview

There is a growing need for flexible education. New forms of blended and hybrid learning offered by dual mode institutions, combining distance and conventional teaching, illustrate this. New educational concepts guide and ICT facilitates these new forms of education. EML/Edubox is developed specifically to enable blended and hybrid learning.

Edubox entails an innovative educational approach in an electronic environment. It comprises of a unique system for management of education supported by advanced ICT. Edubox organizes both the educational and learning processes as the development and maintenance of educational material.

Edubox is a flexible system. The basis for this flexibility is EML (Education Modelling Language). EML provides the tool to developers to convert their educational and didactical ideas and views to digital learning materials. Learning materials and activities are coded in EML. Once coded their value remains. Investments in conversion are no longer needed. When using EML education is described in a platform independent manner. Edubox can be linked to systems already in use, e.g. student administration, personnel administration, financing.

Edubox provides freedom. Every didactical scenario can be described in EML, be it competence-based education, collaborative learning, problem-based learning or a more traditional scenario. In addition, the digital material can be adjusted for face to face education, distance teaching, dual education or combinations thereof. The material can be delivered on paper, CD-ROM, via Internet or e-book. So, Edubox facilitates all kinds of education, teaching and training.

Edubox is a complete electronic environment that assists guides and supports management, authors, teachers and students, in educational, didactical and management aspects.

In Edubox the educational components consist of an accurate description of activities and content. Because Edubox uses EML, the components can be described in a medium neutral way. Therefore it is easy to re-use them in many manners, or exchange them with other institutions.

Edubox does not prescribe a didactical scenario. Every scenario can be modeled in EML. Edubox provides additional options. Every piece of content or activity can be personalized or made available for specific users. This means that education can be made to fit every user's needs and preferences.

Edubox utilizes all potentials of advanced information and communication technologies as a powerful means to create a complete and flexible electronic learning environment. Edubox can represent virtual, simulated multimedial realities in many ways. This is demonstrated in four areas: power, systems, personalized 'studyplace', opportunity. Education can be created without being restricted by the 'real world'. Students are placed in an authentic 'working situation', which stimulates active and dynamic solutions. Support and

assessment can be integrated. Edubox supplies a powerful environment. Users can decide when to use the learning environment. Time and place independent communication is possible via e-mail, listservers, newsserver, video and audioconferencing. Edubox is used to create a learning environment personalized for a user, based on their role in the environment, and a personal dossier/portfolio with preferences, existing knowledge and skills and possibilities. Objective, content, method, order, navigation, presentation and didactics can be personalized by designer, teacher or student. Edubox contains tools to support collaboration between persons in the learning environment, e.g. to enable students to create a group-essay.

Edubox consists of multiple systems. There is an authoring environment in which authors, designers, teachers, create learning material in EML. This material is maintained in a content management system to facilitate re-use. The Edubox player plays EML files, so there is education. It can be delivered to various media. The player can convert EML to webpages that are accessible via the Internet, or convert it e.g. to printable media. The Edubox portal is the point of access for teachers, tutors and students. It contains the personalized 'studyplace'. Edubox also provides tools to manage and administrate a large number of complex processes, e.g. enrolment, run and publication management, etc.

Edubox provides the opportunity to meet the trends in education towards lifelong learning and competence development. With Edubox the curriculum can be optimized to a coherent whole. Education can be provided according to the users' needs. It can be accessed time and place independent. Knowledge and competence are saved in a personal dossier. Educational components are saved digitally and medium neutrally in a content management system. This allows efficient maintenance, re-use and integration.

Edubox creates a personal 'working place' on the Internet for teachers, tutors and students in web-based education. This is the most visible part of Edubox. The author compiles the electronic work and study environment. Therefore it contains only those functionalities needed by each of the roles. Layout and styling can be changed to reflect housestyle. These environments are very dynamic. The environment is based on the personal dossier/portfolio of a student or a group of students. Edubox supports the creation and maintenance of these dossiers. The dossiers contain existing knowledge, skills, preferences and situational factors. A personal education tract is constructed based on this profile. Edubox delivers the corresponding education, based on student data, teacher and study materials. Student reports, essays and assessments are stored in the personal dossier. Changes in the dossier may lead to changes in the education.

Teachers design and structure the education in the electronic learning environment. On their personal desktop they have access to the authoring environment with all instruments needed to code the education in EML. EML is platform independent and does not add layout and styling aspects. Therefore all kinds of didactics can be structured in EML. The material receives an identification number and is stored in a database. This allows easy re-use of materials. It facilitates design of education fit to the users' needs. It can be delivered to any medium or mix of media.

Tutors support a student or group of students in the electronic learning environment. So, they have access to the course or module that they support, and to the same facilities as the students. Of course, tutors find here all facilities to communicate with the students. In addition Edubox provides information about students' progress and indicates how often they have visited the learning environment. Tutors can add additional information, provide tips and hints, or make students aware of new developments. The tutoring environment is specified in EML and is personalized, so tutors see what they need at a particular moment in time, depending on their role in the learning process.

Students follow personalized education in the electronic learning environment. They have access to the learning material, as specified in their curriculum, from their 'studyplace'. Studyplaces can differ amongst students in the same course. Students can see, e.g. intake instruments, activities, assignments, resources, etc. There are a couple of standard facilities to communicate with teachers or fellow student, ask for help, or collaborate.

Communication between students and tutors and between students is possible via built-in functions in the Edubox player, via the Edubox portal, or by linking to external communication facilities, like e-mail and (a)synchronous conferencing tool. EML specifies when to use what facility under which conditions.

EML offers many possibilities in modeling education, not only content but also didactics, which is usually not provided in the other electronic learning systems. Unique is the creation of dossiers and use of portfolios. Several types of interactions can be used. It is possible to create a specific tutoring environment. EML offers high potential in meeting needs of digital learning environments.

Edubox offers tool to manage courses, staff, learners, and roles.

Every educational institution has several organizational and administrative (legacy) systems to store e.g. student progress and personal data. These systems also provide crucial information for organizing the necessary processes. These systems can be linked with Edubox.

EML/Edubox is platform independent. Edubox is based on standards and common practice technologies, like the W3C recommendations (webdesign, accessibility, html, XML, XHTML, soap, XSL etc), and others like SQL, TCP/IP, http, IMS, IEEE, etc. Its architecture is modular and components-based. This allows exchange of modules, for example the authoring environment, content management system.

EML is accepted as basis for the Learning Design Working Group of IMS for developing a learning design standard.

ALFANET innovative features

EML/Edubox allows not only for modeling of education content, but also didactics. Edubox can play any didactics modeled in EML. EML/Edubox delivers personalized education. Pre-assessment is used to determine what building blocks are required to acquire the necessary skills and knowledge. Edubox is unique in creating personal dossiers and the use of portfolios. Edubox allows different environments (belonging to the same course) for students, teachers and tutors. The environment only displays those components that are relevant to the role of the user in the system. Communication and collaboration facilities are present. All material, content and didactics, laid down in EML is medium neutral and can be delivered to several media or mix of media. Re-use of material is easy. The user interface and language of the Edubox player is easily adaptable and can be changed to suit user preferences. Edubox can provide integration with existing backoffice applications. Edubox architecture is based on components, and is not dependent on a specific authoring environment, specific databases or webservers. Edubox server and client are platform independent. Edubox uses standards and is based on commonly accepted technologies and best practices. EML is the basis for the Learning Design standard being developed by IMS. EML is an XML application.

Edubox can be used as an important building block for Alfanet. Edubox enables the support of prespecified adaptive features. Alfanet can extend this by offering a multi-agent approach to learner adaptation dependent from the users' interaction with the system. Integration both approaches will help to keep the focus of the Alfanet project to its main challenge i.e. the educational and technical design and development of the agents and their validation in practice.

A 1.2.2 Blackboard 5

Developer: Blackboard Inc, USA

URL: http://www.blackboard.com

License: Commercial, but educational licenses can be available for example via CHEST in the UK and Surfdiensten in the Netherlands (levels 1 and 2 only).

Product overview

Current version is release 5.5

The Blackboard e-Education Enterprise suite comprises the Blackboard Learning System ™, the Blackboard Community Portal and Blackboard Transaction System.

The platform claims to be highly interoperable and customizable with its Blackboard Building Blocks architecture.

The Blackboard Learning System is a webbased server platform that offers course management and allows for integration with student information systems.

It is available in three levels. The first level encompasses the course manager. This is the basic version, most suitable for an organization that is exploring its use. The portal manager is added in the next level. A portal function is added which allows every user access to specific services depending on his profile. The last level is an extended version of the previous. It offers possibilities for integration with external systems. These have to be tailor-made.

It seems to be a user-friendly tool, in particular for authors and teachers/tutors. In particular authors and tutors just starting in e-learning can make content available in a relatively simple manner. The design is via existing buttons. Content is made available via pre-designed structures. Adding new material is easy, changes in content and design only in limited fashion. Support for creation of tests and question is offered.

Blackboard is involved in IMS.

ALFANET innovative features

There are few possibilities to define and specify roles. Personalization is not possible.

The teacher/tutor environment is fixed, can not be adjusted.

Content can be made available, but Blackboard does not provide an authoring tool, or integration with authoring tools. Content is course specific. When content is used in several courses, the content needs to be uploaded as many times.

There is no student tracking, nor possibilities for automatic adaptive learning.

Blackboard does not offer any tools for didactic design.

There are no repository functions like version control.

Course management, scheduling and planning is minimal, course tracking is absent.

Communication facilities are available, but are limited in function.

There is no multilingual interface. Only limited adaptations of the interface are possible.

Although Blackboard claims to be standards compliant, only subsets of standards are implemented at this stage. The test generator is not QTI compliant.

Several Blackboard customers indicate a poor customer support and high prices, but agree that it is easy to use and is a good tool to use in combination with regular education.

A 1.2.3 Cisco Learning System Virtuoso

Developer: CLI Cisco Learning Institute, a non-profit organization founded by Cisco Systems Inc, but now operating independently from Cisco Systems Inc. USA

URL: http://www.ciscolearning.org

License: CLI intends to make Virtuoso™ available to organizations within its charitable class at little or no cost, under a licensing agreement. When the Solution is ready for distribution, any person or enterprise may license Virtuoso™ from CLI by paying fair market value for their use. CLI is currently developing its detailed plans for making Virtuoso™ available to licensees both within and outside its charity class.

CLI Virtuoso includes course authoring, course delivery (webbased), and course management (learner enrolment, tracking, and assessment) environments. It offers personalized learning experience, and provides a customized, on-demand curriculum, based on learner needs and requirements. It addresses specific learning goals using objects and assessment items. Evaluation of the learner's knowledge and skill before embarking on a course enables a personalized curriculum.

This evaluation is based on information from Cisco Learning Institute and UNICON. No other references could be found.

It is not clear yet, whether the product is available. Cisco Systems Inc uses it for its Cisco Networking Academy Program.

UNICON INC develops the delivery engine, CLI Virtuoso.

CLI Virtuoso consist of course authoring, course delivery and course management. The course authoring system provides content management and an authoring tool. It allows for creation of new content and organizing and manipulation of existing content. Many media types are supported. There is version management of graphic media. Metadata can be added. User and access rights can be set. Several assessment types can be created. Courses can be delivered based on learner profiles. Personalized feedback is provided during the course. Performance levels can be set. Every content page contains a section with additional information. Assessments are dynamically rendered. Look and feel can be adjusted by the use of themes or templates. Content and assessments can be imported from external authoring systems. There is automatic content delivery from the course creation to course delivery. A course portal provides links to course content for current enrolments, assessments, grade book, personalized curriculum, and instructor's website. The course management system is used by tutors/teachers to set or change assessment variables. It also stores results in a 'grade book'. The grade book is available both to student and teacher, and records progress and grades. Students get an overview of all their courses. A simple file format allows importing of external records and management systems. The system provides three roles, administrator, instructor and learner. After authentication, access to resources is based on role.

ALFANET innovative features

Editing of existing content. Courses based on learner profiles. Additional information can be added to every page. Grade book.

A 1.2.4 Generation21 Learning Systems

Developer: Generation21 Learning Systems, wholly owned subsidiary of Renaissance Learning, Inc. USA

URL: http://www.gen21.com

License: Commercial, possible only in USA

This is a paper review, mostly based on information from the Generation21 website.

Started in aerospace industry as a knowledge management system. Now it claims to be "one solution for training, knowledge management, and learning management" ("TKM"). It relies on taking accurate information to employees in an efficient manner, whenever it is applicable. It uses a single-source demand-based knowledge database. There is also a TKM for school districts, used for training of staff and tutors.

It is a system that allows course developers to re-use information, accelerate learning, and provide on-demand performance support, based on Dynamic Learning Objects technology. Dynamic learning objects can be re-used. Changed to dynamic learning objects are immediately reflected in all courses where the object is being used. Intake tests allow users to study only relevant parts. Course development is based on common templates to ensure one common look and feel. It is possible to publish to web, print and CD-ROM. Users can search the database for additional information about the topic. There are LMS functionalities like scheduling, registration, tracking, gradebooks, reporting. Communication is available as e-mail, chat and forums.

Features depend on level of system used, LMS Professional, LMS Enterprise, CMS Enterprise or TKM. Intake assessment and library are add-ons to these systems.

The system is available for Linux, Solaris and Windows SQL. It requires a webserver, a database server and development server. Oracle 8i or MS SQL server 2000 are required. Users need an Internet connection with either IE5.0 or Netscape 4.7 or greater.

The TKM system use XML to define course structure and sequence. It claims to comply with the AICC communication protocols and SCORM. Generation21 participates in the IEEE computer-based training standards committee.

Learning objects should have a defined learning goal, content needed to achieve the outcome, associated media or material, measures to assess whether outcome has been achieved. Dynamic learning objects must be able to be used in whole or part on demand, must be able to be modified on demand, must be able to be used in conjunction with each other in an infinite variety of combinations and on demand.

ALFANET innovative features

Course based on intake. Re-use.

A 1.2.5 LearningSpace

Developer: IBM/Lotus

URL: http://www.lotus.com/home.nsf/welcome/learnspace/

License: Commercial

LearningSpace is part of an innovative family of services and technologies designed to make e-learning in organizations. It covers both one group and entire company training. The LearningSpace family of products includes LearningSpace Forum, and the LearningSpace 5 Core and Collaboration.

- It supports the delivery and tracking of online self-paced learning content. Self-paced courses can be highly structured with controlled paths and prerequisite assessments, providing flexible learner access designed to suit individual learning needs. This can be extended with collaborative learning capabilities that enable learners and instructors to work and learn together using discussion databases or real-time virtual classrooms.

LearningSpace provides a Web-based interface for delivering to a worldwide audience.

LearningSpace includes management capabilities.

The entire e-learning program can be monitored, gaining detailed insights on learner progress

Events -- such as changes in learner enrollment status or course availability -- can trigger delivery of messages to administrators, instructors, and learners.

LearningSpace lets tutors create a wide range of assessments from a browser including tests, surveys, and quizzes (using new questions or by drawing on a question bank). Online quizzes include true/false, yes/no, multiple choice, fill-in-the-blank, and short answers all aimed at assessing learner progress and course effectiveness. LearningSpace lets use a built-in assessment tool, or choose to use any AICC-compliant authoring tool.

LearningSpace is designed to support a number of existing standards, AICC recommendations for tracking and compatibility between content and the learning platform.

LearningSpace offers full support of the API adaptor specification of the Sharable Content Object Reference Model (SCORM).

LearningSpace products are in line with the IMS specifications focused on the development for e-learning.

The collaborative delivery features of LearningSpace are driven by Lotus Sametime technology, which takes advantage of open standards (e.g., H.323 and T.120) enabling real-time sharing of information and communication. H.323 is the ITU-T standard for sending voice (audio) and video using IP on the Internet and within intranets.

Lotus LearningSpace has a documented extensibility architecture based on a Java API. The API can be used to integrate LearningSpace with back-end systems such as HR databases, ERP, and e-commerce systems.

LearningSpace can also be part of your overall Lotus Knowledge Management solution, enabling to link e-learning with Knowledge Management. For example, LearningSpace courses can be integrated into the Lotus K-station knowledge portal.

ALFANET innovative features

No ALFANET innovative (adaptive presentation & navigation, adaptive collaboration, EML standard-compliant, self-assessment, pedagogical methods) feature has been detected in LearningSpace.

A 1.2.6 TopClass e-Learning Suite

Developer: WBT Systems

URL: http://www.wbtsystems.com/

License: Commercial

TopClass e-Learning Suite is a web-based training, authoring, delivery and management. It provides personalized coursework for each student; online authoring based on open HTML standards, secure collaboration tools, and auto testing capabilities. Functionalities are can be extended by using additional modules. TopClass offers e-learning content management and assembly, personalized content delivery, learner collaboration, learner tracking and assessment, web-based training administration, and database administration. It supports conversion of existing learning content to the web, and its unique Learning Object architecture. It enables businesses to provide employees with a single access point to all their training needs. It also allows blended delivery with any combination of on-line self-study and instructor led training including virtual classroom and other materials. TopClass LMS provides catalog and registration functionality, including wait-listing and workflow approval management. The core of TopClass LMS is a catalog that allows learners to find all the learning material in the enterprise from one central source.

TopClass Competencies manages an organization's skills and competencies inventory, and assess resource allocation, while users themselves can identify skill gaps and map training programs specific to their competency shortfalls.

TopClass Mobile enables offline delivery for users without a network connection. TopClass Mobile also facilitates the deployment of content on CD-ROM and DVD. TopClass Mobile also allows course developers a convenient alternative for previewing and evaluating TopClass.

TopClass Publisher is a desktop application for complete web-based course content publishing. It enables conversion of existing content and course assembly.

TopClass Virtual Classrooms provides integration with e-learning providers.

TopClass XML Toolkit integrates TopClass with your existing business systems such as Peoplesoft or SAP.

TopClass provides a browser-based interface. This interface is available simultaneously in multiple languages

TopClass complies with the leading industry standards, including AICC, SCORM, IMS, HTML, and XML.

ALFANET innovative features

Through its comprehensive testing and assessment engine, TopClass LCMS dynamically creates personalized learning paths for each learner on the fly. Allow learners to adjust the format of their courses to the way they learn best. Re-use of Learning Objects to construct multiple similar versions of courses. Track and assess learners. Customize the user interface for each user.

A 1.2.7 Pyxis KMS

Developer: Pyxis Edu

URL: http://www.pyxisedu.com/

License: Commercial

Pyxis KMS provides a personalized learning platform with a set of tools: KMS Enterprise Edition, LCM Tool, Desktop Course player, Offline course builder, Offline assessment builder, Performance analyzer, Virtual classroom. Pyxis KMS empowers collaboration and knowledge sharing between individuals and knowledge experts through chat rooms threaded discussions, emails, and document sharing. Pyxis Virtual Classroom is a teaching and learning environment, which supports collaborative learning among learners.

Pyxis KMS LCM is a learning Content Management Tool. It is standards-based system, including a learning object repository that allows users to search, share, reassemble and store content and learning objects in a central location. Learning objects are reusable units of content. Pyxis KMS LCM can import standards-based learning content and specialized content.

Pyxis Edu Offline Assessment offers a comprehensive solution for an efficient design and execution of evaluation procedures.

Pyxis KMS provides support to learning standards of AICC, SCORM, and IMS.

Pyxis KMS provides an API that allows the integration of external applications such as HRMS, Financials, ERP and CRM platforms.

ALFANET innovative features

Personalized user interface at each level of the system. Pyxis KMS LCM is tightly integrated with Pyxis KMS to automatically and proactively deliver appropriate, personalized content to users based on in-depth profile information. LCM improves knowledge retention and performance by disseminating the right content to learners by tracking results down to the individual questions within a test and document level tracking.

Design and develop the right course curriculum that achieves tangible results.

A 1.3 Assessment

A 1.3.1 Situation and prognosis

There are many systems about; ranging from simple web based course software, to communication applications to full-fledged knowledge management systems. Most available information and even reviews are descriptive, just mention whether a functionality exists, but not how it is implemented, or how well it operates. Presence of functionality is not sufficient, user expectations determine whether it is suitable. A tutor might only want some communication facilities added to his classroom teaching. Most systems provide some form of communication, but it might be more appropriate to use a dedicated communication application instead of a full elearning system as indicated by Cook.

All systems advertise themselves to be innovative and offer new possibilities. They all seem to claim that it is important to use ICT in education and of course that their system is the only one in its kind. However it is very difficult to find any information about theories and technologies underlying the product. Some information about architecture is given, but no more than for example an n-tier application allowing integration with third-party applications.

Hardly any of the systems prescribe structure for the content. On the contrary, most just use any content which can be rendered by a webbrowser. This might make it difficult to re-use material, unless metadata is added according to specific formats and keywords from specified ontologies are used.

There is hardly any information about which didactical methods and models are used by the various system. Many are based implicitly based on a didactical model, others might allow some additional scenario's, like problem based education, or addition of assessment.

Systems that claim to provide personalization hardly present any information about which methods are used to attain this personalization. Personalization often is no more than an adjustable interface.

The same applies to adaptive education. Here also hardly information is available about underlying theories and technologies. One system that did specify how it applies adaptive learning is Aspen Enterprise, which uses pre-testing as a form of adaptive learning.

It is hard to discover any information about which technologies are used by the systems. Of course all have 'electronic delivery', are Internet based, have some kind of repository, etc. However details are usually not provided. Some claim to be based on internet technologies, others to use 'industry standards'.

Presence of APIs merely indicates the possibilities for interoperability, not the ease of customization.

Although all systems seem to stress the importance of content, or emphasize how easy it is for authors to create content in their system, didactics is not mentioned at all. Hardly any of the LMS provide information which pedagogical model is used in their system, if any, nor if other systems can be used. Although some provide a planning, design and/or structuring tool to organize content. Some mention the addition of assessment and communication to content as enrichments of the pedagogical model.

Nevertheless, all systems seem to be evolving towards integration of open industry standards and learning standards as IMS, AICC, SCORM, and IEEE.

A 1.3.2 Conclusions for ALFANET

The usability of the described system depend on user's expectations and needs. As most universities and higher education institutes are only starting in the field, most current systems seem to be value for money for the specific area they market. Many of the systems incorporate the base functionalities and features needed for the ALFANET LMS. None of the systems, except Edubox, model the didactics as well as the content. However, none of the systems meets all criteria for ALFANET. Only the more complex e-learning systems might be a starting point for the ALFANET LMS, but they would require extensive customization. Customization is always risky. It can be very time consuming and expensive. It also makes the product very dependent on specific versions of applications. When the ALFANET LMS is going to build on the basis of an existing system, it might be wise to choose a system of which one or more of the partners of ALFANET has extensive knowledge and in-house capabilities.

A 1.4 Bibliography and references

10 Things Every Manager Should Know About TKM, Generation 21 technical paper

Advies keuze Teleleerplatform 2000, SURF Educatie<F> CINOP, juni 2000

Aspen Enterprise Learning Platform, http://www.click2learn.com

Blackboard, http://www.blackboard.net

Brandon Hall, http://www.brandon-hall.com/

Bruce Landon, http://www.c2t2.ca/landonline/

BVEnet, http://home.bvenet.nl

CETIS, http://www.cetis.ac.uk/directory

CFL, Centre for Flexible Learning, http://www.cfl.mq.edu.au/cfl/resource/revcom.html

CHEST, http://www.chest.ac.uk/

Cisco E-learning, http://www.cisco.com/warp/public/10/wwtraining/elearning/

Cisco Learning Institute, http://www.ciscolearning.org/

Cook, K. 2000. FirstClass vs. Blackboard. An opinion paper: How Blackboard compares to FirstClass in delivering online courses. Available: http://www.bsu.edu/classes/rice/FC/FirstClass_vs_Blackboard.pdf

Docent, http://www.docent.com

Edubox, http://www.ou.nl/edubox

Edusite, http://www.edusite.nl

E-learning plaza, http://www.e-learningplaza.nl

Generation21 Information Brochure, Generation 21 technical paper

Generation21, http://www.gen21.com

HOLO-E, http://www.holomedia.nl

http://cleo.murdoch.edu.au/teach/guide/res/examples/course-servers.html

http://e-learning.surf.nl/edubox/home

http://www.futureu.com/cmscomp/cms_comp.html

http://www.unicon.net

http://www.utwente.nl/du/projecten/kennisdomeinen/rechten/

http://www10.org/cdrom/posters/1024.pdf

Jochems, W. 2000. Restructuring distance teaching in universities for the new millennium. EADTU Paris Millenium Conference. Wiring the ivory tower. 28-30 September 2000. Available: http://www.eadtu.nl/ivorytower/abstracts/jochems-eadtu.ppt

KOEPEL, http://www.koepel.nl

Koper, R. Educational Modelling Language: adding instructional design to existing specifications. Workshop Dokumentation "Standardisierung im eLearning" 10./11.4.2002. Universität Frankfurt/Main. Available: http://www.rz.uni-frankfurt.de/neue_medien/standardisierung/koper_text.pdf

Krämer, B.J. E-learning, Virtual University, Online learning: education embraces technology. The EASST Newsletter, Vol 3, December 2001. Available: http://www.easst.org/newsletter_und_templates/dec2001-EASST.NL.pdf

Leveraging Technology to Transform the Educational Experience. 2002. WebCT white paper Available: http://www.webct.com/products

Lotus Learning Space, http://www.lotus.com/home.nsf/welcome/learnspace

Overzicht Blackboard 5.0. CINOP testgroep teleleerplatforms, http://e-learning.surf.nl/blackboard/producttest/141, http://e-learning.surf.nl/docs/blackboard/bb5matrix-10.pdf

Prins. F. 2001. Voorstudie naar integratie van EML en Blackboard. OUNL internal report

Saba, http://www.saba.com

Sloep, P.B. De Digitale Open Universiteit Available: http://platform.leidenuniv.nl/PeterSloep1312001_files/frame.htm

Stumpp, B. Vom Nutzen und Nachteil der Standardisierung im Bereich der Lerntechnologien. kevih Workshop 16-17 May 2002. Available: http://www.iwm-kmrc.de/kevih/workshops/plattformmat/StumppkevihWS.ppt

Sunil Hazari, Ed.D, Robert H. Smith. Evaluation and Selection of Web Course Management Tools. School of Business. University of Maryland, College Park. http://sunil.umd.edu/webct/

Surf Educatie<F> elearning themasite, http://elearning.surf.nl/e-learning/home

Surf, http://www.surf.nl

The Blackboard Community Portal System™ White Paper

The Blackboard Learning System™ White Paper

TopClass, http://www.wbtsystems.com

VLE/MLE Comparison Grid. http://www.chest.ac.uk/datasets/vle/checklist.html

WebCT, http://www.webct.com

WebNet, http://www.educ.uva.nl/bvenet/elo

X.HLP, http://www.xhlp.com/index.htm

Zeibert, C. 2001 Ten Steps to Successfully Selecting A Learning Management System. An Lguide Publication. Editors L. Kent, M. Flanagan, C. Hedrick. Docent white paper. Available: http://www.docent.com/elearning/tensteps.html, http://www.docent.com/whitepaper/pdf/selecting_LMS.pdf

Appendix 2 CSCL Environments and Approaches

Scope: Approaches and E-learning platforms facilitating CSCL, e-learning communities, ...

Responsible: UNED

Contributing: OUNL

A 2.1 Overview

Current e-learning platforms support collaborative learning tasks, but they are not CSCL environments. Pure CSCL environments exist mainly in research scenarios and are applied to a particular domain, where collaboration dialogues are completely structured to a specific audience. E-learning platforms promote collaboration by providing collaborative tools and a more open interaction to their users.

However, CSCL field can be of help in specifying objectives and tasks regarding collaboration in broader frameworks like web environments for e-learning. To clarify objectives and tasks, we introduce a methodology developed by OUNL, a general framework for designing CSCL environments.

In this appendix the main features of CSCL systems and some systems already implemented in research scenarios are described. To complement this view, e-learning systems that support collaboration are also mentioned, such as BSCW and FLE3. In particular, a platform called ALF (Active Learning Framework), which is being developed at UNED, is described in deeper detail. This platform provides facilities for collaborative learning and is related to issues that are the focus of Alfanet project, such as Adaptation and Multiagent Systems.

A 2.1.1 Definition

Computer-Supported Collaborative Learning (CSCL) is focused on how collaborative learning supported by technology can enhance peer interaction and work in groups, and how collaboration and technology facilitate sharing and distributing of knowledge and expertise among community members. Partly, the inspiration for CSCL arose from the research on Computer-Supported Cooperative Work (CSCW). There have been different interpretations and suggestions for the second “C” of the acronym such as, collective, coordinated, co-operative, and collaborative, that finally seem to take more acceptation [Lakkala, 2001].

Collaborative learning has been promoted in general for a variety of reasons over many years. The Constructivism educational theory was introduced by Piaget and after reinforced with the Vigotsky's work on Social Constructivism and the John Dewey's liberal educational views [Boden, 1979; Vigotsky, 1978; Dewey-Center, 2002].

Thus successful learning is regarded as [Michaelson, 1999]:

- an *active* process
- one in which the learner constructs meaning and systems of meaning
- a reflective activity - in which 'hands on' processes inform 'mental' ones
- a social activity - learning takes place through discussion and interaction with others, not just on a one-to-one basis with a student and information
- context dependent
- often requiring a long period for assimilation
- heavily dependent on motivation

A CSCL system provides different tools to support communication. There are two interaction modes: Asynchronous interaction refers to communication with arbitrary delay, such as E-mail, News groups, HyperNews, HyperMail, Shared Workspaces, etc. Synchronous interaction refers to communication which takes place at the same time, including text-based applications such as Chat, Instant Messenger and Whiteboard software, phone calls and video conferences.

The integration and combination of different tools within the same technological environment provide mutual enrichment. Both synchronous and asynchronous communication have specific features and by combining them, educational activities can have a higher impact.

A 2.1.2 Classification Framework

Jermann, Soller and Muehlenbrock [Jermann et al., 2001] review systems that support the management of collaborative interaction, and propose a classification framework built on a simple model of coaching. Some of the more relevant issues are the following.

Collaboration management follows a simple homeostatic process that continuously compares the state of interaction with a target configuration. Actions are taken whenever a perturbation arises, in order to bring the system back to equilibrium.

Systems that support collaboration generally adopt one of the following approaches:

- The first approach, structures the situation in which the collaboration takes place. Learning situations can be structured by requiring the students to use a set of structured software tools, structuring the group itself or structuring the task. These factors may encourage group members to engage in certain types of interaction such as argumentation or peer tutoring via external means.
- The second approach involves structuring the collaboration itself through coaching or self regulation. As the collaboration progresses, the state of interaction is evaluated with respect to a desired state, and remedial actions may be proposed to reduce discrepancies between these states. Structuring and coaching are not exclusive approaches, as structuring interaction might take place during interaction as a remedial action.

Two approaches to guiding learning interaction can also be distinguished:

- In the first case, the system gathers data about the students' interaction, and shows some visualization of this information to the user, possibly aside information about what an ideal interaction might look like. It is then up to the students or teacher to interpret the visualization and decide what actions (if any) to take.
- In the second case, the model of interaction and the system's assessment of the current state is hidden from the students. The system uses this information to make decisions about how to moderate the group.

Fundamentally, these two paradigms are the same, in that first data is collected, then a model of interaction is constructed and instantiated to represent the current state, and possibly the desired state, and finally, some decisions are made about how to proceed. The difference between these two approaches lies in the locus of processing. Systems that collect interaction data and construct visualizations of these data place the focus of processing at the user level, whereas systems that advise process this information directly.

The benefits of coaching student interaction (via human or computer) are clear, given a correct diagnosis and appropriate remedial actions. Students who view and analyse indicator values may learn to understand and improve their own interaction. Students might, however, lack the understanding to interpret the visualizations correctly, leading them to take unnecessary actions. Without the time and understanding to develop their own models of interaction, students may naturally rely on implicit social norms (status, equality) to manage the interaction. Collaborative learners, guided by indicator displays, may need to follow a more introspective process to develop an understanding of their interaction than when they are guided by an advisor.

The framework distinguishes between three types of supportive collaborative learning systems.

- Systems that reflect actions, termed mirroring systems, collect raw data in log files, and display it to the collaborators.
- Systems that monitor the state of interaction, termed metacognitive tools, model the state of interaction and provide collaborators with visualizations that can be used to self-diagnose the interaction. These visualizations typically include a set of indicators that represent the state of the interaction, possibly alongside a set of desired values for those indicators.
- Finally, coaching or advising systems guide the collaborators by recommending actions students might take to improve their interaction.

A 2.1.3 Design Recommendations for CSCL

Strijbos and Martens [2001a] describe a model with guidelines to classify CSCL in terms of pedagogy design for face-to-face, as well as, for computer supported learning environments, identifying the key elements for a process-based classification of computer supported groups for educational purposes.

In this framework, group-based learning can be differentiated along three dimensions closely related to group interaction and performance. Each dimension refers to a key element for group-based learning design: *level of pre-structuring* (high or low level)*, type of learning objectives* (open versus closed skills) and *task type (*well-structured tasks with limited solutions versus ill-structured tasks) [Strijbos and Martens, 2001b].

Apart from these three key elements, two additional elements can be identified that appear essential for the design of group-based learning: *group size* and *type of computer support*. Group performance effectiveness depends, as size increases, on the one hand on the groups' use of increased resources and alternate opinions and on the other hand on the handling of increased co-ordination and group management processes [Shaw, 1981; Saavedra, Earley, & Van Dyne, 1993].

A distinction between modes of communication can also be made [Rafaeli and Sudweeks, 1997]: *one-way* communication (or Asymmetrical interaction), *two-way* communication (or Reactive interaction), *interactive* communication (Reciprocal interaction) and *networked* communication (or Networked interaction).

A fifth element for group-based learning design is the chosen type of computer support for education. Crook [1998] distinguishes three categories: interaction *with* computers (individual student interaction with a computer), interaction *at* computers (a group of students interacting with a computer) and interaction *through* computers (interaction between group members via computers).

In general, group-based learning design should start with a clarification of the design, or more specifically, whether the combination of chosen 'learning objectives', 'task-type', 'level of pre-structuring interaction', group size and the type of 'computer support', will elicit the type of interaction envisioned.

Six steps can be identified for group-based learning design [Strijbos and Martens, 2001a]:

1) Determine the learning objectives
2) Determine the envisioned interaction
3) Select the task-type with respect to the learning objectives
4) Determine whether and how much interaction will be pre-structured in advance
5) Determine which group size is best suited with respect to choices already made
6) Determine how computer support can be deployed best to support learning.

However, a set of design rules that will guide designers to the most optimal combination of elements in a group-based learning setting is at this moment a bridge too far. Designers can only be assisted in 'thinking through' their design. A tentative evaluation schema has been proposed by the OUNL research team [Strijbos and Martens, 2001a]. The object of this scheme is to provide support in designing and assessing group-based learning through explicit reasoning about the combination of key elements and the extent to which that combination facilitates the actual occurrence of the envisioned group interaction.

A 2.1.4 Current Trends

The current research on CSCL on Europe [Lakkala, 2001] says that:

- It is focused on: technical design principles and their educational implications, the nature of processes of communication, collaborative inquiry and knowledge building in network based environments, motivation, engagement and participation, as well as the role of the teacher or tutor.
- Face-to-face collaboration is combined with collaboration that takes place in a network-based learning environment within the classroom.
- Collaborative technologies are shown to enhance student motivation, self-reflection, working with complex problems, and promote collaboration between learners.
- There are, also, strong cultural constraints on the level of teachers and, indeed, the learners themselves.

A 2.2 Solutions

After introducing some research systems of current literature, we include some samples of commercial web systems with special relevance in supporting CSCL. Next, we describe in detail ALF, as example of an e-learning platform that support collaborative work and has been developed by UNED. This platform provides facilities for collaborative learning and is related to issues that are the focus of Alfanet project, such as Adaptation and Multiagent Systems.

A 2.2.1 Research Systems

The study of Jermann [Jermann et al., 01] provides a summary of systems that support collaborative learning, classified according to their structure.

Some systems that monitor the state of interaction are:

- HabiPro [Vizcaino, Contreras, Favela and Prieto, 2000] is a collaborative programming environment that both displays the students' participation statistics, and models more complex interaction variables. The system includes a group model, and an interaction model, which includes a set of "patterns" describing possible characteristics of group interaction (e.g. the group prefers to look at the solution without seeing an explanation). During the collaborative activity, the group model compares the current state of interaction to these patterns and proposes actions (such as withholding solutions until the students have tried the problem).
- EPSILON [Soller and Lesgold, 2000] monitors group members' communication patterns and problem solving actions in order to identify situations in which students effectively share new knowledge with their peers while solving object-oriented design problems. First, the system logs data describing the students' speech acts (e.g. Request Opinion, Suggest, Apologize) and actions (e.g. Student 3 created a new class). In the second phase, the system collects examples of effective and ineffective knowledge sharing, and constructs two Hidden Markov Models which describe the students' interaction in these two cases. A knowledge sharing example is considered effective if one or more students learn the newly shared knowledge (as shown by a difference in pre-post test performance), and ineffective otherwise. In the third phase, the system dynamically assesses a group's interaction in the context of the constructed models, and determines if the students need mediation.

Two systems that offer advice are:

- Barros and Verdejo's [2000, 2001] asynchronous newsgroup-style system, DEGREE, requires users to select the type of contribution (e.g. proposal, question, or comment) from a list each time they add to the discussion, identifying the social aspects of interaction. The system's model of interaction is constructed using high-level attributes such as cooperation and creativity (derived from the contribution types mentioned above), as well as low-level attributes such as the mean number of contributions. Next, the system rates the collaboration between pairs of students along four dimensions: initiative, creativity, elaboration, and conformity. These attributes, along with others such as the length of contributions, factor into a fuzzy inference procedure that rates students' collaboration on a scale from "awful" to "very good". The advisor in DEGREE elaborates on the attribute values, and offers students tips on improving their interaction.
- GRACILE [Ayala and Yano, 1998] is an agent-based system designed to help students learn Japanese. The system maintains user models for each of the students, and forms beliefs about potential group learning opportunities. Group learning opportunities are defined as those that promote the creation of zones of proximal development [Vygotsky, 1978], enabling a student to extend his/her potential development level. GRACILE's agents assess the progress of individual learners, propose new learning tasks based on the learning needs of the group, and cooperate to maximize the number of situations in which students may effectively learn from one another.

A 2.2.2 Web-based Commercial Systems

BSCW – (Basic Support for Cooperative Work) [GMD, 2002]

The BSCW Shared Workspace System is an Internet /World Wide Web (WWW) based groupware system. It is an extension of a standard Web server through the server CGI Application Programming Interface. The central metaphor of the system is the shared workspace. A BSCW server (Web server with the BSCW

extension) manages a number of shared workspaces, i.e., repositories for shared information, accessible to members of a group using a simple user name and password scheme.

This system is analysed in more detail in D72 - Market Analysis

FLE3

Fle3 [Fle3, 2002] is a web-based learning environment, designed to support learner and group centered work that concentrates on creating and developing expressions of knowledge (i.e. knowledge artefacts). We consider specially interesting the system dialogue structure as well as their compatibility with EML.

Fle3 contains three learning tools and several administration tools.

Fle3 WebTops can be used by teachers and students to store different items (documents, files, links, knowledge building notes) related to their studies, organize them to folders and share them with others. WebTop also includes shared "course folder" for each course. The same shared "course folder" is available in the Knowledge Building and Jamming tools as well. The items in the WebTops can be called learning objects - if you wish.

With Fle3 Knowledge Building tool groups may carry out knowledge building dialogues, theory building and debates by storing their thoughts into a shared database. In the Knowledge Building study group may use Knowledge Types to scaffold and structure their dialogues. The Knowledge Type sets are fully editable and one may export and import them from one Fle3 to another. Fle3 comes with two default Knowledge Type sets: (1) Progressive Inquiry, and (2) Design Thinking.

Fle3 Jamming tool is a shared space for collaborative construction of digital artefacts (pictures, text, audio, video). A study group may work together with some digital artefacts by simply uploading and downloading files. Versions are tracked automatically and different versions are displayed graphically. Jamming can be used for many kind of collaborative work requiring versioning.

For teachers and administrators Fle3 offers tools to manage users and courses / study projects. The administrator may also export and import courses or the fully content of the Fle3 database in XML format (compatible with the Educational Modelling Language - EML).

A 2.2.3 aLF: a web-based collaborative framework

Platforms should provide the necessary services that allow users to communicate with one another and share information. Merely focusing on the information in the web logs wil just provide with information on the pages accessed, but it will not give information about the interactions with the rest of the system. Consequently, in order to obtain this additional information, the platforms must fulfil the following requirements [Gaudioso and Boticario, 2002b]:

- Information should be stored on the user's actions with any system service
- Related to the previous point, user data should be accessible, for example, personal data, responses to surveys, configuration data (permits on personal files, group membership, ...)
- This information should be stored in the most structured possible way to facilitate the subsequent data preprocessing

Therefore, in order to achieve adaptation tasks regarding collaboration, it is necessary to have certain user interaction data that provide more information than web logs. In the beginning, the main problems within the web-based adaptive systems was data gathering on users. However, nowadays, more and more systems are able to monitor user actions and record complete user interactions. But if a system gathers a great amount of data, its performance might be adversely affected. Thus it is important to determine which data is really necessary to provide a good adaptation of the user without slowing down the system [Gaudioso and Boticario, 2002b]. A platform that integrates both the data gathering and an access and efficient management of the usual Internet communication and collaboration resources is aLF [Boticario, Gaudioso and Catalina, 2001], which is describe next.

A 2.2.3.1 ALF

aLF (**a**ctive **L**earning **F**ramework) is a platform that has been developed at UNED in order to support the requirements for collaborative work [Gaudioso and Boticario, 2002a]. The platform builds upon the ArsDigita Community System [ACS, 2002], a multiplatform and open source set of tools for constructing web-based applications. aLF is composed of a Web server connected to a relational database and a set of TCL (*Tool Command Language*) scripts allowing management of the interaction with the data model.

The first important feature of aLF is using a relational database to manage the information provided by the web server. This information is usually managed directly through the filesystem tools provided by the operating system. Although this approach can provide adequate functionality for static sites, it is neither flexible nor reliable enough when it has to manage large amounts of transactions. In a collaborative environment like aLF, there is a huge stream of information flowing between users and through the site. These sites are very dynamic since users are regularly interacting in a variety of ways, such as sending messages to the bulletin boards, organizing interactive meetings or publishing materials. Accordingly, the relational database component of aLF provides support for these requirements. This database does not only store the user personal information and the contents they send, but also serves to structure potentially everything that is happening in the site.

Another key feature of aLF is that it offers different views based on user navigation and actions through the site. All kinds of information on users can thus be gathered with TCL scripting language. The scripts can interface with the database so that this information can be easily stored and retrieved on demand. TCL scripts can also contribute to maintaining a dynamic environment, since they enable web pages to be dynamically constructed and this process can make use of the database information if required.

aLF services

To benefit from the features provided by aLF, users have to register. Users registered on aLF are grouped into *workgroups* (see Figure 1). The administration of each workgroup is done by the person in charge who may not be the web site manager. The setting up of these kind of workgroups is particularly useful for distance learning since it allows different university departments, courses, study groups and research groups to be managed.

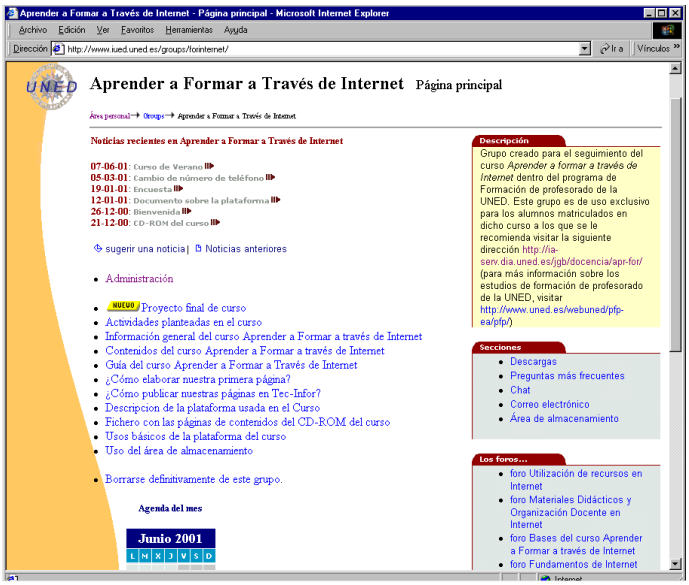


Fig. 1. aLF course workgroup

aLF workgroups offer several services thus allowing a learning and collaborative environment to be easily set up:

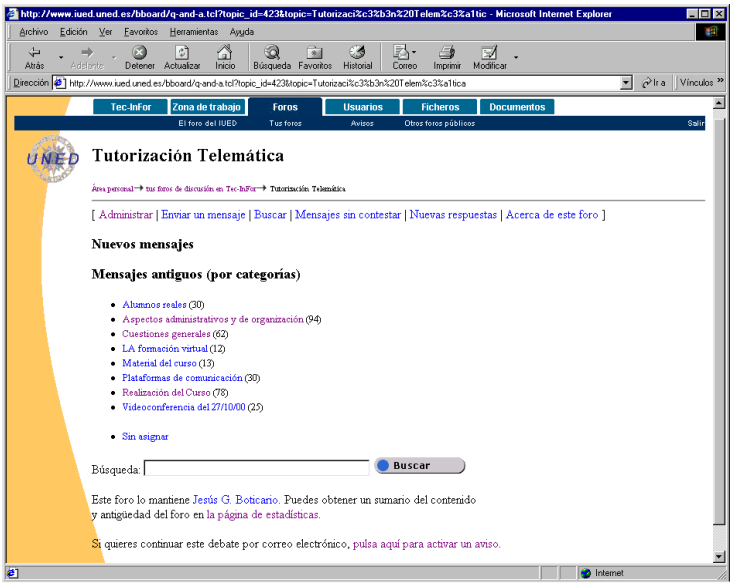


Fig. 2. *An example of newsgroups pages*

- ***Newsgroups*****:** They can be reached via HTML pages (see Figure 2), automatically index the messages by categories defined by the manager or establish a notification system so users are automatically informed about new replies. They can be moderated or unmoderated. They are very useful in an educational community because they permit communication between students and lecturers.
- ***Bulletin boards:*** With this service aLF users can create news for the rest of the community members without having to use electronic mail. The author of the news item or message can control the date of publication of the news, its expiry date and even the group of people for whom it is intended. It is particularly useful since the rest of the workgroup members can comment on the news item and group interaction is thereby enhanced (see Figure 3).

Fig. 3. *An example of an aLF bulletin board*

- ***Document management:*** A user can manage all his/her documents on line by having them in a central place and accessible to all those users that he/she desires. The system has a permission management service so that each document can be read, modified or managed by a particular user, by a series of users or by a whole workgroup. These users can access from anywhere with a browser to see, update or delete their files. Furthermore, we can know the history of the document and thus see who has worked on it and the date when the changes were made. This is particularly useful since it enables development in workgroups suggested by the lecturer; with this version control it is possible to see which student has worked on which part, how they have worked together, etc. (see Figure 4).
- ***Chats:*** Internet group work is useful particularly from an asynchronous point of view, i.e., when the different collaborators do not coincide at the same time when they do their work. Sometimes distance and the feeling of isolation can be counterproductive. With the chat tool small group chats

can be organized, thereby avoiding the problem of overcrowded chats which make the service unintelligible. The advantage of this service in aLF, as well as gaining access via the HTML pages, is that all the conversations are recorded so that the lecturer can select relevant information from his/her conversation with the students and thus avoid repetition. Key chat issues can be transferred to the forum.

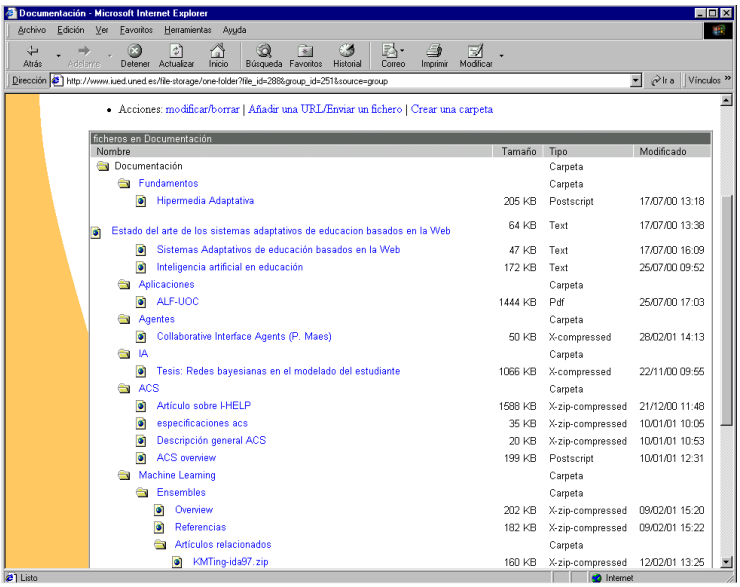


Fig. 4. *Document Management in an aLF workgroup*

- ***Calendar:*** There are private calendars for each of the users, group calendars and a public calendar for all the aLF communities. The most commonly used service is the fixing of appointments within the workgroup.
- ***Presentations:*** They allow, by just completing forms, to prepare an instant presentation for the Web which is quickly downloaded. There is therefore no need to send files to the server. They have automatic indexes and all the necessary links for navigation support. They are really useful for teaching since they allow both lecturers and students to specify certain contents on a few screens, each screen being a HTML page. They also have a further advantage because you do not need to know the web format to edit the screen with a pleasant and navigable visual structure.
- ***Project management:*** It allows certain project tasks to be created. For each task, project or tasks assignments can be created which become the responsibility of the task owner. These assignments can be different types (error, improvement or new task), have different levels of severity (critical, serious, medium or low) and states (open, need clarification, fixed waiting approval, deferred or closed). From the point of view of distance learning, the usefulness of a project management tool as presented here is obvious. We can create projects for certain student workgroups and tasks for these projects, so that students work in their corresponding area as they receive assignments, either from their lecturer or their fellow students. The group members can check their work on a list where all the group member tasks are described, as well as progress reports on each one of them. Each project can have forums, project marks and a task and assignment monitoring system, thereby creating a user-friendly project knowledge base which can be consulted and accessed from the web.
- ***Tools to contact the other users:*** There are a set of different tools enabling users to publish their own personal pages, to see which users are connected at the same time, to see which users are already in the community and what their participation is. This utility promotes the use of the chat tool.

It is important to point out that aLF manages all these services through the database. Information about the messages sent to a forum or a bulletin board, the task assignments (tickets) sent to the project management tool, the appointments on the calendar, the annotations to a particular news item on the board or the conversation in a chat are stored in the database, so a great amount of information on user interaction with the platform is available.

A 2.2.3.2 An aLF course experience

We will now describe an experience of a distance course being taught through aLF. The goal of this course was precisely to teach the use of the Internet in education.

We set up a group in aLF for the students on the course; this group had a bulletin board, news service, chat, file-storage area and a personal web-pages storage area where the students could publish their own web pages.

Regarding the conceptual organization of the contents, it was based on the proposal made by Roger Schank in the ASK system [Schank and Cleary, 1995]. The course concepts were organized as a conceptual network where each node was a course concept for the student to learn and the arcs represented certain links between concepts such as, prerrequisite, consequence, example or exercise. These links were presented by several questions or options that the student could choose (see Figure 5 and 6).

As we have mentioned before, the whole goal of the course was to teach the use of the Internet in education for lecturers. It started by introducing basic concepts about the Internet and its services. It then taught a possible organization of the courses using these services.

The course also had certain general subgoals that the student may attain, such as *Basis of the course*, *Fundamentals of the Internet*, *Use of the Internet in education* and *Practices*.

The course contents were represented by a set of HTML pages, each one corresponding to one concept in the course conceptual network. This conceptual network, the goals of the course, and other elements are represented by an XML page that the system may interpret to construct the student knowledge model.

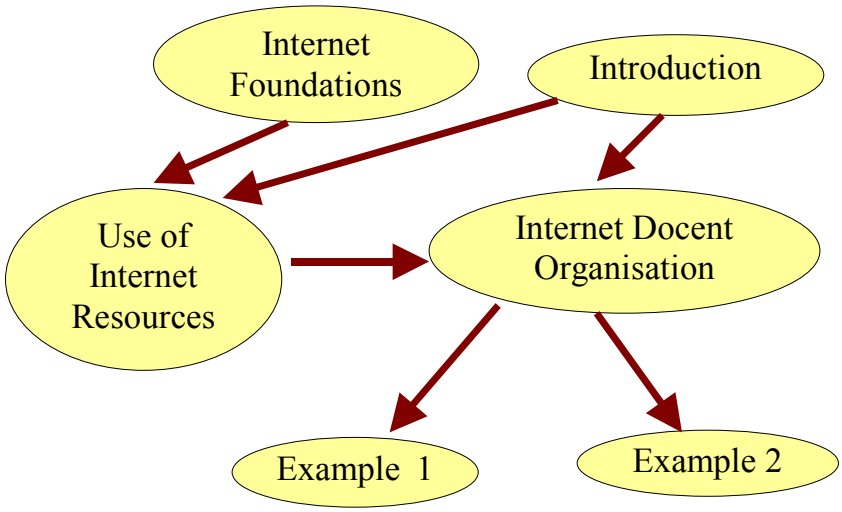


Fig. 5. Partial conceptual network of the course contents

For each student on the course, the system stores a model of the contents that the student has already learned. For this course the system assumes that a student has learned a concept if he/she has successfully done a test or has simply visited the pages related to the concept.

First we presented the students with a survey in order to evaluate how familiar they were with educational software or the Internet. In most cases the students were in turn lecturers interested in the use of the Internet in education, and on the whole, they had little experience in the use of computers and Internet services. However, as the course proceeded we realised that we had a very hetereogenous group of students.

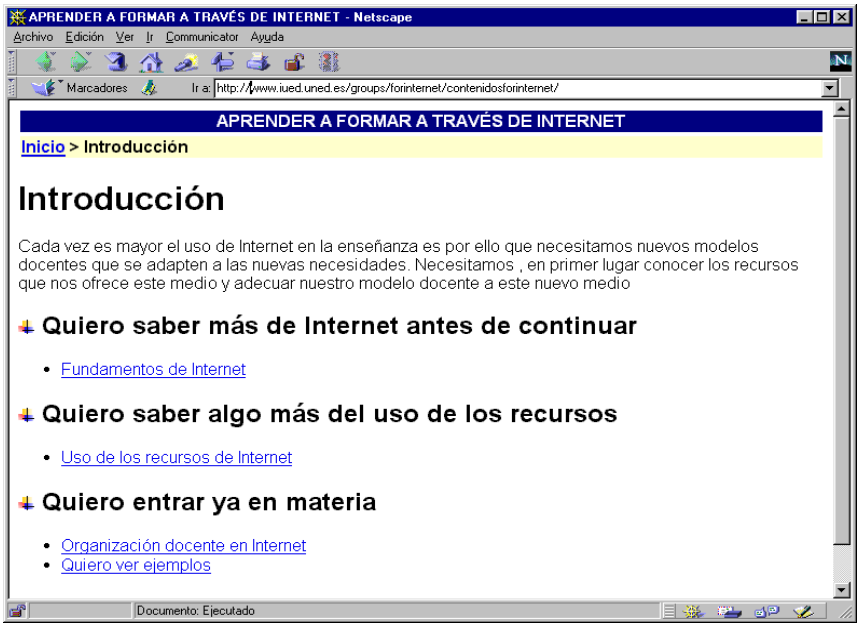


Fig. 6. *Course page resulting from the conceptual network of Figure 5*

Consequently, the course lecturers had to closely monitor the students on the course. They solved (on demand) the difficulties that arose. The lecturers proposed several activities (such as sending a message to the bulletin board) in order to guide the students more closely.

Although there was also the possibility of contacting the lecturers by phone or face-to-face, the main channels of communication were the course bulletin boards and electronic mail. The lecturer also visited the chat room established for the course to detect if the students had difficulties in the course.

We have also discovered that students do not collaborate among themselves except when they use the chats and send personal e-mails. If this lack of collaboration could have been foreseen, lecturers could have suggested scenarios or tasks where students were encouraged to interact.

These problems could have been avoided if the system had detected student difficulties, such as problems in the use of the course services and student similarities. Then lecturers could have been informed of these problems right from the very beginning and they could have set up subgroups of students with similar characteristics, needs and interests. The system could also automatically do this.

Thus our proposal for a personalized environment is based on a cooperative tutorization scheme. In this scheme the lecturer, besides solving the doubts of the students in a personalized way, is responsible for improving collaboration between the group components.

A 2.3 Assessment

A 2.3.1 Conclusions for Alfanet

CSCL is a vast field of research. Moreover, it can be used for other purposes than supporting collaboration, such as for transmitting and delivering knowledge.

The aim of the system to be developed in Alfanet Project is not a particular CSCL platform, but has to provide some features related to CSCL. Therefore, ideas from CSCL can be taken into account when designing the system for Alfanet. CSCL environments have a very ambitious approach, regarding just collaborative issues, whereas Alfanet project has a broad approach which is intended to provide an interactive, adaptive and personalised e-learning framework.

In this appendix we have mentioned some key points to be considered in the Alfanet design.

- To include synchronous and asynchronous tools to support communication, and shared workspaces to promote collaboration
- To include some method to structure the communication

- To collect learner interaction data and to create a model of interaction
- To provide collaborators with indicators that represent the state of the interaction
- To advise collaborators by recommending actions to improve their interaction

CSCL research field can be of help in specifying objectives and tasks regarding collaboration in broader frameworks, like Web environments for e-learning. This technology is intended to solve the challenges of collaborative scenarios, but the Web environment adds complexity to a traditional CSCL system.

In this sense, it can be very useful to consider OUNL methodology for designing CSCL environments. This framework provides a model to define and describe the CSCL key elements that Alfanet should support. It can be used as a starting-point for developing functionality for the Alfanet system, which is intended to support a dynamic and adaptive collaboration framework. Therefore, it is desirable to start the development of Alfanet taking as a basis an existing system that provides collaboration facilities.

A 2.4 References

ACS (2002), ArsDigita Community System: http://www.arsdigita.com;

http://www.redhat.com/software/ccm/community/

Ayala, G, and Yano, Y (1998). A collaborative learning environment based on intelligent agents. *Expert Systems with Applications*, 14, 129-137.

Barros, B., and Verdejo, M.F. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education*, 11, 221-241.

Barros, B., and Verdejo, M.F. (2001) Entornos para la realización de actividades de aprendizaje colaborativo a distancia *Revista Iberoamericana de Inteligencia Artificial*. ISSN 1137-3601

Boden, M. A. (1979) Piaget. Fontana modern masters series, Fontana, London

Boticario, J.G., Gaudioso E., Catalina, C. *"Towards personalised learning communities on the Web"*. First European Conference on Computer-Supported Collaborative Learning, pp. 115-122, 2001.

Dewey-Center (2002) The Center for Dewey Studies, http://www.siu.edu/~deweyctr/

Crook, C. (1998). Children as computer users: the case of collaborative learning. *Computers & Education*, *30(3/4)*, 237-247.

Fle3, 2002 : http://fle3.uiah.fi/index.html

Gaudioso E., Boticario, J.G. *"Supporting personalization in virtual communities in distance education"*. To appear in "Virtual Environments for Teaching and Learning" to be published by World Scientific Publishing Company Pte Ltd, 2002a.

Gaudioso E., Boticario, J.G. *"An ensemble of classifiers approach to usage modeling on adaptive learning communities"*. To appear in "IEEE Transactions on Knowledge and Data Enginieering (TKDE), 2002b.

GMD, 2002 : http://bscw.gmd.de/

Jermann, P., Soller, A. and Muehlenbrock M. (2001) From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning

Michaelson, R. (1999) Web-based Group Work, *Proceedings of the10th Annual CTI-AFM Conference*, 58-64, August, CTI-AFM Publications, East Anglia.

Lakkala M., Rahikainen M. and Hakkarainen K., (2001), D2.1 Perspectives of CSCL in Europe: A Review, ITCOLE Project - IST-2000-26249,

Rafaeli, S., & Sudweeks, F. (1997). Networked interactivity. Journal of Computer Mediated Communication, 2 (4). Retrieved July 19, 1999, from http://www.ascusc.org/jcmc/vol2/issue4/rafaeli.sudweeks.html

Saavedra, R, Earley, P.C., & Van Dyne, L. (1993). Complex interdependence in task- performing groups. *Journal of Applied Psychology*, *78*, 61-72.

Schank, R.C. and Cleary, C. (1995) *Engines for education*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Shaw, M.E. (1981). *Group dynamics: the psychology of small group behavior (3rd ed.).* New York: McGraw-Hill.

Soller, A., and Lesgold, A. (2000). Knowledge acquisition for adaptive collaborative learning environments. *Proceedings of the AAAI Fall Symposium: Learning How to Do Things*, Cape Cod, MA.

Strijbos, J.W. & Martens, R.L. (2001a). Structuring group-based learning. Paper presented at the EARLI Conference, August 28th - September 1st, Fribourg, Switzerland.

Strijbos, J.W., & Martens, R.L. (2001b). Group-based learning: dynamic interaction in groups. Proceedings of the first european conference on computer supported collaborative learning (pp. 569-576), 22-24 March, 2001, Maastricht, The Netherlands. Retrieved August 22, 2001, from http://www.mmi.unimaas.nl/euro-cscl/Papers/154.doc

Vizcaino, A., Contreras, J., Favela, J., and Prieto, M. (2000). An adaptive, collaborative environment to develop good habits in programming. *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, Montreal, Canada, 262-271.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. London: Harvard University Press.

Appendix 3 Adaptive educational systems and approaches

A 3.1 Introduction

The purpose of educational systems in general is to provide information to learners to help them to reach certain learning goals, that is, to help learners to learn certain skills and/or to increase their knowledge about a certain topic. The term information is used here in its broadest sense, so it also includes more directive information like tasks to perform, feedback, assignments, questions, etc. Different learners might have different needs, characteristics, prior knowledge, etc., which might require the presentation of different information to different learners or in a different format. It is the purpose of adaptive educational systems to take these aspects of individual learners into account when presenting information in order to make the learning process as effective, efficient, and motivating as possible.

In the next section we will first provide a brief historical overview of the research that lead to the development of adaptive educational systems. Next we will describe three examples of state of the art solutions for making educational systems adaptive. Finally we will discuss the consequences for the Alfanet system.

A 3.2 Historical Overview

A 3.2.1 CAI: Computer Aided Instruction

The use of computers for educative purposes started in the 50's, with systems used for training and practicing. These systems were no more than an evolution of books, using traditional teaching methods to teach the contents already structured, plus some representation assistance. No flexibility neither dynamism existed, and therefore the contents could not be modified on the fly. The systems took advantage of teachers experience to advance the behaviour of the program, but it has been proved impossible to design a program that encode all the possible decisions to be taken.

A 3.2.2 ITS: Intelligent Tutoring Systems

ITS are the first systems that use Artificial Intelligence techniques to build educational systems. There is no difference with CAI regarding teaching methods nor learning philosophy, since the tutor controls the system and the student learns through short questions. However, the knowledge and reasoning of the human tutor is represented in the system, and as a consequence, ITS can instruct in a more detail way that CAI systems. There exist two new elements in ITS, the student model, that is, what the system supposes the student has learnt, and the pedagogical model, the knowledge to manage the learning process, with rules encoding the experience regarding the tutoring process itself. This allows a one-to-one tutoring. The system selects the task and problems to be done by the student, and decides when the student has to provide some kind of feedback.

The core of the ITS is an expert system that has enough knowledge about a particular area to provide ideal answers to questions and correct not only the final result, but also each reasoning step to the final result. This allows to model the correct way to solve a problem and therefore it is possible to monitor the student while solving the problem. When a student makes a mistake, the ITS tells the student how the expert system would have performed in that step. CAI systems used the question as the minimal unit, but ITS use each individual reasoning step, so a more detailed feedback is achieved, which leads to a better diagnosis of errors, and therefore a faster learning.

The developers of ITS believe that a competent tutor is in a better position to take decisions about what the student needs to learn effectively than the student him/herself. The system knows the contents the student must learn and the particular status of the student knowledge at any time through the expert system and the student model. Therefore, the student is provided with a limited number of choices, being the software the one that selects the next task to be done, gives assistance to the student, asks for feedback and decides the nature of the information provided.

These systems have only succeeded in areas where it is easy to analyse all the possible tasks to be done and the possible errors to be made. Their limitations are due to poor pedagogical components and that there is only one teaching/learning model; the systems do not get all the practical knowledge of the expert, they have only been built to develop a task, not to teach or explain the subject.

A 3.2.3 ILE: Interactive Learning Environments

These systems use a constructivist approach rather than the behavioural approach used in ITS. This means that the intelligence does not rely on the tutor who acts to guarantee that what the student learns corresponds to the expert model, but these systems provide tools to encourage students enquiry without an external control. Students do not follow an organized sequence of exercises and practices, but they build their own knowledge. The difficulties of these systems are that the process of acquiring the knowledge is rather inefficient and that it is very difficult to define the learning goals.

A 3.2.4 LAS: Learning apprentice systems

Learning apprentice systems are programs that help build rule-based expert systems semi-automatically by generalizing the training examples provided by the users. They provide a very flexible way to acquire new knowledge and overcome the problem of knowledge acquisition bottle-neck faced by many expert systems. One of the major advantages of LAS is in helping the system to gain autonomy.

A very comprehensive definition of LAS is given by Mitchell: "We define learning apprentice systems as the class of interactive knowledge based consultants that directly assimilate new knowledge by observing and analysing the problem solving steps contributed by their users through their normal use of the system" (Mitchell, 1985).

A prototype of learning apprentice is CAP (Dent, Boticario, McDermott, Mitchell, and Zabowski, 1992) which acts as a personal assistant for managing the meeting calendar of a university faculty member.

A 3.2.5 Hypermedia Systems

Hypermedia systems came up in the late 80's and allowed to develop complex environments oriented to content. The learning was contextualised, that is, the computer became to be used as a communication tool to relate students with real contexts. Hypertext is a way to organize non-lineal information, connecting nodes with links in an associative way, that can be obtained interactively by browsing. These systems are information systems that facilitate the retrieval of information, and can be used for learning if they are design to reach some specific learning goal and its achievement is assessed. The main features are their high interactivity, the use of large data bases and multimedia information and the knowledge representation similar to the human mind. However, the simple exploration is not a learning goal, and sometimes a guided learning is needed.

A 3.2.6 Collaborative Learning Systems

They became really useful when computers were connected to networks and were used for collaborative learning. As a consequence, some issues were taken into account, such as user modelling, adaptive interfaces, learning through the web and courses and contents standardization. These systems provide collaboration recommendations by comparing the actions taken in the shared space with those in the private one according to some predefined rules. A student model is needed to monitor his/her interaction, but also a group model is needed to monitor the actions done together with the other students. There are three kinds of these systems: those that only show the actions done in log files, those that monitor the state of the interaction, by comparing it with an ideal model, and those that analyse the state of the collaboration, giving advise to improve the learning.

A 3.2.7 Internet and Teaching/Learning Interactive Systems

Internet provides new facilities and learning methods, since information is independent of time and place, and therefore, an unlimited number of students can use any kind of information, easily updateable, and taking their own speed. However, there is a bottleneck in the limited capacity of the teacher to supervise the learning. Moreover, the needs of the collaborative learning model are different (although currently the attention is not focused on this issue and the classroom concept is taken directly to the Internet environment, with all the problems associated). Both of these reasons had led to the development of virtual educative communities. These are groups of people with different levels of experience and common interests that use Internet resources to communicate and coordinate with the rest of the community. These systems provide technical resources both for the teacher, to become an information facilitator, a critical analyst of knowledge, a study guide and a reviser and assessor of the learner education, and for the learner, to take an active role in the learning process as member of these virtual community. Contents are no longer the most valuable part of learning, but the guidance of proficient teachers through the working material and the social experience of sharing the learning process with other students. However, the use of the resources depends

on the user experience, so the systems has to provide suitable help. All these has lead to the development of Educational Adaptive Systems on the Web, based on techniques used on TIS, tools and field models of hypermedia systems, general machine learning ideas and specific learning systems ones.

A 3.2.8 Adaptive Hypermedia Systems

These systems are based on hypertext and hypermedia and use a user model to adapt the system to the particular needs of each user, mainly information and links, and to limit the navigation space. Together with World Wide Web provide personalized access to a huge amount of information from any where and at any time. They are made up of hyper documents connected by links, and both content of the pages and links can be adapted to the user. These systems also provide help for course and virtual communities management, such as detecting students needs, finding students with similar affinities, putting in touch people with the same interests, ... creating an environment where group of students learn under the tutor supervision, who manage and leads the cooperation, giving adequate goals for the progress of each individual student.

A 3.2.9 Web-based Adaptive Educational Systems

Web-based Adaptive Educational Systems (AES) are not an entirely new or unique kind of systems. Historically, Web-based AES inherit from two earlier kinds of AES: intelligent tutoring systems (ITS) and adaptive hypermedia systems. Traditionally, problems of developing AES were investigated in the area of intelligent tutoring systems (Burns and Capps, 1988). ITS use the knowledge about the domain, the student, and about teaching strategies to support flexible individualized learning and tutoring. Adaptivity was one of the goal features of any ITS. Adaptive hypermedia is a much newer research domain (Brusilovsky, 1996). Adaptive hypermedia systems apply different forms of user models to adapt the content and the links of hypermedia pages to the user. Education is one of the main application areas for adaptive hypermedia and a number of adaptive educational hypermedia systems was built before "Web Rush". From a systemic point of view current Web-based AES can be considered simply as ITS or adaptive educational hypermedia systems implemented on the Web. However, WWW context provides serious impact on design and implementation of these system and let us treat them as a special subclass. For example, very few standalone ITS use adaptive hypermedia, while almost all Web-based AES can be classified as both ITS and adaptive hypermedia systems. It is the impact of a "hypertext" nature of the Web.

A 3.3 Solutions

A 3.3.1 ADAPTit: the TASKi module

A 3.3.1.1 Introduction

TASKi is a system for the adaptation of a training program to the learning capacity of individual learners in order to optimise the process of learning a complex cognitive skill. It is currently being developed in a EU funded project called ADAPTit (De Croock, Paas, Schlanbusch, & van Merriënboer, 2002). Because learners differ with respect to their learning capacity and abilities, some learners might need more practice and more support to master a skill, whereas others might need less. TASKi can estimate the rate of skill development for a particular learner and use this information to optimise the cognitive load the training program imposes on a learner. The training program induces an optimal cognitive load if the learner perceives a load that is not too high but also not too low. If the cognitive load is too high, the learner is not able to perform the learning task, gets frustrated and might give up learning. If the load is too low, no optimal use of the learners' cognitive resources is made which might demotivate the learner. The assessment of skill development is based on a learner's performance and on a learner's mental effort expenditure on learning tasks. Based on these measures, after each learning task an estimation of the learners' current level of competency is made. The competency level is then used as a basis for selecting the next most optimal learning task. The most important dimensions on which the training program can be adapted to a learner are: 1) The amount of learner support that is provided while the learner is working on a learning task and 2) the rate in which the complexity of learning tasks that are presented to a learner increases. The system is generic, which means that it can be used for training programs for complex cognitive skills in various domains, however it only functions with training programs that are fully compliant with the 4C/ID methodology.

A 3.3.1.2 4C/ID training design

Before TASKi can be deployed a designer has to make a detailed analysis of the to be trained complex skill and identify all the constituent skills that are involved in performing the complete complex skill. These constituent skills are modelled in a so called skills hierarchy. In this hierarchy for each skill also standards of successful performance are specified. Next, the designer develops a training program according to the 4C/ID methodology. Such a program consists of four basic components (see Figure 1): (1) Learning tasks (represented as circles), which are the actual tasks the learners will be working on during the training program. Learning tasks are organized in a simple-to-complex sequence of task classes (the dotted boxes around a set of learning tasks), that is, categories of equivalent learning tasks. Learning tasks within the same task class start with high build-in learner support(indicated by the dark filling of the circles), which decreases and disappears well before the end of the task class; (2) Supportive information (represented as L-shaped, light grey figures), which is helpful to the learning and performance of aspects of the learning tasks that require variable performance over problem situations. It explains how a domain is organized and how to approach tasks or problems in this domain, and provides cognitive feedback (CFB) on the quality of task performance; (3) Just-in-time information (represented in the dark grey rectangles, with upward arrows), which is information that is prerequisite to the learning and performance of aspects of learning tasks that show invariant performance over problem situations. It provides algorithmic specifications of how to perform those aspects; and (4) Part-task practice (represented by sequences of small circles), which provides additional repetitive practice for selected constituent skills that need to be performed at a very high level of automaticity after the training. It is only necessary if the learning tasks do not provide enough repetition to reach the desired level of automaticity.

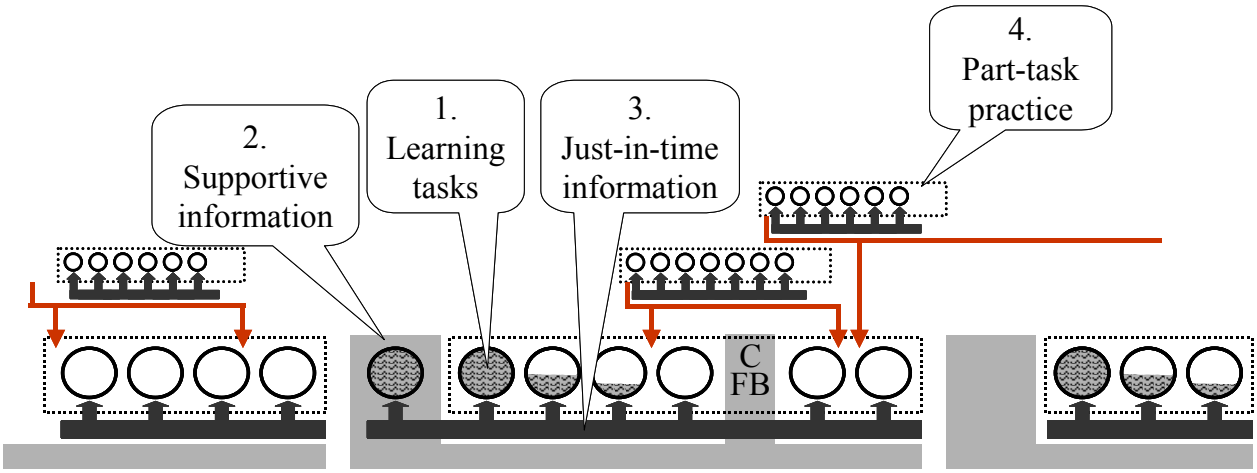


Figure 1.: Four basic components of the 4C/ID model

The moment at which supportive information is presented is fixed and always occurs at the beginning of a task class. In the current TASKi design also the moments at which part-task-practice should be introduced is fixed. In addition a designer has the opportunity to fix the presentation of other blueprint elements. TASKi can arrange the following aspects of the presentation of the learning elements: (a) decision to continue training in the previous, current or next task class; (b) the variability in the presentation of learning tasks within a task class; (c) decision on the amount of learner support for the next to be presented learning task; (d) fading of JIT info, i.e. determining what JIT info needs to be presented with a particular learning task, based on previously presented learning tasks and the current competency on recurrent skills; and (e) the moment at which a PTP session should be stopped.

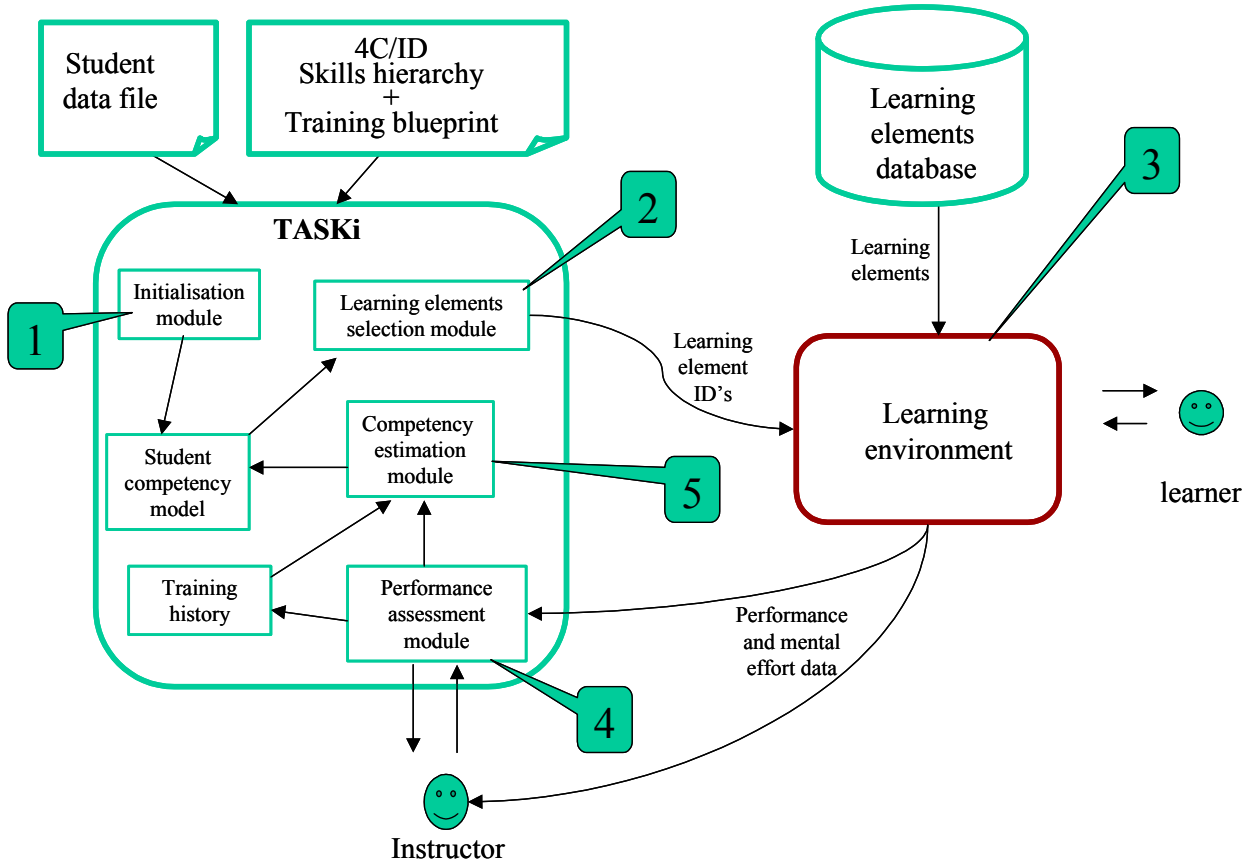


Figure 2.: TASKi architecture

A 3.3.1.3 TASKi process overview

Figure 2 shows an overview of the components of the TASKi system and how it interacts with external components. The constituent skills with their associated performance objectives and standards are stored in the skills hierarchy. The contents (i.e. the learning elements) and the structure of the training program are described in the training blueprint. The concrete, worked out learning elements that can be readily presented to learners are stored in the learning elements database. General information about a student and the learning results achieved at previous training sessions are stored in the student data file.

When the system starts, the initialisation module loads the skills hierarchy, the training blueprint and the student data file and uses this information to initialise the other TASKi components (1). Next, on the basis of the state of the student competency model the learning elements selection module selects the most suitable learning elements to continue training with for this particular learner and sends this information to the learning environment (2). The student competency model indicates how well a learner is able to perform certain learning tasks. The competency level depends on the task class the learner is currently working in and on the amount of learner support the learner currently receives. In other words, the competency model assumes that if a learner is competent to perform a learning task with a certain amount of learner support, this learner will be less competent if the amount of learner support will be lower or if the learning task would be chosen from a task class further on in the sequence. The competency level is modelled for the whole complex skill and for each constituent skill in the hierarchy. From low to high the competency level can contain the values: "incompetent", "novice", "intermediate" and "expert". When selecting learning elements, the learning elements selection module first checks whether training should continue with blueprint elements that are fixed in the blueprint. If this is not the case, then the module selects a next learning task with a sufficient amount of learner support (based on the current competency) and the required additional information elements. After TASKi selected the learning elements, the learning environment is configured with the new elements from the learning elements database (automatically or by an instructor). After configuration is finished the learning process starts and the learner interacts with the learning environment (3). When the learning episode is finished the learner reports the amount of mental effort he or she invested while interacting with the environment. Next the learner's performance for each skill in the hierarchy is assessed—automatically or by an instructor (4). Performances are compared to the performance standards

in the skills hierarchy and then categorised as good, adequate or insufficient. If the performance is judged by an instructor that observed the learner while performing the learning task, then this instructor has to enter the ratings for each constituent skill into TASKi. The competency estimation module estimates on the basis of the latest performance, the mental effort expenditure and the training history the current competency level for the complete complex cognitive skill and each of the constituent skills and stores these values in the student competency model (5). The estimation is based on a) the performance assessment for each skill that was practiced in the last learning task b) the mental effort measure on the last learning task, c) the training history and 4d) the current competency level.

A 3.3.2 ELM-ART: An adaptive versatile System for Web-based Instruction

A 3.3.2.1 Introduction

The WWW-based introductory LISP course ELM-ART (ELM Adaptive Remote Tutor) is based on ELM-PE (Weber and Möllenberg, 1995), an on-site intelligent learning environment that supports example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. The intelligent features of ELM-PE are based on the ELM model (Weber, 1996). For several years, ELM-PE was used in introductory LISP courses at the University of Trier. ELM-PE was limited by the platform dependent implementation of the user interface and the large size of the application. Both limitations hindered a wider distribution and usage of the system. So, the developers decided to build a WWW-based version of ELM-PE that can be used both in intranets and in the Internet. The first step was to translate the texts of the printed materials into WWW-readable form (html files), dividing it into small subsections and text pages that are associated with concepts to be learned. These concepts were related to each other by describing the concepts' prerequisites and outcomes, building up a conceptual network. All interactions of the learner with ELM-ART were recorded in an individual learner model. For each page visited, the corresponding unit of the conceptual network was marked correspondingly. When presenting text pages in the WWW browser, links shown in section and subsection pages and in the overview were annotated corresponding to a simple traffic lights metaphor referring to information from the individual learner model (Schwarz et al., 1996).

The approach of converting printed textbooks to electronic textbooks used in ELM-ART has been developed further in INTERBOOK (Brusilovsky et al., 1996b), an authoring tool for creating electronic textbooks with adaptive annotation of links. However, the first experiences with using ELM-ART showed that printed textbooks are not suitable for being transformed to hypertext pages in electronic textbooks in a one-to-one manner. Textbooks are usually written in sequential order so that single pages cannot be read easily when they are accessed from any page within the course. Additionally, the simple adaptive annotation technique used in ELM-ART had to be improved. Users should get more information on the state of different concepts that they had already visited and learned or had to learn. And, perhaps most importantly, inferring the knowledge state of a particular user from only visiting (and possibly reading) a new page is not appropriate. These objections and shortcomings were the motivation for building ELM-ART II, a new version of ELM-ART we describe in the following sections.

A 3.3.2.2 Functionalities of ELM-ART

ELM-ART II represents knowledge about units to be learned with the electronic textbook in terms of a conceptual network (Brusilovsky et al., 1996a). Units are organized hierarchically into lessons, sections, subsections, and terminal pages. Terminal pages can introduce new concepts or offer problems to be solved. Each unit is an object containing slots for the text to be presented with the corresponding page and for information that can be used to relate units and concepts to each other. Static slots store information on prerequisite concepts, related concepts, and outcomes of the unit (they are the concepts that the system assumes to be known when the user worked on that unit successfully). Units for terminal pages have a tests slot that may contain the description of a group of test items the learner has to perform. When test items have been solved successfully the system can infer that the user possesses the knowledge about the concepts explained in this unit. Problem pages have a slot for storing a description of a programming problem.

Dynamic slots are stored with the individual learner model that is built up for each user. This user model is updated automatically during each interaction with ELM-ART. For each page visited during the course, the corresponding unit is marked as visited in the user model. Moreover, when the test items in a testgroup or a programming problem are solved correctly, the outcome concepts of this unit are marked as known and an inference process is started. In the inference process, all concepts that are prerequisites to this unit (and recursively all prerequisites to these units) are marked as inferred. Information from the dynamic slots in the user model are used to annotate links individually and to guide the learner optimally through the course.

Testgroups are collections of test items that are associated with page units. Single test items may belong to different testgroups. In ELM-ART II, four different types of test items are supported: yes-no test items, forced-choice test items, multiple-choice test items, and free-form test items. In yes-no test items users simply have to answer yes-no questions by clicking the "yes" or the "no" button. In forced-choice test items, users have to answer a question by selecting one of the alternative answers and in multiple-choice test items users have to answer a question by selecting all correct answers provided by the system. In free-form test items, users can type an answer to the question asked freely into a form. Each testgroup has parameters that determine how many single test items are presented to the learner. The *group-length* parameter determines how many test items are presented on a single page. The *min-problems-solved* parameter defines the minimal number of test items that have to be solved correctly within the testgroup. The *max-errors* parameter determines how many errors maximally are allowed in the test items that were presented on a single page. These parameters can be set for each testgroup.

Test items from a testgroup are presented as long as not enough test items have been answered correctly. A fixed number of test items are presented simultaneously on one page. The system gives feedback on the number of errors in the test items presented on the last page and presents all erroneous test items with both the users' answers and the correct answers. Additionally, an explanation is given why the answer provided by the system is the correct one. These explanations are stored separately with each test item. Correctly solved test items from the current testgroup can be accessed via an icon on that page. They are displayed in a new window showing the correct answers as well as the reason why this answer was correct. Users are called on to solve more test items as long as not enough test items have been solved correctly and not too many incorrect answers have been submitted with the last test items. In the individual user model, all test items that are solved correctly for a particular testgroup are stored in a dynamic slot. When enough test items are solved correctly without making too many errors, the outcome concepts of the corresponding unit are marked as solved and the inference process is started. In the current version of ELM-ART II, the values of the *min-problems-solved* parameter vary between 4 and 10 depending on the difficulty of the tests and, in most cases, the *max-errors* parameter is set to 1. That is, after solving at least a number of *min-problems-solved* test items correctly, in the next group of test items shown on a page one error is allowed. In the LISP course, tests play a twofold role. On the one hand, tests are used to check whether the user possesses the correct declarative knowledge. This is especially useful in the beginning of the course when a lot of new concepts (data types and function definitions) are introduced. On the other hand, tests can be used in evaluation tasks to check whether users are able to evaluate LISP expressions correctly. Skills used in evaluation are the inverse skills to generating function calls and function definitions. Evaluation skills are needed to decide whether programs work correctly and to find errors in programming code. Program creation skills are practiced in special tasks.

Visual adaptive annotation of links. ELM-ART II uses an extension of the traffic lights metaphor to annotate links visually (see Figure 3). On the top of each terminal page (below the navigation button line) all links belonging to the same subsection are shown with the links annotated corresponding to their current state. Green, red, yellow, and orange balls are used to annotate the links (additionally, the texts of the links are outlined in different styles to aid color-blind users). A *green* ball means that this page is ready and suggested to be visited and the concepts taught on this page are ready to be learned. That is, all prerequisites to this concept have been learned already or are inferred to be known. A *red* ball means that this page is not ready to be visited. In this case, at least one of the prerequisite concepts is not known to the learner (that is, the system cannot infer from successfully solved tests and programming problems that the user will possess the required knowledge). However, the user is allowed to visit this page and in the case that he or she solves the corresponding test or programming problem correctly, the system infers backwards that all the necessary prerequisites are known. This is a very strong assumption in diagnosing the user's knowledge state and will be changed through the use of fuzzy or probabilistic models in the future. A *yellow* ball has different meanings depending on the type of page this link points to. In the case of a terminal page with a test or a problem page, the yellow ball means that the test or the problem have been solved correctly. In the case of any other terminal page, the yellow ball indicates that this page has been visited already. In the case of a lesson, section, or subsection link the yellow ball means that all subordinated pages have been learned or visited. An *orange* ball has different meanings, too. In the case of a terminal page, an orange ball means the system infers from other successfully learned pages that the content of this page will be known to the learner (as described above). In the case of a lesson, section, or subsection link an orange ball means that this page has been visited already but not all subordinated pages have been visited or worked at successfully.

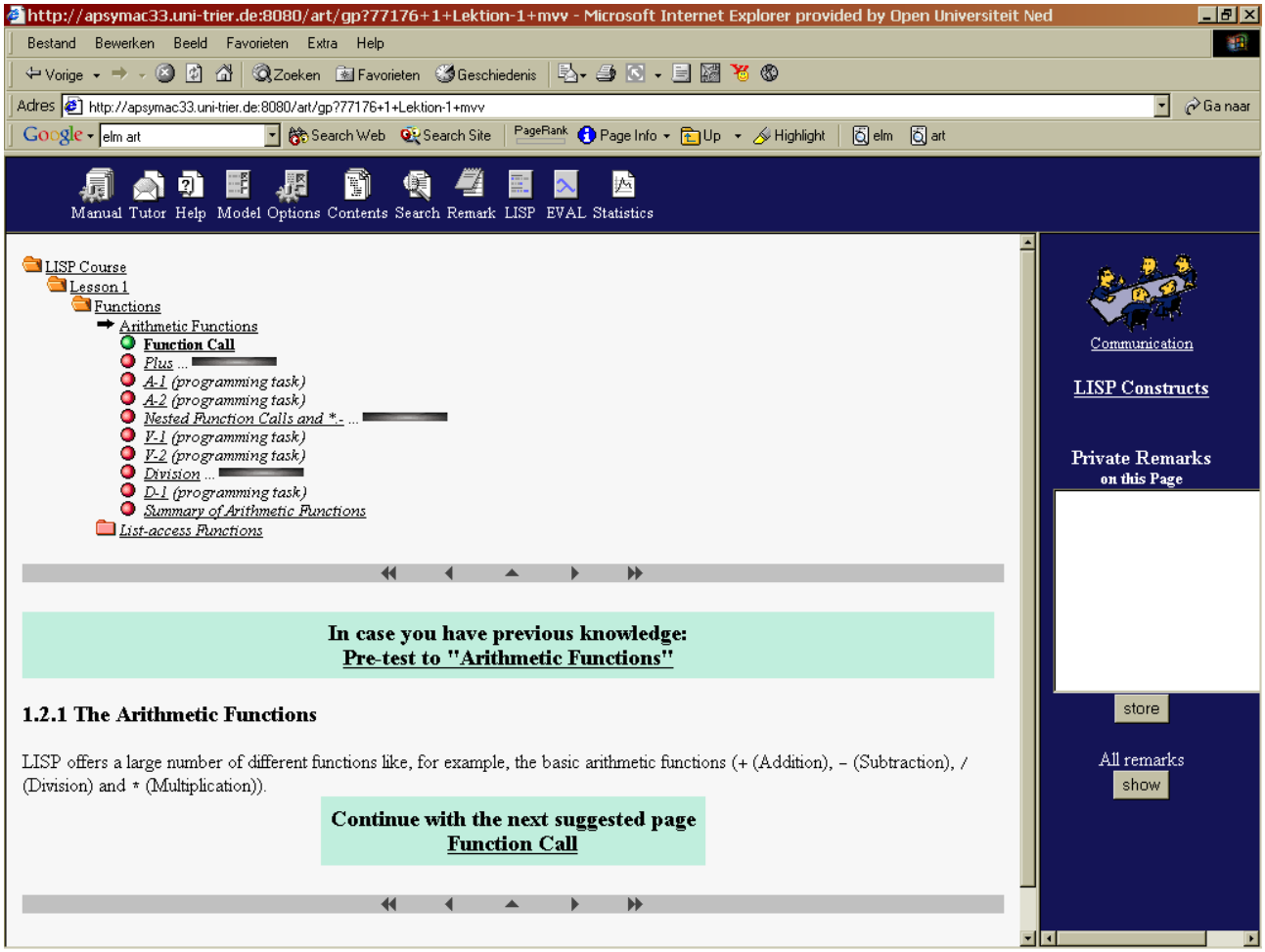


Figure 3. Example of a text page with free-form test items and adaptive annotation of links.

In browsers supporting JavaScript, the different meanings of a state of a link are explained in the status line at the bottom of the window when the cursor is located over the link (Figure 3).

While adaptive annotation of links is a powerful technique to aid users when navigating through the pages of the course, some users may be confused about what the best next step should be to continue with the course. This may happen when the learner moves around in the hyperspace and loses orientation. Or, the learner wants to follow an optimal path through the curriculum in order to learn as fast and as completely as possible. To meet these needs, a NEXT button in the navigation bar of the text pages allows the user to ask the system for the best next step depending on the current knowledge state of the particular user.

The algorithm to select the best next step for a particular user works as follows: Starting from the current page, the next page is searched for a page that is annotated as *suggested to be visited*. This may be the same page as before if it is a terminal page with a testgroup or a problem not solved correctly up to that moment. When no further page can be found with all prerequisites fulfilled, all pages from the beginning of the course are checked to see if they have not yet been visited or solved and the first one found is annotated as *suggested to be visited*. The learner completes the course successfully when no best next page can be found at all.

ELM-ART II supports example-based programming. That is, it encourages students to re-use the code of previously analyzed examples when solving a new problem. The hypermedia form of the course and, especially, similarity links between examples help the learner to find the relevant examples from his or her previous experience. As an important feature, ELM-ART II can predict the student's way of solving a particular problem and find the most relevant example from the individual learning history. This kind of problem solving support is very important for students who have problems with finding relevant examples. Answering the help request, ELM-ART II selects the most helpful examples, sorts them corresponding to their relevance, and presents them to the student as an ordered list of hypertext links. The most relevant example is always presented first, but, if the student is not happy with this example for some reason, he or

she can try the second and the following suggested examples. The implementation of this feature directly was adopted from the recent version of ELM-PE (Burow and Weber, 1996).

If the student failed to complete the solution to the problem, or if the student cannot find an error that was reported when evaluating the code in the evaluator window, he or she can ask the system to diagnose the code of the solution in its current state. The system gives feedback by providing a sequence of help messages with increasingly detailed explanation of an error or suboptimal solution. The sequence starts with a very vague hint on what is wrong and ends with a code-level suggestion of how to correct the error or how to complete the solution. In many cases, the student can understand from the very first messages where the error is or what can be the next step and does not need any more explanations. The solution can be corrected or completed, checked again, and so forth. The student can use this kind of help as many times as required to solve the problem correctly. In this context, the option to provide the code-level suggestion is a very important feature of ELM-ART II as a distance learning system. It ensures that all students will ultimately solve the problem without the assistance of a human teacher.

Both the individual presentation of example programs and the diagnosis of program code are based on the episodic learner model (ELM, Weber, 1996). ELM is a type of user or learner model that stores knowledge about the user (learner) in terms of a collection of episodes. In the sense of case-based learning, such episodes can be viewed as cases (Kolodner, 1993). To construct the learner model, the code produced by a learner is analyzed in terms of the domain knowledge on the one hand and a task description on the other hand. This cognitive diagnosis results in a derivation tree of concepts and rules the learner might have used to solve the problem. These concepts and rules are instantiations of units from the knowledge base. The episodic learner model is made up of these instantiations.

In ELM only examples from the course materials are pre-analyzed and the resulting explanation structures are stored in the individual case-based learner model. Elements from the explanation structures are stored with respect to their corresponding concepts from the domain knowledge base, so cases are distributed in terms of instances of concepts. These individual cases--or parts of them--can be used for two different purposes. On the one hand, episodic instances can be used during further analyses as shortcuts if the actual code and plan match corresponding patterns in episodic instances. On the other hand, cases can be used by the analogical component to show up similar examples and problems for reminding purposes.

The text above is an abstract from: Gerhard Weber and Marcus Specht ' User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems'. For more details see also: 'ELM-ART: An adaptive Versatile System for Web-based instruction'.

A 3.3.3 WebDL

A 3.3.3.1 Introduction

WebDL (Web-based Distance Learning) is an interactive system that is intended to personalize and adapt all the sources of information and communication channels available on the web to meet individual needs and preferences as and when they arise (Boticario, Gaudioso, and Hernandez, 2000; Boticario, Gaudioso and Catalina, 2001). The adaptation is obtained by presenting different structured resources every single moment on the screen in a variety of ways and it also includes, whenever appropriate, advice on using the material available. This approach is therefore wider than the one usually adopted by adaptive educational systems, where adaptation and tutoring tasks focus on access to course contents (Nakabayashi, 1996 and Brusilovsky, 1998).

WebDL is a multiagent architecture designed to adapt to user needs (Boticario and Gaudioso, 1999) and is based on a combination of techniques applied in intelligent tutoring systems (ITS) (Weber and Specht, 1997), adaptive hypermedia programs (AH) (Brusilovsky, 1996) and learning apprentice systems (Dent, Boticario, McDermott, Mitchell and Zabowski, 1992). It falls into the category of so-called Web-based Adaptive Educational Systems (Brusilovsky, 1998). With the learning apprentice approach the initial knowledge base is expanded in order to reduce the effort required for user decision-making (adaptive hypermedia).

A 3.3.3.2 WebDL architecture

The WebDL core (Boticario and Gaudioso, 2000) is designed in terms of a multiagent decision approach and is organized as follows (see Figure).

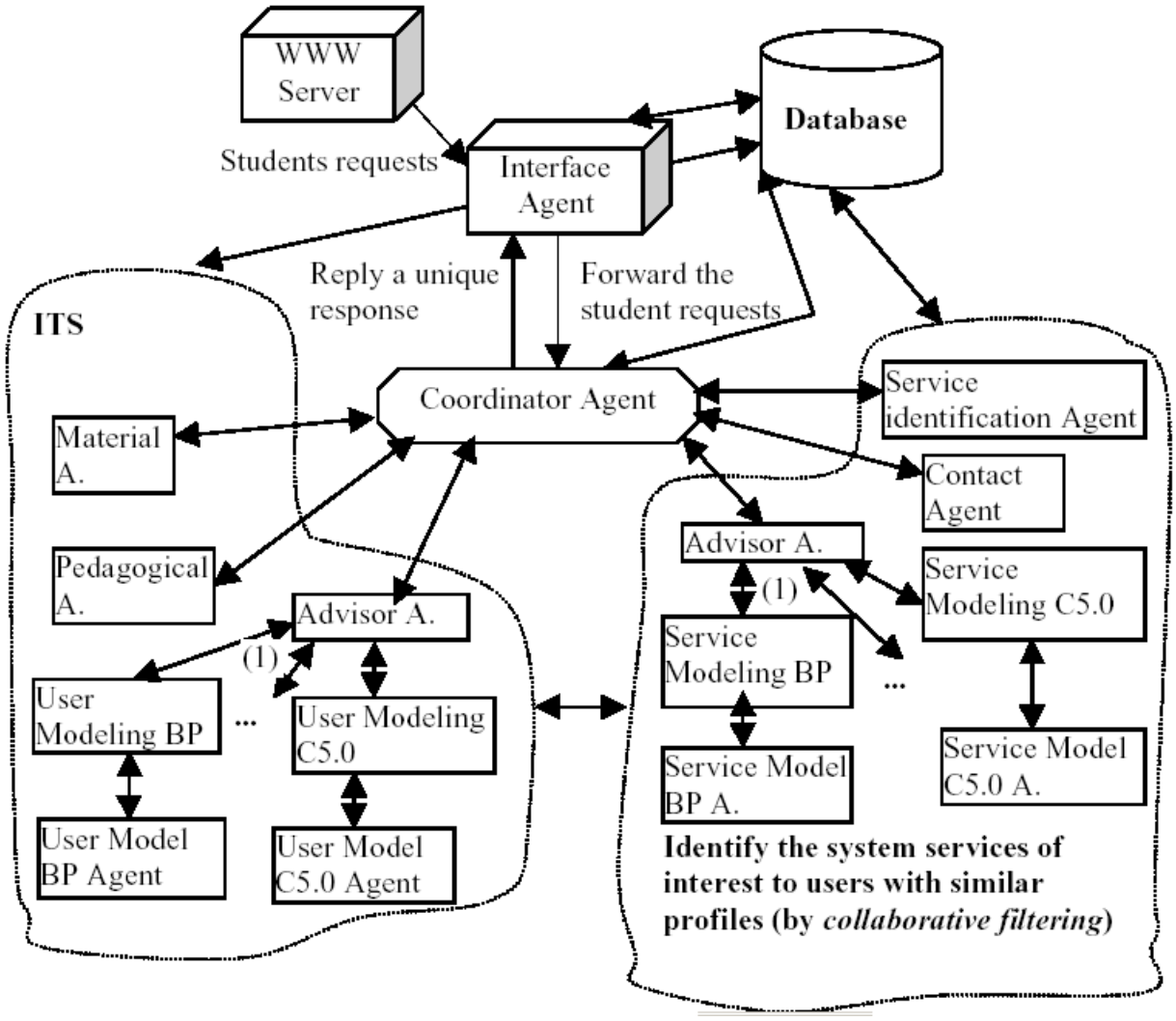


Figure. WEB-DL general Architecture

Two main components are involved: user interaction and the adaptive module. The former is implemented by the interface agent and is in charge of the organized presentation of the different types of material. This module provides a single, integrated response to the user. The adaptive module consists of the following agents: user model agent, user modelling agent, material agent, pedagogical agent, contact agent, service identification agent, service model agent, service modelling agent and coordinator agent. The first four provide basic ITS functionality. The next four are useful for identifying, by collaborative filtering, the system services of interest to users with similar profiles. Each one of these agents has their own knowledge base with the entities that they control together with their respective relations. The agents that perform learning tasks are able to extend their knowledge base dynamically adding new entities and changing the existing ones as the user interacts with the system (they adjust to what have been called learning apprentice systems). All the entities contained in the aforementioned knowledge bases are represented via THEO (Mitchell, Allen, Chalasani, Cheng, Etzioni, Ringuette and Schlimmer, 1990), a framework for learning and problem solving. A brief description follows of each of the agents highlighted above.

The Interface Agent has two main functions: to gather student requests and forward them to the coordinator agent and to construct pages finally presented to a student. These pages are constructed according to the coordinator agent's response and design rules defined in its knowledge base.

The Coordinator Agent is responsible for gathering the interface agent request and forwarding the student request to the agents involved in the request. When the coordinator agent finally receives the response from the rest of the agents it constructs one response to send to the interface agent. In order to construct this response, this agent creates a special entity, *advice*, for each request in its knowledge base, where other agent responses to the request are stored.

The User Model Agent and the Service Model Agent are responsible for acquiring, saving and managing all the WebDL user or service information. They collaborate with the other agents to provide them with data that might be necessary for their tasks.

The Material Agent is responsible for gathering and managing all the Website material.

The Pedagogical Agent is responsible for personalizing teaching to connecting students. It is the equivalent of the pedagogical module of an Intelligent Tutoring System. So far it provides adaptive guidance to the student via adaptive navigation support through the course material.

The Contact Agent is responsible for cooperation and communication among students and with the professors. It thus uses the user model, student and professor contributions, data about the users that are connected at every single moment, etc.

The Service Identification Agent selects the services of interest to a significant number of users through clustering techniques; the intention here is for the system to learn the characteristics determining which services may be of interest to a given user.

The User Modeling Agent and the Service Modeling Agent are responsible for learning each system user model and service model (see Advisor Agent description).

In order to achieve good user personalization, heterogeneous agents that combine the solutions learned with different biases corresponding to different machine learning methods (*C5.0 (*Quinlan, 1997), *Naive Bayes* (Smith, 1988), *Progol* (Muggleton, 1995), *Backpropagation* (Rumelhart, Hinton and Williams, 1986), *Tilde* (Blockeel and Raedt, 1997) and *Autoclass* (Cheesman, Stutz, Hanson and Taylor, 1990)) have been chosen. It is thus possible to apply different approaches to different tasks and clarify whiych option is the most appropriate for each instance. To implement this combination, a special type of agent has been introduced: the Advisor Agent. Several advisor agents learn the competence range of each of the other agents. A task is only distributed to those agents that have been proved to be competent in it. Competence region learning is done by reinforcement, accounting each modeling option agreement or disagreement with the right or wrong solution.

Concerning system functionality, two fundamental learning tasks are carried out: one when the different modeling agents infer a certain value of their corresponding models; another when the advisor agents learn the degree of competence of the corresponding modeling agents, once system interaction with a user has finished.

A 3.3.3.3 Adaptive Navigation Support and Adaptive Collaboration Support in WebDL

With respect to the adaptive navigation support task, the system guides students through the course contents via adaptive link annotation. In this instance, the pedagogical model monitors the adaptive navigation support task, which consists of guiding students through Web contents via link annotations.

With respect to the adaptive collaboration support task, the system is responsible for guiding cooperation and communication among students and with lecturers. From the data stored on the user, the system can determine all the information regarding his/her collaboration in the workgroups in which he/she participates. This is because aLF stores in the database all the annotations made by the user, every message sent to the forums and to the group components, and all the group interactions. Therefore, the system can include in the user model certain attributes that reflect his/her interaction and collaboration profile.

A 3.4 Assessment

A 3.4.1 Situation and prognosis

Adaptive educational systems can be viewed upon from two perspectives: 1) from an educational point of view, what is the purpose of interacting with the educational system, what are the learner's needs, wishes, traits or particular circumstances that the educational system should adapt to, etc., and 2) from a technical point of view, what are the methods and techniques available to adapt a system to a user. In most studies the focus is primarily on the second viewpoint (e.g. Brusilowski, 1996,1998, 2001; Ostyn, 2001). In these studies the technical capabilities of adaptive educational systems are described without too many considerations about what the educational requirements for adaptive systems are. In this section we will first attempt to provide an overview of educational requirements for adaptive instruction and then analyse how well the adaptive capabilities of the systems described above map on these requirements.

The purpose of educational systems is to provide information to learners to help them to reach certain learning goals, that is, to help learners to learn certain skills and/or to increase their knowledge about a certain topic. Different learners might have different needs, characteristics, prior knowledge, etc., therefore the purpose of making educational systems adaptive to individual learners is to take these aspects into account when presenting information in order to make the learning process as effective, efficient, and motivating as possible. For the purpose of this review we identified the following educational requirements for adaptive systems:

- Information should adapt to what a learner already knows (prior knowledge) or can do (prior skill).
- Information should adapt to a learners' learning capabilities.
- Information should adapt to a learners' learning preferences or style.
- Information should adapt to a learners' performance level and knowledge state (i.e. system should provide feedback)
- Information should adapt to a learners' interests.
- Information should adapt to a learners' personal circumstances (location, tempo, etc.).
- Information should adapt to a learners' motivation.

The following table provides an overview of the three discussed adaptive systems and if and how they fulfil each requirement.

System adapts to a learners':	TASKi	ELM-ART II	WEB-DL
prior knowledge and skill	indirectly If a leaner has prior knowledge and skill it is assumed that this will lead to an initially high performance and the learner will advance more quickly through the course.	yes Test are used to assess a learners' initial; knowledge state.	yes Tests are used to consider learners' preferences and background
learning capabilities	yes The systems explicitly attempts to model the rate of skill development and adapt the training by adjusting task difficulty and amount of learner support.	partially Systems models knowledge state and adjusts the presentation of links to new knowledge accordingly.	yes To date focus on collaboration capabilities
learning preferences or style	no	no	no
performance level/ knowledge state	yes/no System adapts to performance level by adjusting difficulty of learning tasks and amount of learner support.	no/yes System attempts to model the knowledge state and to adjust the presentation of new knowledge accordingly. System provides increasingly more feedback on incorrect task performance.	no/yes System attempts to model the knowledge state and to adjust the presentation of new knowledge by means of adaptive navigation support.
interests	no	yes System allows learner to choose his/her own learning path. Systems searches for relevant examples to solve problems.	yes system attempts to identify web-resources and services that might be of interest to the learner. System allows learner to choose his/her own learning path.
personal circumstances	depends on the implementation of the learning environment	no direct support, but the system is fully web-based and can therefore be individually paced and used at different locations.	no direct support, but the system is fully web-based and can therefore be individually paced and used at different locations.

System adapts to a learners':	TASKi	ELM-ART II	WEB-DL
motivation	indirectly	indirectly	indirectly

Each of the three systems have their own particular advantages and disadvantages. The TASKi system is the only system that explicitly attempts to adapt to differences in learners' capabilities by modelling skill development and adjusting task difficulty and learner support accordingly. ELM-ART II uses elaborated episodic learner models to allow for adaptive guidance and to individualized help and problem solving support. WEB-DL uses more traditionally knowledge overlay models that model what a learner knows and not knows, but not how well a learner can perform a certain skill. Both ELM ART II and Web-DL adapt to a learners interests by allowing learners to choose their own learning path and by giving advice on best choices by means of ordering or highlighting of links. TASKi does not allow a user to choose his or her own learning path. A further advantage of WEB-DL is that it explicitly addresses collaborative learning. On the basis of a users' interaction with the system and/or other users WEB-DL generates advise for joining collaborative workgroups or using collaborative services. In addition, WEB-DL is the only system that can identify and present information that is not part of the pre-defined adaptive course. None of the systems adapts to learning preferences or learning style. According to Brusilowski (2001) this is not surprising because at present it is unclear what exactly learning styles are and how instruction should adapt to it.

A 3.4.2 Conclusions for Alfanet

From the viewpoint of the developing Alfanet the most important requirements for an adaptive educational approach are that it will allow for an adaptive presentation of contents 1) that is effective, 2) that is domain independent, 3) that allows for collaborative learning, and 4 that is web-based. Regarding the first point no information is available. No information was available about the effectiveness of adaptive educational systems compared to traditional non-adaptive systems. Regarding point two it can be concluded that the approaches of all systems are generally applicable in different domains. However, for TASKi and ELM ART II courses will have to be specifically developed in great detail. Only WEB-DL has functionality that allows for the adaptation of contents that are not part of a purposely, pre-specified instructional design. With regard to point three, none of the systems disallows collaborative learning. WEB-DL is the only system that specifically addresses collaborative learning by providing advice on the use of collaborative services. With respect to point four, ELM ART II and WEB-DL are purposely developed for usage on the internet. Whether the TASKi module can be used on the internet, depends on the implementation of the learning environment that TASKi controls. No clear conclusions can be made for the use of the described approaches in Alfanet. None of the approaches appears to have characteristics that would make it problematic to include it in Alfanet. Each of the approaches have their specific advantages and disadvantages.

A 3.5 References

Boticario, J.G., Gaudioso E., Catalina, C. *"Towards personalised learning communities on the Web"*. First European Conference on Computer-Supported Collaborative Learning, pp. 115-122, 2001.

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

Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), Spec. Iss. on Adaptive Hypertext and 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" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, August 16-19, 1998.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996a). ELM-ART: An intelligent tutoring system on World Wide Web. In Frasson, C., Gauthier, G., and Lesgold, A., eds., Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS-96. Berlin: Springer. 261-269.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996b). A tool for developing adaptive electronic textbooks on WWW. Proceedings of WebNet'96, World Conference on the Web Society. Charlottesville: AACE. 64-69.

Burns, H. L. and Capps, C. G. (1988) Foundations of intelligent tutoring systems: An introduction. In: M. C. Polson and J. J. Richardson (eds.): *Foundations of intelligent tutoring systems*. Hillsdale: Lawrence Erlbaum Associates, pp. 1-19.

Burow, R., and Weber, G. (1996). Example explanation in learning environments. In Frasson, C.,

De Croock, M. B. M., Paas, F. G. W. C., Schlanbusch, H., & van Merriënboer, J. J. G. (2002). ADAPTIT: ID Tools for Training Design and Evaluation. Educational Technology, Research and Development.

D.E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing: Explorations in the Microstucture of Cognition. MIT Press, Cambridge, MA, 1986.

Dent, C.L., Boticario, J.G., McDermott, J.P. Mitchell, T.M. and Zabowski, D. *A Personal Learning Apprentice*. AAAI 1992: 96-103.

Gerhard Weber and Marcus Specht. User modeling and adaptive navigation support in www-based tutoring systems. In Proceedings of the Sixth International Conference on User Modeling, pages 289_300, Chia Laguna, Sardinia, Italy, June 1997.

H. Blockeel and L.D. Raedt. Top-down induction of logical decision trees. In Proceedings of the 9th Dutch Conference on Arti_cial Intelligence NAIC97, 1997.

J. Smith. Decision Analysis, A Bayesian Approach. Chapman and Hall, 1988.

Jesus G. Boticario and Elena Gaudioso. A multiagent architecture for a web-based adaptive educaticon system. In Seth Rogers and Wayne Iba, editors, Adaptive User Interfaces, Papers from the 2000 AAAI Spring Symposium, TR SS-00-01, pages 24_27. AAAI Press, March 2000.

Jesus G. Boticario and Elena Gaudioso. Towards personalized distance learning on the web. In J. Mira and J.V. Sanchez-Andres, editors, Foundations and Tools for Neural Modeling, number 1607 in Lecture Notes in Computer Science, pages 740_749. Springer Verlag, 1999.

John C. Mallery. A common lisp hypermedia server. In Proceedings of The First International Conference on The World-Wide Web, Geneva: CERN, 1994.

K. Nakabayashi. An intelligent tutoring system on the www supporting ubteractive simulation environments with a multimedia viewer control mechanism. In Proceedings of WebNet96, page 366, San Francisco, CA, October 1996. World Conference of the Web Society.

Kolodner, J. L. (1993). Case-Based Reasoning. San Mateo, CA: Morgan Kaufmann.

L. Dent, J. G. Boticario, J. McDermott, T. M. Mitchell, and D. T. Zabowski. A personal learning apprentice. In Proceedings of the Tenth National Conference on Artifcial Intelligence, pages 96_103, San Jose, CA, 1992. MIT Press.

Makedon, F., eds., Proceedings of ED-TELEKOM 96 - World Conference on Educational Telecommunications. Charlottesville, VA: AACE. 302-307.

Mitchell, T. (1991) Personal Learning Apprentices. ISMIS 1991: 35-37.

Mitchell, T., Mahadevan, S. and Steinberg, L. Leap: A learning apprentice for vlsi design. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence. Morgan Kaufmann, August 1985.

P. Cheesman, J. Stutz, R. Hanson, and W. Taylor. Autoclass III. Rsearch Institute for Advanced Computer Science, NASA Ames Research Center, 1990.

Ross Quinlan. C5.0 Data Mining Tool. www.rulequest.com, 1997.

S. Muggleton. Inverse entailment and progol. New Gen. Comput., 13:245_286, 1995.

Schwarz, E., Brusilovsky, P., and Weber, G. (1996). World-wide intelligent textbooks. In Carlson, P., and

Tom M. Mitchell, John Allen, Prasad Chalasani, John Cheng, Oren Etzioni, Marc Ringuette, and Jeffrey C. Schlimmer. Theo: A framework for self-improving systems. In K. VanLehn, editor, Architectures for Intelligence. Erlbaum, Hillsdale, NJ, 1990.

Weber, G. (1996). Episodic learner modeling. Cognitive Science 20:195-236.

Weber, G., and Möllenberg, A. (1995). ELM programming environment: A tutoring system for LISP beginners. In Wender, K. F., Schmalhofer, F., and Böcker, H.-D., eds., Cognition and Computer Programming. Norwood, NJ: Ablex Publishing Corporation, 373-408

Appendix 4 Advanced instructional design models

A 4.1 Overview

The main goal of any instructional design process is to construct a learning environment in order to provide learners with the conditions that support desired learning processes. With regard to models that may sustain this process, Van Merriënboer (1997) makes a distinction between Instructional Systems Development (ISD) models and Instructional Design (ID) models. ISD-models have a broad scope and typically divide the instructional design process into five phases: (1) analysis, (2) design, (3) production, (4) implementation and/or delivery, and (5) summative evaluation. In such stage-models, formative evaluation is typically conducted during all phases. ISD-models provide guidelines and directions for performing the activities that form part of each of the phases. ID-models are less broad in scope and focus on the first two phases of ISD-models (i.e., analysis and design). They concentrate on the analysis of a to-be-trained skill in a process of job and task analysis and the conversion into a training strategy, or the design of a learning environment (often taking the form of some kind of blueprint) that is ready for production. If it comes to the analysis of to-be-trained skills and the design of learning environments, ID-models typically provide more specific guidelines and directions than ISD-models. In this section the focus lies on describing the strong and weak points of state-of-the-art ID-models.

Recently Merrill (2002) identified the *First Principles of Instruction* to predict the quality and effectiveness of ID models. First principles are design oriented, fundamental instructional principles that apply under all conditions of learning but can be implemented by a wide variety of programs and practices. Learning from a given instructional program will be facilitated in direct proportion to the explicit implementation of first principles of instruction. Based on studies into the effectiveness of many instructional programs, Merrill suggests that the most effective learning environments are problem based and involve the students in four distinct phases of learning: (1) activation of prior experiences, (2) demonstration of skills, (3) application of skills, and (4) integration of these skill into real-world activities. Based on this model he then lists the following first principles of instruction:

Learning is facilitated when learners are engaged in solving real-world problems.

- Learning is facilitated when learners are introduced to the task that they will be able to do or the problem they will be able to solve as a result of completing a module or a course.
- When solving problems, learning is facilitated when learners are engaged in all levels of instruction for solving problems, that is, the problem, the tasks required to solve the problem, the operations that comprise the tasks and the actions that comprise the operations.
- Learning is facilitated when learners solve a progression of problems that are explicitly compared to one another.

Learning is facilitated when existing knowledge is activated as a foundation for new knowledge.

- Learners are directed to recall, relate, describe, or apply knowledge from relevant past experience that can be used as a foundation for the new knowledge.
- Learners are provided relevant experience that can be used as a foundation for new knowledge.
- Learners are given the opportunity to demonstrate their previously acquired knowledge or skill.

Learning is facilitated when the instruction demonstrates what is to be learned rather then merely telling information about what is to be learned.

- Learning is facilitated when the demonstration is consistent with the learning goal: (a) examples and non-examples for concepts, (b) demonstrations for procedures, (c) visualisations for processes, (d) modelling for behaviour.
- Learning is facilitated when learners are provided appropriate learner guidance including some of the following: (a) learners are directed to relevant information, (b) multiple representations are used for the demonstrations, or (c) multiple demonstrations are explicitly compared.
- Learning is facilitated when media plays a relevant instructional role.

Learning is facilitated when learners are required to use their new knowledge or skill to solve problems.

- Learning is facilitated when the application (practice) and the posttest are consistent with the stated or implied objectives: (a) information-about practice—recall or recognize information, (b) parts-of practice—locate, name and/or describe each part, (c) kinds-of practice—identify new examples of each kind, (d) how-to practice—do the procedure ,and (e) what-happens practice—predict a consequence of a process given conditions, or find faulted conditions given an unexpected consequence.
- Learning is facilitated when learners are guided in their problem solving by appropriate feedback and coaching, including error detection and correction, and when this coaching is gradually withdrawn.
- Learning is facilitated when learners are required to solve a varied sequence of problems.

Learning is facilitated when learners are encouraged to integrate (transfer) the new knowledge or skill into their everyday life.

- Learning is facilitated when learners are given an opportunity to publicly demonstrate their new knowledge or skill.
- Learning is facilitated when learners can reflect-on, discuss, and defend their new knowledge or skill.
- Learning is facilitated when learners can create, invent, and explore new and personal ways to use their new knowledge or skills.

It is important to note theta Merrill does not see collaboration as first principle of instruction. In Merrill's view collaboration is only one of many possible ways for implementing first principles.

In the next section the most relevant ID models are briefly described. In the third section (assessment) they will be compared to the first principles as identified by Merrill and conclusions will be drawn as to which (elements of) the described ID models are most relevant for the further development of the aLFanet system.

A 4.2 ID Models and tools

A 4.2.1 Constructivists Learning Environments (CLE)

According to Jonassen (1998), Objectivists conceptions of learning assume that knowledge can be transmitted by technologies and acquired by learners. Against this conception Jonassen states the constructivists assumption that knowledge is individually constructed and socially co-constructed by learners based on their interpretations of experiences in the world. So, because knowledge cannot be transmitted Jonassen states that Constructivists learning environments (CLE's) should facilitate knowledge construction by providing experiences that engage learners in meaning making.

The central component of a CLE is a problem, question, or project that learners should attempt to solve or resolve. This problem drives the learning and students learn domain content in order to solve the problem. Depending on the type of problem CLE's can be question-based, issue-based, case-based, project-based, or problem based. These CLE types represent an approximate increase in complexity but they all share the same assumptions of active, constructive and authentic learning. The main constructivists principle is that meaningful learning can only be achieved if the learners take ownership of the problem. Therefore the problems should be interesting, relevant, and engaging to solve. This can be achieved by presenting learners with ill-structured problems, that is, they are not overly circumscribed and some aspects of the problems are emergent and definable by the learners. Jonassen states that ill-structured problems should:

- have un-stated goals and constraints
- possess multiple solutions, solution paths, or no solutions at all
- possess multiple criteria for evaluating solutions
- present uncertainty about which concepts, rules, and principles are necessary for the solution or how they are organized
- offer no general rules or principles for describing or predicting the outcome of most cases
- require learners to make judgements about the problem and to defend their judgement by expressing personal opinions or beliefs

Furthermore, problems need to include three integrated components: (1) the problem context, describing the physical, socio-cultural and organizational climate surrounding the problem; (2) the problem representation or simulation, a detailed description of the authentic problem; and (3) the problem manipulation space, describing the objects, signs and tools required for the learner to manipulate the environment.

Surrounding the problem are various interpretative and intellectual support systems. The first two system, Related cases and Information resource, support understanding of the problem and suggest possible solutions. Related cases assist students in understanding the issues implicit in the problem representations and provide solutions that can be used as analogies for solving the central problem of the CLE. Information resources provide learners with the information that learners need to construct their mental models and formulate hypotheses that drive the manipulation of the problem space. The third system, cognitive tools, help learners to interpret and manipulate aspects of the problem. Cognitive tools are intellectual devices that that are used to visualize (represent), organize, automate, or supplant thinking skills. They scaffold a learners performance on novel tasks by engaging and facilitating specific kinds of cognitive processing that are required to perform the task (Kommers, Jonassen & Mayes, 1992). Fourth, conversation/collaboration tools enable communities of learners to negotiate and construct meaning for the problem. CLEs should provide access to shared information and shared knowledge-building tools to help learners to collaboratively construct socially shared knowledge. The final type of systems, social/contextual support systems, are not intended to be used by learners but by CLE designers. These systems help designers to implement the CLE by addressing important physical, organizational and cultural aspects of the environment in which the CLE has to be implemented.

According to Jonassen, Learning in a CLE is supported by three types of instructional interventions that each support a specific type of learning activity. The first instructional intervention is modelling. It supports a learner to explore the aspects of a problem, that is, to investigate relates cases for similarities, and to peruse information resources to find evidence to support a solution to the problem or completion of the project that the CLE focuses on. The CLE can provide modelling support by behaviour modelling of the overt performance and by means of cognitive modelling of the covert cognitive processes. Behavioural modelling demonstrates how to perform the activities that are required to solve the problem. Cognitive modelling articulates the reasoning (reflection-in-action) that learners should use while engaged in the activities. The second instructional intervention, coaching, supports a learner to articulate what they know and have learned. Requiring learners to articulate what they are doing in the environment and the reasons for their actions and to explain the strategies they use supports knowledge construction and metacognition. Coaching is provided by motivating learners, analysing their performances, providing feedback and advice on the performances and how to learn and perform, and by provoking reflection on and articulation of what was learned. The third instructional intervention, Scaffolding, should support a learner to reflect on what they did, why it did or didn't work, and what they have learned from the activities. To scaffold a learners performance the CLE can (1)perform part of the task for the student, (2) supplant the students ability to perform some part of the task by changing the nature of the task or imposing the use of cognitive tools that help the learner perform, or (3) adjust the nature or difficulty of the task.

A 4.2.2 Learning by Doing

Schank, Berman and Macpherson (1999) identified several problems with traditional methods of instruction that impair learning. In their view teaching is concentrated too much on impacting factual knowledge on students whereas students should be learning skills. They also state that student activities are focussed on finishing homework problems or passing tests and not on achieving relevant and meaningful goals that relate to the subject matter. Finally, in traditional teaching students are learning in a decontextualised fashion, in which the use of knowledge or skills has no resemblance with they way they are used in real life. A result of all these less optimal teaching conditions is that learners acquire and store knowledge in a way which makes it difficult to remember and apply in real life. To overcome these problems they developed a structure for teaching and learning called goal-based scenarios (GBS), that fosters skill development and the learning of factual information in the context of how it will be used. Basically a GBS is a learning-by-doing simulation in which students pursue an authentic and motivating goal by practicing target skills and using relevant content knowledge to help them achieve their goal.

The GBS is based on the theory of case based reasoning (CBR) that describes how people remember and use memories in order to solve new problems. The prime principle of CBR is that people learn from experiences. Experts have had many experience in their area of expertise. They have these experiences stored in memory as 'mental libraries' which they use to retrieve examples that help them to solve new problems. Experts can retrieve relevant examples for given problems, because their experiences are indexed in memory by the context of the experience, the goal of the experience, or the lesson learned from the experience. It allows for far transfer of knowledge to new situations by means of analogical reasoning.

An important part of the learning process according to CBR are the occurrence of expectation failures. Expectation failures occur when certain actions do not lead to an anticipated goal. They create prime conditions for learning, because they stimulate a learner to look for explanations for the failure and to abstract lessons from the experience that can be applied to expectations in the future.

When designing a GBS, the main goal is to create a motivating environment in which students are situated such that they become ready to learn. Students become ready to learn when they are driven by their own internal motivation to learn new knowledge or skills. In GBS this condition is created by designing a task for the students that is motivating, difficult to achieve and that requires acquisition of the knowledge and skills that one wishes to impart. There are seven essential components that comprise a GBS. The first component are the learning goals. The *learning goals* clearly state what a designer wants the students to learn. Learning goals fall under two categories: (1) process knowledge, the knowledge of how to practice skills that contribute to goal achievement, and (2) content knowledge, the information that achievement of a goal acquires. The second component, *the mission*, is a goal that the students will pursue while they work with the GBS. It is important that the mission is authentic, realistic, motivating and suitable for reaching the learning goals that were identified in the first step. The third component, the *cover story*, is a background story line that creates the need for the mission to be accomplished. For example, Schank, Berman and Macpherson (1999) used a story about two fictitious countries that had gone to war and in which it was the mission of the students to advice the president whether or not to intervene. The cover story serves as a channel through which the student is required to practice the skills and learn the content that comprise the learning goals. The forth component, *the role*, defines who the student will play in the cover story. A role should be chosen that is best in the scenario to practice the necessary skills and that is truly motivating to a student. The fifth component, *the scenario operations*, comprises all of the activities the student has to perform in order to work towards the mission goal. They should be closely related to both the mission and the learning goals. The scenario operations also should be constructed such that periodically during role-play decision points arise. The effects of the decisions the student makes signify success or failure of progress toward successful completion of the mission. The sixth element, *resources*, provide the information the students need to achieve the goal of the mission. It is readily accessible and well-organised information to help the student to complete the mission successfully. It is best not presented as decontextualised facts, but rather in the form of stories that the students can understand as an extension of the stories he or she already knows. Finally the seventh component, *feedback*, allows students to properly index information about expectation failure as it is given. It is situated in an appropriate context and provided just in time for the student to use. It can be given in three ways: (1) as a consequence of an action, (2) by means of a coach, providing advice when needed, and (3) through domain experts who tell stories that pertain to similar experiences as the one the student is encountered with at that particular moment.

A 4.2.3 Collaborative Problem Solving

Collaborative problem solving (CPS) is a theory that provides guidelines for the development of collaborative learning environments for problem based learning (Nelson, 1999). It address the whole of the collaborative learning processes, including building a readiness in students to learn collaboratively, developing group skills, forming groups, engaging in collaborative problem solving, and finalizing the process through appropriate synthesis, assessment, and closure activities. CPS emphasizes the following pedagogical values:

- maximizing the natural collaborative processes of learners in stead of imposing artificial collaborative structures;
- creating learning environments which are situated, learner-centred, integrated and collaborative, versus ones which are decontextualised, isolated and competitive;
- honouring the importance of authenticity, ownership, and relevance of the learning experiences for students in relation to the content to be learned and the process by which it is learned;
- allow students to learn by doing as active participants in their own learning processes;
- fostering the development of critical thinking and problem solving skills;
- encouraging the exploration and analysis of content from multiple perspectives;
- acknowledging the importance of rich social contexts for learning;
- cultivating supportive, respectful relationships among learners, as well as between learners and the instructor;
- develop a desire for life-long learning and the skill to sustain it.

The primary goals of the CPS theory are to develop contents knowledge in complex domains, problem solving and critical thinking skills, and collaboration skills. Therefore the theory should only be used when those types of learning are paramount and when the student and the instructor are receptive to this approach to learning, with its shift in roles and power relationships.

Nelson grouped her guidelines into two general categories: (a) *comprehensive guidelines*, which are applied to and support the entire process of developing a CPS environment, and (b) *process activities*, which are used during specific phases of the process.

Comprehensive guidelines

The comprehensive guidelines are categorized in four groups that reflect the commonalities between the guidelines and their instructional intent. The first group are Instructor-Implemented Methods, that a teacher can use to reconceptualise his or her role. The guidelines are:

Act a resource and tutor/facilitator rather then as a dispenser of knowledge

Create learning environments that allow learners to work in a variety of small groups, each for an extended period of time

Formulate questions to focus the learners on the important aspects of the content and their own learning processes

Provide just-in-time instruction when requested by learners

The second group, learner implemented methods, are guidelines for learners to form groups and manage group activities. The guidelines are:

Determine how the acquired knowledge and resources will be used to resolve the problem

Determine and account for individual and group time spent on project activities

The third group, Instructor- and Learner-Implemented Methods, are guidelines that are intended to be implemented collaboratively as the instructor and learners work in concert. The guidelines are:

Collaborate to determine learning issues and objectives

Conduct group progress meetings with the instructor

Collect needed resources

Evaluate learners in multiple ways

Provide group and individual evaluations and grades

The fourth group, Interactive Methods, provide guidance for the interactions which take place during CPS. They assist the instructor in supporting the work of the learners and in dealing with concerns and problems as they arise, and they help the learners engage in powerful, meaningful, and effective collaborative processes that allow them to learn the content and develop better group problem-solving skills. The guidelines are:

Learn and purposefully use appropriate social skills, such a s leadership, decision making, trust building, communication, and conflict management.

Engage in team building activities

Promote notions of investigation, interaction, interpretation, and intrinsic motivation

Encourage simultaneous interaction where there are multiple active participants engaged in problem-solving tasks

Ensure equal participation so that all learners have an opportunity to contribute

Promote positive interdependence, in that each group member is positively linked with others in such a way that the individual cannot succeed unless the group does

Advocate face-to-face promotive interaction in which the student promote each other's success by praising, encouraging, helping, and supporting each other's efforts to learn

Require individual accountability where students are held responsible for doing their share of the work

Process activities

The process activities provide a general blueprint to learning groups about the type of activities in which they should engage as they collaborate on their solution or project. These learning activities will likely take place over a number of weeks and are often iterative and/or concurrent. Below follows a list of the process activities and their associated guidelines.

Instructor and learners establish and build their readiness to engage in collaborative group work

Overview the collaborative problem-solving process

Develop an authentic problem or project scenario to anchor instruction and learning activities

Provide instruction and practice in group process skills

Either the instructor or the learners form small, heterogeneous work groups, and then the groups engage in norming processes

Form small, heterogeneous workgroups

Encourage groups to establish operational guidelines

Groups engage in a preliminary process to define the problem they will work on

Negotiate a common understanding of the problem

Identify learning issues and goals

Brainstorm preliminary solutions or project plans

Select and develop an initial design plan

Identify sources of needed resources

Gather preliminary information to validate the design plan

Each group defines what roles are necessary to accomplish the design plan and then assigns them

Identify the principles and roles needed to complete the design plan

Negotiate the assignment of roles

The group engages in the primary, iterative CPS process

Refine and evolve the design plan

Identify and assign tasks

Acquire needed information, resources, and expertise

Collaborate with instructor to acquire additional resources and skills needed

Disseminate acquired information, resources, and expertise to the other group memebers

Engage in solution- or project-development work

Report regularly on individual contributions and group activities

Participate in intergroup collaborations and evaluations

Conduct formative evaluations of the solutions or project

Groups begin to finalize their solutions or projects

Draft the preliminary final version of the solution or project

Conduct the final evaluation or usability test of the solution or project

Revise and complete the final version of the solution or project

The instructor and learners engage in activities to help them reflect and synthesize their experiences

Identify learning gains

Debrief experiences and feelings about the processes

Reflect on group and individual learning processes

The instructor, and, when appropriate, the learners assess their products and processes

Evaluate the products and artefacts created

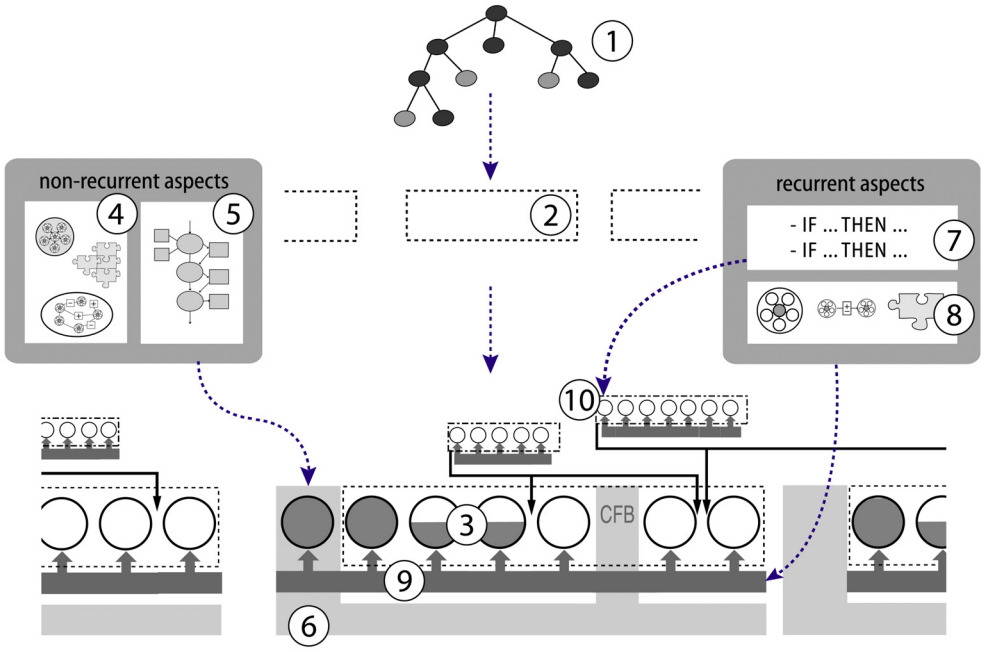
Evaluate the processes used

The instructor and learners develop an activity to bring closure to the learning event

Formalise group adjournment through a closure activity.

A 4.2.4 Four Component Instructional Design Model (4C/ID*)

The four-component instructional design model (for short, 4C/ID* model; van Merriënboer, 1997; van Merriënboer, Clark, & de Croock, 2002) is an ID model for the design of competency based training



programs. The model takes real-life tasks as the driving force for learning. The general assumption is that such tasks help learners to integrate the knowledge, skills and attitudes necessary for effective task performance; give them the opportunity to learn to coordinate constituent skills that make up complex task performance; and eventually enable them to transfer what is learned to their daily life or work settings (Clark & Estes, 1999).

The model starts from an analysis of desired performance, stressing the importance of integrated performance objectives. In particular, complex skills require the coordinated performance of recurrent constituent skills or “routines”, which are highly consistent over problem situations, and non-recurrent constituent skills such as knowledge-based problem solving and reasoning, which vary over problem situations. Thus, it makes no sense to make a strict distinction between declarative and procedural learning because they are both an integrated part of complex learning.

Next, a training blueprint is designed, with as a base a sequence of whole task practice situations that support integration and coordination of the constituent skills. The blueprint design explicitly aims at performance improvement. There is no interest in memorization and knowledge construction for its own sake, but primarily in the ability to apply this knowledge to cope with the demands of real-life task performance.

A basic assumption of the 4C/ID* methodology is that environments for complex learning can be described in terms of four interrelated blueprint components (van Merriënboer, Clark, & de Croock, 2002):

1. Learning tasks, which are the actual tasks the learners will be working on during the training program. Learning tasks are organized in a simple-to-complex sequence of task classes, that is, categories of equivalent learning tasks. Learning tasks within the same task class start with high built-in learner support, which disappears well before the end of the task class (i.e., a process of “scaffolding”).

2. Supportive information, which is helpful to the learning and performance of aspects of the learning tasks that require variable performance over problem situations. It explains how a domain is organized, how to approach tasks or problems in this domain, and provides cognitive feedback on the quality of task performance.

3. Just-in-time information, which is information that is prerequisite to the learning and performance of aspects of learning tasks that show invariant performance over problem situations. It provides algorithmic specifications of how to perform those aspects.

4. Part-task practice, which provides additional repetitive practice for selected constituent skills that need to be performed at a very high level of automaticity after the training. It is only necessary if the learning tasks do not provide enough repetition to reach the desired level of automaticity.

The bottom part of Figure 4.1. shows a schematic view of the four components. The learning tasks are represented as circles (numbered “3”), organized in task classes (the dotted boxes around a set of learning tasks), and showing a decrease of support within task classes (indicated by the dark filling of the circles). The supportive information is represented in the L-shaped, light gray figures that are connected to the task classes (numbered “6”) and may also contain cognitive feedback (CFB). The just-in-time information is represented in the dark gray rectangles, with upward arrows that indicate that units of just-in-time information are connected to separate learning tasks (numbered “9”). Finally, part-task practice is represented by sequences of small circles (i.e., practice items; numbered “10”).

The 4C/ID* methodology can be described as an organized set of ten activities that may help to create a detailed training blueprint. Four activities pertain to the design of the blueprint components described above: the design of learning tasks (3), supportive information (6), just-in-time information (9), and part-task practice (10). The other six activities are preparatory and analytical in nature. They provide the input necessary for the design activities. The first activity, decompose the complex skill (1), is concerned with identifying the constituent skills and their interrelationships. The result is a so-called intertwined skills hierarchy. The goal of the second activity, sequence task classes (2), is to make a first rough training design, by specifying a simple-to-complex sequence of categories of learning tasks characterizing authentic problem situations. These first two activities form the basis for the design of learning tasks, which completes the skeleton of the training blueprint.

The remaining activities flesh out this skeleton. The activities analyze mental models (4) and analyze cognitive strategies (5) are concerned with the identification and description of knowledge structures that may be helpful to perform the so-called non-recurrent constituent skills, that is, the skills that require variable performance over problem situations. The activities analyze rules and procedures (7) and analyze prerequisite knowledge (8) result in the identification and description of the knowledge that must be presented to the learners because it enables the performance of so-called recurrent skills, that is, the skills that show identical performance over problem situations.

Applying the 4C/ID* methodology for the design of competency based training is not an easy task. The amount of intermediate and final products that are produced during analysis and design is large and those products are highly interrelated. As a result it is easy to loose the overview over the complete design process, which may impair decision-making. Therefore, in a European Community funded project called Advanced Design Approach for Personalized Training – Interactive Tools (ADAPTIT), currently a set of software tools that will help designers to apply the 4C/ID* methodology is being developed (De Croock, Paas, Schlanbusch, & van Merriënboer, 2002). A first tool, CORE, supports the analysis of a complex skill and the design of a competency based training blueprint. A second accompanying tool, EVAL, supports the evaluation of the training program and the subsequent revision of the blueprint on which the training program is based.

A 4.3 Assessment

A 4.3.1 Situation and prognosis

When comparing the instructional design guidelines in the four discussed ID models, it is clear that all models are problem-based. All models give authentic, real life problems that learners have to solve a central role in their proposed learning environments. When comparing the ID models with respect to instructional design guidelines for implementing, Merrill’s (2002) four First Principles of Instruction, it becomes apparent that in general all models are in agreement with Merrill’s first principles. None of the models includes principles that are in contradiction with Merrill’s principles but they do tend to emphasize different principles.

The table below provides an overview of the extend in which the different models provide guidelines for the design of instruction that follow first principles.

	ID models			
First principles	CLE Jonassen	GBS Schank	CPS Nelson	4C/ID Van Merriënboer
Activation	+/-	+	+	+/-
Demonstration	+	+	+/-	+
Application	+	+	+	+
Integration	+/-	+/-	+	+

A minus sign (-) indicates a model does not address a first principle, a plus/minus sign (+/-) indicates a first principle is partly or indirectly addressed, and a plus (+) sign indicates a first principle is fully addressed. For a more detailed discussion the reader is referred to Merrill's original publication. Nelson emphasizes collaboration, which is also stressed by Jonassen. According to Merrill, collaboration is an important method for implementing the principles of integration and activation, but is in itself not a first principle. Jonassen further emphasizes problem solving in learning environments. Van Merriënboer emphasizes problem sequence and the sequence of supporting information. Schank emphasizes stories (a form of demonstration) and problem solving (cases).

A 4.3.2 Conclusions for ALFANET

From the viewpoint of the developing ALFANET the most important requirements for an ID model are that it will allow for the creation of contents that 1) are effective, 2) are adaptable to learners, 3) allow for collaborative learning, and 4) provide clear, complete and broadly applicable instructional design guidelines.

With regard to the effectiveness of an ID model a first conclusion that can be drawn is that there appears to be a consensus among ID scientists that an effective learning environment is a problem based learning environment; all described models are problem based. According to Merrill effective ID models should address all first principles of instruction for each phase of the activation-demonstration-application-integration learning cycle. From the assessment described above it can be concluded that all four ID models incorporate many but not all first principles of instruction. In this respect Jonassen's CLE is estimated to be the least effective model, because it incorporates the least amount of first principles. The remaining three models 4C/ID, GBS, and CPS each appear to have weaknesses in different phases of the learning cycle. A combination and fusion of these models might be the basis for a highly effective set of ID guidelines.

With respect to adaptability to learners, neither of the models appears to have aspects that would disallow an adaptive instructional process. Of the four models only the 4C/ID model provides guidelines for how an instructional program designed according to the model could be adapted to individual learners. These guidelines are described in appendix 3, section 3.3.1: ADAPTit: the TASKi module.

With regard to collaborative learning it should first be noted that according to Merrill collaborative learning is not a first principle of instruction and should therefore not be a starting point for instructional design. Clearly what is lacking in current instructional design theories are guidelines that state the conditions when collaborative learning is useful and when it is not. Of the four models CLE and CPS emphasize collaborative learning and take it as a starting point for their instructional design. The other two models do not address it but also do not disallow it.

With regard to the clearness, completeness and applicability of the guidelines, van Merriënboer's 4C/ID model is probably the most comprehensive recent model of instructional design that involves all of Merril's first principles. The model focuses on the design of training for learning a skill, it does not address instructional design for learning a body of knowledge. A strong point of the model is that it provides clear guidelines for the design of a simple to complex sequence of practice conditions in which learners practice authentic, learning tasks to gradually develop their skill to the required level. The information needed to perform the skill is designed, partitioned and presented in such a way that it optimally scaffolds the performance of the skill. Another strong point is that an authoring tool for developing 4C/ID training designs is already being developed. Schank's GBS primarily focuses on the design of learning by doing simulations, which could be compared to the learning task component of the 4C/ID model. Nelson's CPS takes collaborative learning as a starting point. In terms of the 4C/ID model it could be viewed upon as a ID-model for the design of collaborative learning tasks. The guidelines in Jonassens' CLE are less clear then in the other models. The conditions for learning that should be achieved are described at a abstract, higher level

and the guidelines for achieving the conditions are less directive then in the other models. Therefore, as also stated above, for the ALFANET system a combination and fusion of the models: 4C/ID, GBS and CPS might be a good basis for specifying instructional design guidelines that covers all of the requirements of the ALFANET system with respect to instructional design.

A 4.4 References

Clark, R. E., & Estes, F. (1999). The development of authentic educational technologies. Educational Technology, 39(2), 5-16.

De Croock, M. B. M., Paas, F. G. W. C., Schlanbusch, H., & van Merriënboer, J. J. G. (2002). ADAPTIT: ID Tools for Training Design and Evaluation. Educational Technology, Research and Development.

Jonassen, D. (1999). Designing constructivist learning environments. In: C.M. Reigeluth (Ed.), In C. M. Reigeluth (Ed.), Instructional design theories and models: A new paradigm of instructional theory (Vol. 2) (pp. 215-239). Mahwah, NJ: Lawrence Erlbaum Associates.

Kommers, P., Jonassen, D., & Mayes, T., (1992). Cognitive tools for learning. Heidelberg, Germany: Springer Verlag.

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

Nelson, L.M. (1999). Collaborative problem solving. In: C.M. Reigeluth (Ed.), In C. M. Reigeluth (Ed.), Instructional design theories and models: A new paradigm of instructional theory (Vol. 2) (pp. 241-267). Mahwah, NJ: Lawrence Erlbaum Associates.

Schank, R.C., Berman, T.R., & Macpherson, K.A. (1999). Learning by doing. In: C.M. Reigeluth (Ed.), In C. M. Reigeluth (Ed.), Instructional design theories and models: A new paradigm of instructional theory (Vol. 2) (pp. 161-181). Mahwah, NJ: Lawrence Erlbaum Associates.

Van Merriënboer, J. J. G., Clark, R. E., & de Croock, M. B. M. (2002). Blueprints for complex learning: The 4C/ID* model. Educational Technology, Research and Development, 50(2).

Van Merriënboer, J. J. G. (1997). Training complex cognitive skills: A four-component instructional design model for technical training. Englewood Cliffs, NJ: Educational Technology Publications

Appendix 5 Web Mining

A 5.1 Overview

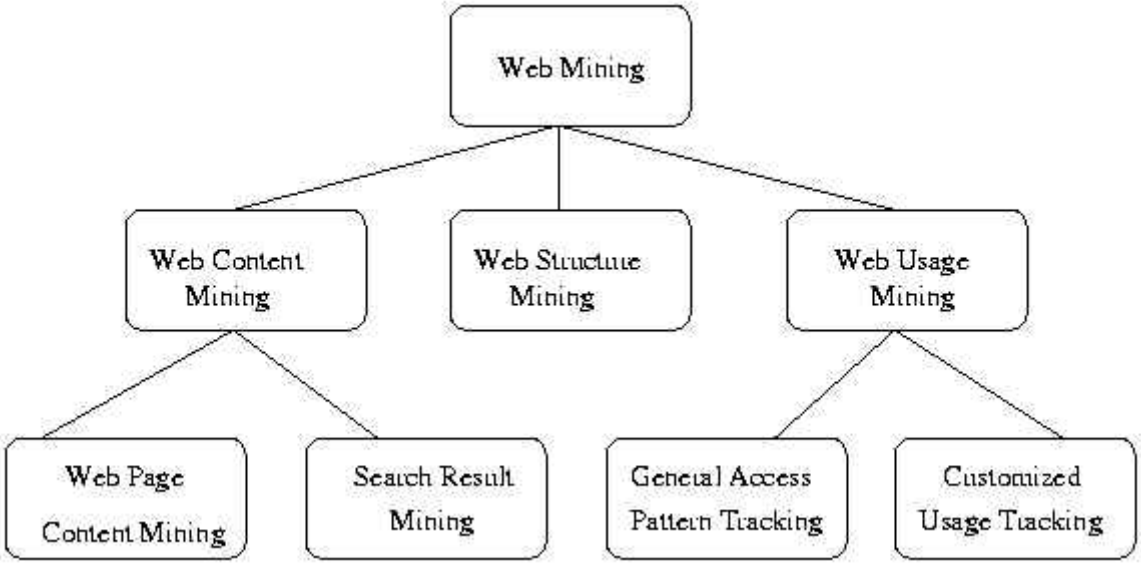
Web Mining is the extraction of interesting and potentially useful patterns and implicit information from artifacts or activity related to the World Wide Web. One of the key points of the system to be developed in aLFanet Project is adaptation in its three dimensions, Content Level Adaptation, Navigation Level Adaptation and Collaboration Level Adaptation. To achieve adaptation in our system, as much information as possible about the user activity and behavior has to be collected and analysed by the system. Web mining techniques focus on content, structure and usage mining, and they may be useful for processing the available information.

The Web mining research is at the cross road of research from several research communities, such as database, information retrieval, and within AI, especially the sub-areas of machine learning and natural language processing.

Currently there is an information overload on the Web, which makes difficult to find information, create new knowledge out of the information available, personalise this information and learn about individual users.

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services. Web mining tasks can be decomposed into **resource finding** (retrieving intended web documents), **information selection and pre-processing** (from retrieved web resources), **generalization** (discovering general patterns at web sites) and **analysis** (validation and/or interpretation of mined patterns) [Etzioni, 1996].

Depending on the part of the Web to mine, three Web mining categories can be distinguished [Madria, Bhowmick, Ng and Lim, 1999 and Borges and Levene, 1999], as the next figure shows:



- **Web content mining** (knowledge discovery in texts [Feldman and Dagan, 1995] and multimedia data mining [Zaiane, Han, Li, Chee and Chiang, 1998]): the discovery of useful information from the Web contents, data and documents. The goals are to improve the information finding or filtering the information to the users based on inferred user profiles (IR) and model data on the Web to integrate them to be used in sophisticated queries (DB).

  Web content mining is an automatic process that goes beyond keyword extraction. Since the content of a text document presents no machinereadable semantic, some approaches have suggested to restructure the document content in a representation that could be exploited by machines. The usual approach to exploit known structure in documents is to *use wrappers to map documents to some data model*. Techniques using lexicons for content interpretation are yet to come. There are two groups of web content mining strategies: *those that directly mine the content of documents and those that improve on the content search of other tools like search engines*.

- **Web structure mining** [Chakrabarti, Dom, Gibson, Kleinberg, Kumar, Raghavan, Rajagopalan and Tomkins, 1999]: the discovery of the model underlying the link structures on the Web. Used to categorize and to discovery authority sites. The line of research is inspired by the study of social networks and citation analysis. Some algorithms are HITS [Kleinberg, 1998] and PageRank [Brin and Page, 1998]. Some application models are web pages categorization, discovering micro communities, frequency of local links, replication of web documents and hierarchy of hyperlinks.

  WorldWide Web can reveal more information than just the information contained in documents. For example, links pointing to a document indicate the popularity of the document, while links coming out of a document indicate the richness or perhaps the variety of topics covered in the document. This can be compared to bibliographical citations. When a paper is cited often, it ought to be important. The PageRank [Brin and Page, 1998] and CLEVER [Chakrabarti, Dom, Gibson, Kleinberg, Kumar, Raghavan, Rajagopalan and Tomkins, 1998] methods take advantage of this information conveyed by the links to find pertinent web pages. By means of counters, higher levels cumulate the number of artifacts subsumed by the concepts they hold. Counters of hyperlinks, in and out documents, retrace the structure of the web artifacts summarized.

- **Web usage mining** [Cooley, Mobasher and Srivastava, 1997]: tries to make sense of the data derived from interactions of the users while interacting with the Web (web server access logs, proxy server logs, browser logs, user profiles, registration data, user sessions, cookies, user queries, bookmark data, mouse clicks and scrolls, ...). There are two approaches, mapping the usage data into relational tables before an adapted data mining technique is performed and using the log data directly by utlizing special pre-processing techniques. Typical data mining methods plus composite association rules, an extension or a traditional sequence discovery algorithm and hypertext probabilistic grammars are used. Often it uses some background or domain knowledge such as navigation templates, web content, site topology, concept hierarchies and syntatic constraints. Applications can be classified into two main categories, learning a user profile or user modeling in adaptive interfaces to learn user's needs and preferences, and learning user navigation patterns to adapt the site design to user's behavior.

  Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the web access logs of different web sites can help understand the user behaviour and the web structure, thereby improving the design of this colossal collection of resources. There are two main tendencies in Web Usage Mining driven by the applications of the discoveries:

  - **General Access Pattern Tracking:** analyzes the web logs to understand access patterns and trends. These analyses can shed light on better structure and grouping of resource providers. Many web analysis tools exist but they are limited and usually unsatisfactory. Techniques for using data mining and OnLine Analytical Processing (OLAP) on treated and transformed web access files have been proposed (WebLogMiner [Zaiane, Xin and Han, 1998]). Applying data mining techniques on access logs unveils interesting access patterns that can be used to restructure sites in a more efficient grouping, pinpoint effective advertising locations, and target specific users for specific selling ads.
  - **Customized Usage Tracking:** analyzes individual trends. Its purpose is to customize web sites to users. The information displayed, the depth of the site structure and the format of the resources can all be dynamically customized for each user over time based on their access patterns.

The following is an overview of the Web Mining categories [Kosala and Blockeel, 2000]:

	Web Mining			
	Web Content Mining		**Web Structure Mining**	**Web Usage Mining**
	IR View	**DB View**		
View of data	Unstructured Semi-structured	Semi-structured Website as a DB	Links structure	Interactivity
Main data	Text documents Hypertext documents	Hypertext documents	Links structure	Sever logs Brower logs

	Web Mining			
	Web Content Mining		Web Structure Mining	Web Usage Mining
	IR View	DB View		
Representations	Bag or words, n-grams Terms, phrases Concepts or ontology Relational	Edge-labeled graph (OEM) Relational	Graph	Relational table Graph
Method	TFIDF and variants Machine learning Statistical (including NLP)	Propietary algorithms ILP (Modified) association rules	Propietary algorithms	Machine Learning Statistical (Modified) association rules
Application Categories	Categorization Clustering Finding extration rules Finding patterns in text User modeling	Finding frequent substructures Web site schema discovery	Categorization Clustering	Site construction, adaptation and management Marketing User modeling

Web mining is often implemented within an agent paradigm, using the ***content-based approach*** (the system searches for items that match based on an analysis of the content using the user preferences) applied mainly for Web content mining, or the ***collaborative approach*** (the system tries to find users with similar interest to give recomendations to by analyzing user profiles and sessions) for Web usage mining [Balabanovic and Shoham, 1997]. Agents relevant for web mining tasks are ***user-interface agents*** [Delgado, 2000] (information retrieval agents, information (content/reputation/collaborative or social/event)-based or hybrid filtering agents, personal assistant agents) that try to maximize the productivity of current users interaction with the system by adapting behavior (personalization) and ***distributed agents*** [Kargupta, Hamzaoglu and Stafford, 1997] which are used for knowledge discovery and data mining.

Agent paradigm categories	Web mining categories
Content-based filters	Content mining
Reputation-based filters	Structure and content mining
Collaborative or social-based filters	Usage mining
Event based filters	Usage mining
Hybrid filters	Combination of categories

A 5.2 Solutions

Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. This broad definition on the one hand describes the automatic search and retrieval of information and resources available from millions of sites and on-line databases, i.e., *Web content mining*, and on the other hand, the discovery and analysis of user access patterns from Web usage data, i.e., *Web usage mining*. The research in this area has progressed along the following three dimensions.

A 5.2.1. Automatic Web Personalization Through Usage Mining

[Mobasher, Cooley and Srivastava, 2000]

Web personalization can be described, as any action that makes the Web experience of a user personalized to the user's taste, and the actions can range from simply making the presentation more pleasing to an individual to anticipating the needs of the user and providing the right information as well as performing a set of routine book-keeping functions automatically.

Principal elements of Web personalization include modeling of Web objects (pages, etc.) and subjects (users), categorization of objects and subjects, matching between and across objects and/or subjects, and determination of the set of actions to be recommended for personalization. Existing approaches used by many Web-based companies, as well as approaches based on collaborative filtering rely heavily on getting human input for determining the personalization actions. The drawbacks of this are:

- the input is often a subjective description of the users by the users themselves, and thus prone to biases, and
- the profile is static, and thus good for personalization for some time after it is collected; but its performance degrades over time as the profile ages.

Recently, a number of approaches have been developed dealing with specific aspects of Web usage mining for the purpose of automatically discovering user profiles:

- optimizing the structure of Web sites based co-occurrence patterns of pages within usage data for the site
- techniques for using path profiles of users to predict future HTTP requests, which can be used for network and proxy caching
- data mining techniques to extract usage patterns from Web logs, for the purpose of deriving marketing intelligence
- clustering of user sessions to predict future user behavior

This approach to usage-based Web personalization takes into account the full spectrum of Web mining techniques and activities, and is described by the architecture shown in the figure, which heavily uses data mining techniques, thus making the personalization process both automatic and dynamic, and hence up-to-date. Specifically, techniques for preprocessing of Web usage logs and grouping URL references into sets called *user transactions* (semantic unit of activity in which performing data mining is more meaningful) have been developed.

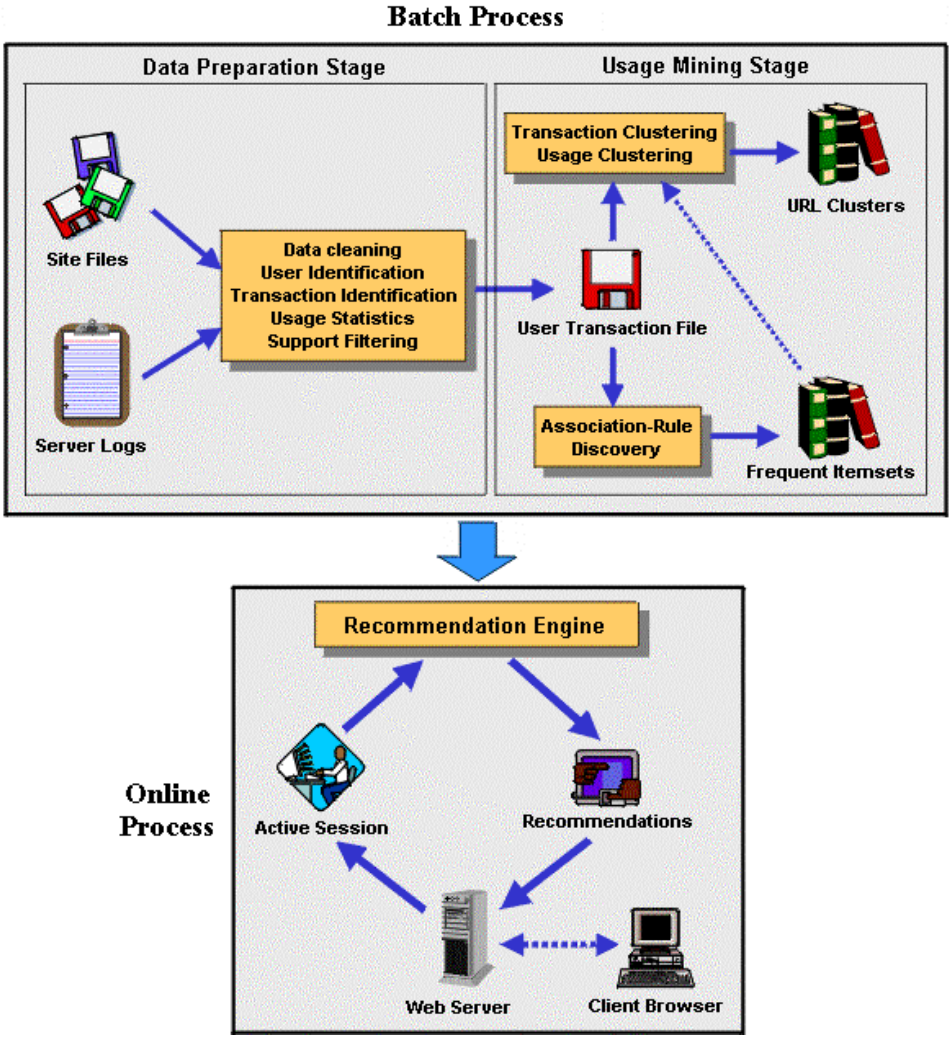
There exist three different Web usage mining techniques, based on **transaction clustering**, **usage clustering**, and **association rule discovery**, to extract usage knowledge for the purpose of Web personalization. Techniques for combining this knowledge with the current status of an ongoing Web activity to perform real-time personalization are used.

Architecture for Usage-based Web Personalization

The overall process of usage-based Web personalization can be divided into two components. The **offline component** is comprised of the data preparation tasks resulting in a user transaction file, and the specific usage mining tasks, which involve the discovery of association rules and the derivation of URL clusters based on two types of clustering techniques.

Once the mining tasks are accomplished, the frequent itemsets and the URL clusters are used by the **online component** of the architecture to provide dynamic recommendations to users based on their current navigational activity. The online component is comprised of a recommendation engine and the HTTP server. The Web server keeps track of the active user session as the user browser makes HTTP requests. This can be accomplished by a variety of methods such as URL rewriting, or by temporarily caching the Web server access logs. The recommendation engine considers the active user session in conjunction with the URL clusters and the discovered association rules to compute a set of recommended URLs. The recommendation set is then added to the last requested page as a set of links before the page is sent to the client browser.

A generalized architecture for the system is depicted in the figure:



The offline component of usage-based Web personalization can be divided into two separate stages. The first stage is that of preprocessing and data preparation, including, data cleaning, filtering, and transaction identification. The second is the mining stage in which usage patterns are discovered via methods such as association-rule mining and clustering. Each of these components is discussed below.

Preprocessing tasks

The prerequisite step to all of the techniques for providing users with recommendations is the identification of a set of user sessions from the raw usage data and cleaning the server log. Transaction identification is needed to dynamically create meaningful clusters of references for each user and filtering is used to remove noise from the data, and to achieve dimensionality reduction in clustering tasks.

Usage Mining stage

In this stage usage patterns are discovered by association-rule mining and clustering.

- **Association Rules methods**: finds group of items occurring frequently together in many transactions (frequent itemsets) and use *association rules* to capture the relationships among items based on their patterns of co-occurrence across transactions.
- **Clustering Transactions**: clusters user transactions based on mined information from access logs
- **Usage Clusters**: computes clusters based on how often they occur together accross user transactions.

Recommendation Process

The task of the recommendation engine is to compute a recommendation set for the current session, consisting of links to pages that the user may want to visit based on similar usage patterns. The recommendation set essentially represents a "short-term" view of potentially useful links based on the user's

navigational activity through the site. These recommended links are then added to the last page in the session accessed by the user before that page is sent to the user browser.

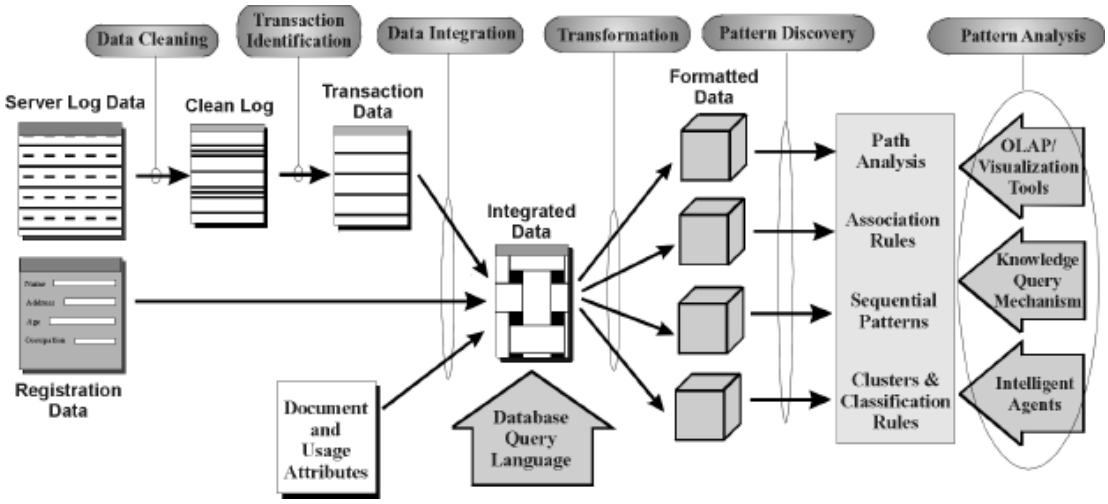
Computing recommendations can be done directly from frequent itemsets or based on URL clusters.

A 5.2.2. Pattern Discovery from World Wide Web Transactions

[Mobasher, B., Jain, Han and Srivastava, 1996]

This research involves the development of a general and flexible framework for Web usage mining, the application of data mining techniques, such as the discovery of association rules and sequential patterns, to extract relationships from data collected in large Web data repositories. The proposed framework includes a modular architecture for the Web mining process which distinguishes between *the domain dependent data transformation tasks*, such as the discovery and identification of several types of *user transactions*, and *the generic data mining engine*, and data and transaction models for each of these components.

The type of knowledge discovered can be used, for instance, to restructure a Web site for increased effectiveness, for better management of workgroup communication in intranets, and in analyzing user access patterns to dynamically present information tailored to specific groups of users. The WEBMINER system, based on the proposed framework and data models to discover association rules, sequential patterns, and classification rules from WWW data. The general architecture for WEBMINER is depicted in the figure.



In the context of web data *user transactions* have to be indentified to allow the discovery phase to focus on relevant access points of a particular user rather than the intermediate pages accessed for navigational reasons. Special algorithms must be used to identify unique user sessions and to find user transactions, since generally references are not uniquely identified by user and many of the references are cached by client-side agents or proxy servers. A combination of standard methods such as client-side cookies and heuristics are used to identify unique user sessions. The heuristics include using IP, agent, and OS fields as key attributes; using session time-outs; using synchronized referrer log entries to expand user paths belonging to a session; and using sophisticated algorithms to infer cached references by completing and disambiguating user paths belonging to a session.

Once unique user sessions are identified, grouping user references into transactions must use information about both the *nature of the data* and the *type of analysis* to be done, using such information in a 2-step process.

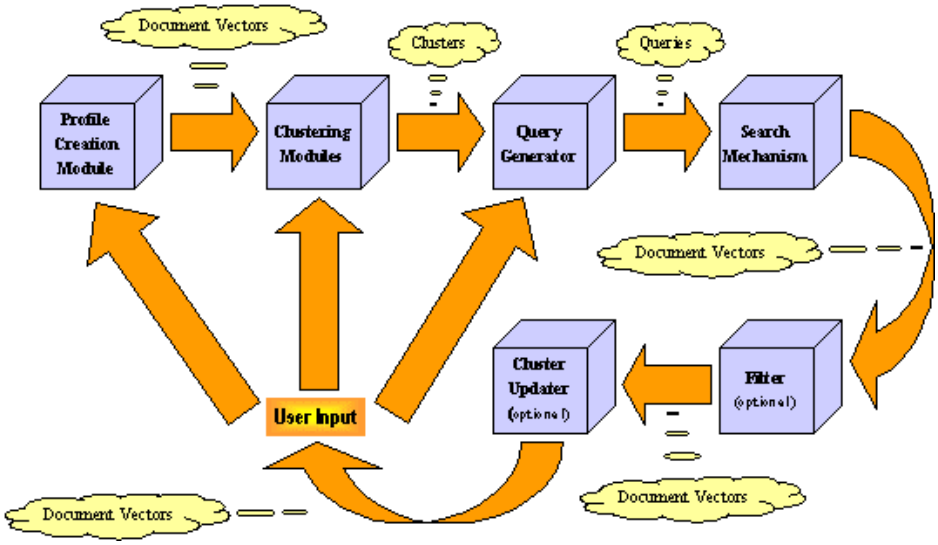
- In the first step clustering is used as a general approach to grouping references into transactions. The clustering is based on comparing pairs of log entries and determining the *similarity* between them by means of some kind of distance measure(s). WEBMINER also uses a model of user browsing behavior and statistical techniques to atomatically determine if a particular user treats individual references as *content* or *navigational* references,
- In the second step, information about the type of analysis is used and the groups formed in step 1 are specialized into transactions suited to the specific analysis. In using clustering to determine the similarity of two references, i.e. whether they belong to the same group, distance metrics on many different attributes can be defined. Determining an appropriate set of attributes to cluster on, and defining appropriate distance metrics for them is an important problem.

A 5.2.3. Document Categorization and Exploration

[Mobasher, Han, Boley, Gini, Gross, Hastings, Moore, Karypis and Kumar, 1998]

WebACE is an intelligent agent that can retrieve, filter, and categorize web documents guided by a user profile. At the heart of this system are two novel clustering algorithms, based on hypergraph partitioning and principal component analysis, which are very effective in quickly clustering a high dimensional space, and can be used to automatically filter and categorize documents. Based on characteristics of the discovered clusters, the agent can also generate effective search queries used to search for new related documents and filtering the resulting documents to extract the set of documents most closely related to the starting set. The document categories are not given a priori. The resulting document set could also be used to update the initial set of documents. The new clustering algorithms provide a significant improvement in performance over traditional clustering algorithms used in information retrieval, such as the distance-based *Hierarchical Agglomerative Clustering* (HAC) and the probability-bases *AutoClass.*

Many traditional algorithms break down as the size of the document space, and thus the dimensionality of the corresponding feature space, increases. High dimensionality is characteristic of the type of information retrieval applications which are used to filter and categorize documents on the World Wide Web. In contrast, these partitioning-based algorithms do not rely on a specific choice of a distance function and do scale up effectively in a high dimensional space. The general architecture of WebACE is depicted in the next figure.



One of the main tasks of the agent is to search the Web for documents that are related to the clusters of documents. WebACE does this by deriving the optimum search query as a boolean formula based on the text frequency and the document frequency of important words within each cluster. The documents returned as the result of queries can be handled in several ways. For example, the Hypergraph Partitioning method could be used to filter out non-relevant documents. The degree of filtering can be increased either by setting higher support criteria for association rules discovery or by having a tighter connectivity constraint in the partition. Resulting documents can also be incrementally added to the existing clusters using data structures maintained as part of either of the clustering methods.

A 5.3 Assessment

A 5.3.1 Conclusions for ALFANET

Web mining techniques can be very useful for achieving several of the adaptation tasks proposed to be done by the system:

- Free-text or semi-structured analysis can be very useful in order to retrive information from news, forum messeges, etc. that can be used to learn the interests of each individual user, which will be used for content and collaboration adaptation
- Link mining can give support for navigation adaptation tasks
- Information extracted from interacitions of the users with the Web is also useful to construct the user model used for adaptation.

Therefore, Web Content Mining (process of extracting knowledge from the content of documents or their descriptions), Web Structure Mining (process of inferring knowledge from the WorldWide Web organization and links between references and referents in the Web) and Web Usage Mining (process of extracting interesting patterns in web access logs) techniques could be used in aLFanet Project, although a more detailed analysis is necessary to confirm its viability and utility in the system.

Finally, it is important to know that the success depends on what and how much valid and reliable knowledge one can discover from the large raw log data, and that for an effective web usage mining, an important cleaning and data transformation step before analysis has to be done.

A 5.4 References

http://www.cs.ualberta.ca/~tszhu/webmining.htm

http://maya.cs.depaul.edu/~mobasher/Research00.html

Balabanovic, M. and Shoham, Y (1997) Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66-70.

Borges, J. and Levene, M. (1999) Data Mining of user navigation patterns. In *Proceedings of the WEBKDD'99 Workshop on Web Usage Analysis and User Profiling*, pags. 31-36.

Brin, S. and Page, L. (1998) The anatomy of a large-scale hypertextual web search engine. In *7th Int. Conf. WWW*, Brisbane, Australia.

Chakrabarti, S., Dom, B., Gibson, D., Kumar, S. Raghavan, P., Rajagopalan, S. and Tomkins, A. (1998) Experiments in topic distillation. In ACM SIGIR workshop on Hypertext Information Retrieval on the Web, Melbourne, Australia.

Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Kumar, S. Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) Mining the link structure of the world wide web. *IEEE Computer*, 32(8):60-67.

Cooley, R., Mobasher, B. and Srivastava, J. (1997) Web mining: Information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97).*

Delgado, J.A. (2000) *Agent-Based Information Filtering and Recommender System On the Internet*. PhD thesis, Dept. Of Intelligence Computer Science, Nagoya. Institute of Technology.

Etzioni, O. (1996) The world wide web: Quagmire or gold mine. *Communications of the ACM*, 39(11): 65-68.

Feldman, R. and Dagan, I. (1995) Knowledge discovery in textual databases (KDT). In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pags. 112-117. Montreal, Canada.

Kargupta, H., Hamzaoglu, I. and Stafford, B. (1997) Distributed data mining using an agent based architecture. In Proceedings of Knowledge Discovery And Data Mining. AAAI Press, pags. 211-214.

Kleinberg, J. (1998) Authoritative sources in a hyperlinked environment. In *Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms,* pages 668-677, ACM Press, New York.

Kosala, R. and Blockeel, H. (2000). *Web Mining Research: A Survey*. ACM SIGKDD. Vol. 2, Issue 1.

Madria, S.K., Bhowmick, S.S., Ng, W.K. and Lim, E.P. (1999) Research issues in web data mining. In *Proceedings of Data Warehousing and Knowledge Discovery, First International Conference, DaWaK'99*, pags. 303-312. San Diego, USA.

Mobasher, B., Cooley, R. and Srivastava, J. (2000) Automatic Personalization Based on Web Usage Mining Communications of the ACM. , 43(8), pp. 142-151

Mobasher, B., Jain, N., Han, E. and Srivastava, J. (1996) WEBMINER: Pattern Discovery from World Wide Web Transactions. *Technical Report TR96-050*, Department of Computer Science, University of Minnesota.

Mobasher, B., Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Moore, J., Karypis, G. and Kumar, V. (1998) WebACE: A Web Agent for Document Categorization and Exploration. In *Proceedings of the 2nd* International Conference on Autonomous Agents (Agents'98).

Zaiane, O.R., Xin, M. and Han, J. (1998) Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. *Proceedings from the ADL'98 - Advances in Digital Libraries*, Santa Barbara.

Zaiane, O.R., Han, J., Li, Z.N., Chee, S.H. and Chiang, J. (1998) Multimediaminer: a system prototype for multimedia data mining. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pags. 581-583.

Appendix 6 Machine learning

A 6.1 Overview

A 6.1.1 Definition

The field of machine learning concern both cognitive psychology and artificial intelligence, and studies the computational processes that underlie learning in both humans and machines. Machine learning cannot ignore issues of knowledge representation, memory organization and performance, which are central concerns for artificial intelligence and cognitive science [Langley, 1996]. Machine learning is concerned with developing computational theories of learning processes and building machines that learn [Gilmore and Self, 1988].

Learning is the improvement of performance in some environment through the acquisition of knowledge resulting from experience in that environment [Langley, 1996].

A popular definition of learning itself is a "change in a system that allows it to perform better the second time on repetition of the same task or on another task drawn from the same population" [Simon 1983].

The key element of this definition of learning is skill refinement. In living systems this skill refinement is achieved through the acquisition and application of knowledge. All intelligent organisms must be able to learn in order to adapt and, at a more fundamental level, survive. Consequently, the ability to learn is seen as a key indicator of intelligence and so, a basic requirement of artificially intelligent systems. Given the enormity and often changing nature of the tasks that Artificial Intelligence systems need to conquer, the ability to learn is seen as fundamental.

A 6.1.2 Some issues about Machine Learning

[Langley, 1996]

One-step vs. Multi-step techniques

An important distinction in machine learning concerns whether one uses the learned knowledge for one-step classification and prediction or for some form of multi-step inference or problem solving. Most research has focused on the former performance task, but techniques developed for classification tasks can often be adapted to learn knowledge for use in multi-step problem solving.

Supervised learning vs. Unsupervised learning

The degree of supervision also influences learning. In some cases, a domain expert gives the learner direct feedback about the appropriateness of its performance. Such supervised learning problems contrasts sharply with unsupervised learning tasks, in which feedback is absent. The vast majority of research in machine learning has dealt with supervised tasks, but many methods designed for supervised problems can be adapted to unsupervised ones. Both of these tasks can arise in learning for either classification or problem solving. For classification problems, the supervised tasks assumes that each training instance includes an attribute that specifies the class of that instance, and the goal is to introduce a concept description that accurately predicts this attribute. There is less agreement on the goal of unsupervised learning, but one can define analogous prediction tasks over the entire set of attributes. In problem-solving tasks, supervised learning occurs when the domain expert suggests the correct step at each point in the search or reasoning process; systems that operate on such feedback are sometimes referred to as learning apprentices. However, most work on learning in problem solving has dealt with unsupervised tasks, in which the agent must distinguish desirable actions from undesirable ones on its own. This subtask has been called the credit assignment problem, since the learner must identify the decisions responsible for success or failure of its problem-solving efforts. Once the learner has dealt with this issue, it can draw directly on supervised methods for classification learning to acquire problem-solving knowledge.

Online learning vs. Offline learning

Yet another aspect of the environment concerns the manner in which training cases are given to the learner. Offline learning occurs when all instances are presented simultaneously, in contrast with online settings, in which the instances are presented one at a time. Intermediate schemes are also possible, in which the learner encounters one set of instances, followed by another set, and so on. Many of the learning tasks confronting humans appear to be online in nature, if only because people exist in a temporal world and thus

experience events in sequence. However, in some cases humans are confronted with a mass of pre-collected data. Thus, both forms of task model situations that occur in realistic situations and both have a role to play in a complete theory of machine learning. Nevertheless, most research has emphasized offline learning problems.

Regularity of the environment

One can identify four different but related environmental factors that can influence the difficulty of learning: 1) the complexity of the target knowledge that must be acquired, 2) the number of irrelevant features or attributes; if the environment contains many such features, the learning system can have difficulty distinguishing them from the relevant features that it should use in making predictions, 3) the amount of noise in the environment; this can take on two forms in supervised learning tasks: class noise, which involves the corruption of supervised feedback, so that the learner is given incorrect feedback, and attribute noise, which involves corruption of the instance description itself, thus, an attribute's value may be replaced or shifted, and 4) the consistency of the environment over time; in some cases, a learned concept may suddenly cease to be valid, though it may retain partial overlap with the new situation (such cases of concept drift can be difficult to distinguish from noise).

Knowledge representation

Both learning and performance rely on the ability to represent knowledge. Relevant issues refer to the representation of experience (the input to learning) and to the representation and organization of acquired knowledge (the output of learning). Moreover, the effect of knowledge cannot be separated from the processes that use it, so the learned knowledge is related to the performance component.

Representational bias

Machine learning must address the problem of induction, which states that generalizing from any set of observations is never logically justified, since there always exist many hypotheses that could account for the observed data. Clearly, a learning system must somehow limit or direct its search through the space of possible knowledge structures. The machine learning literature often refers to this as the bias of the system. One important form of constraint, known as representational bias, restricts the space of possible structures by limiting the concept description language. Other approaches are the notion of search bias, which consider all possible concept descriptions but examines some earlier that others, and the background knowledge available to the learning system.

Incremental algorithms vs. Non-incremental algorithms

It has been previously distinguished between offline learning tasks and online ones. A similar distinction holds for learning algorithms, which can either process many training instances at once, in a *non-incremental* manner, or handle them one at a time, in a *incremental* fashion.

A 6.1.3 Paradigms for machine learning

[Langley, 1996]

Machine learning is a diverse field that is held together by a common set of goals and similar evaluation methodologies. Despite these similarities, researchers in machine learning tend to associate themselves with one or another of five main paradigms, which differ in their basic assumptions about representation, performance methods, and learning algorithms.

One major paradigm, often termed neural networks, represents knowledge as a multilayer network of threshold units that spreads activation from input nodes through internal units to output nodes. Weights on the links determine how much activation is passed on in each case. The activations of output nodes can be translated into numeric predictions or discrete decisions about the class of the input. The neural net framework typically attempts to improve the accuracy of classification or prediction by modifying the weights on the links. The typical learning algorithm carries out a hill-climbing search through the space of weights, modifying them in an attempt to minimize the errors that the network makes on training data.

A second framework, known as instance-based or case-based learning, represents knowledge, in terms of specific cases or experiences and relies on flexible matching methods to retrieve these cases and apply them to new situations. One common approach simply finds the stored case nearest (according to some distance metric) to the current situation, then uses it for classification or prediction. The typical case-based learning method simply stores training instances in memory: generalization occurs at retrieval time, with the power residing in the indexing scheme, the similarity metric used to identify relevant cases, and the method for adapting cases to new situations.

Genetic algorithms constitute a third paradigm within machine learning. This framework typically represents acquired knowledge as a set of Boolean or binary features, which are sometimes used as the conditions an actions of rules. The most common interpreter for this knowledge employs a logical matching process, using strengths associated with rules to resolve conflicts. In some cases, a production-system architecture lets rules apply in sequence, producing multi-step behavior. The standard learning operators in genetic algorithms generate new candidate rules from parents that have high strengths, where strength reflects some measure of performance on training cases. In effect, genetic methods carry out a parallel hill-climbing search, retaining a set of competing an sometimes complementary descriptions in memory.

A fourth paradigm, which we will call rule induction, employs condition-action rules, decision trees, or similar logical knowledge, structures. Here the performance element sorts instances down the branches of the decision tree or finds the first rule whose conditions match the instance, typically using a logical matching process. Information about classes or predictions are stored in the action sides of the rules or the leaves of the tree. Learning algorithms in the rule-induction framework usually carry out a greedy search through the space of decision trees or rule sets, using a statistical evaluation function to select attributes to incorporate into the knowledge structure. Most methods partition the training data recursively into disjoint sets, attempting to summarize each set as a conjunction of logical conditions

A final approach, sometimes termed analytic learning, also represents knowledge as rules in logical form but typically employs a performances system, that uses search to solve multi-step problems. A common technique is to represent knowledge as inference rules, then to phrase problems as theorems and to search for proofs. Learning mechanisms in this framework use background knowledge to construct proofs or explanations of experience, then compile the proofs into more complex rules that can solve similar problems either with less search or in a single step. Most work on analytic learning has focused an improving the efficiency of the search process, but some has dealt with improving accuracy on classification tasks.

A 6.1.4 Machine learning methods

Machine learning methods can be classified into two big areas, non-symbolic, used in control processes, and symbolic, which are the ones we will focus on. Symbolic machine learning methods can be classified, in turn, into:

- Inductive methods, used to acquire general knowledge from specific examples. These methods are usually used in concept learning, that is, to construct a class definition. Concept learning algorithms often generate class definitions in the form of decision trees which are able to solve difficult problems of practical importance, and which need lots of training examples.
- Deductive methods, used when there are few examples, but a complete domain theory exists. These methods are not very useful by their own, because usually the theory of the domain is not complete, but they can be used to assure the results obtained and to find more efficient ways to solve the problems.
- Methods on demand (or lazy methods), which are used to solve problems in domains where there are a lot of examples but no theory. The basic algorithm for these methods is the k-NN.

All these methods have their own advantages and drawbacks, and it is difficult to choose the appropriate one when trying to solve a real problem. The best solution is to have the possibility of using the best method depending on the problem to be solved. This is the main feature of the multistrategic learning systems, in which the learning method can be chosen, usually manually.

Research on machine learning is a very active field. The came up of multiagent systems has extended the kind of problems that can be tackled. There are research groups trying to include machine learning into multiagent systems. The reason for this is clear: agents work on very dynamic environments and they need to adapt themselves if they want to be useful. Several issues have to be taken into account, namely the rol of each agent in the system or the interaction among agents. Two levels of learning have to be distinguished: a global one, which involves the improvement of the whole system, and a local one, that is the one achieved by the agent itself.

A 6.2 Solutions

A 6.2.1 General overview of Machine Learning Solutions

[MLnet]

Solutions in machine learning can be provided in terms of learning tasks, methods and algorithms, according to some theoretical setting. Each task can be related to one or more methods. Methods make use of a specific set of hypotheses and allow only for a specific kind of examples. While methods provide a more general view on how problems are addressed, algorithms are closer to implementation.

The next subsections give a classification of each of these topics, and shows the interrelation among them.

A 6.2.1.1 Learning Tasks

Learning tasks are formal definitions of "what we want to learn". In order to find the learning method suitable for a certain application, it is neccessary to identify the formal task that covers the learning problem. These are:

Association Rule Learning

Association rules describe correlation of events and can be regarded as probabilistic rules. Correlation of events means, that events are frequently observed together. Discovering association rules in large databases can be a first step of knowledge discovery in databases (KDD).

Methods: APRIORI

Characterization (Descriptive Setting)

The characterization task in Inductive Logic Programming is about finding interesting regularities in datasets. If you have a complex dataset, it is often impossible to directly see such regularities in the data. Having a description of the same dataset, represented in a readable formalism (e.g. first order logic rules), often makes finding interesting regularities a much easier task.

Methods: GOLEM, PROGOL

Theories: Inductive Logic Programming (ILP)

Clustering (Unsupervised Learning)

The task of clustering is to structure a given set of unclassified instances of an example language by creating concepts, based on similarities found on the training data. So the main difference to supervised learning is, that there is neither a target predicate nor an oracle, dividing the instances of the training set into categories. The categories are formed by the learner itself.

Methods: Star, COBWEB

Concept Learning

The task of concept learning is to acquire general concepts from specific training examples. Training examples are instances, which either belong to a special concept, and therefore are positive examples, or do not belong to the concept, and therefore are negative examples.

Methods: Lazy Learning, Star, Searching Version Spaces, Top-Down Induction of Decision Trees (TDIDT), Top-Down Induction of Horn Clauses, Bottom-Up Induction of Horn Clauses, Naive Bayes, Boosting, Logistic Regression, Support Vector Machine (SVM), GOLEM, PROGOL, Incremental decision tree learning

Theories: Bayesian Learning, Inductive Logic Programming (ILP), Learning as Search, Probably Approximately Correct (PAC-) Learning, Statistical Learning

Function Approximation

Many general learning tasks, especially concept learning, may be regarded as function approximation.

Given are examples of the function to be approximated. The aim is to find a hypothesis (a function as well), that can be used for predicting the function values of yet unseen instances, e.g. to predict future events.

Good hypotheses are those, often predicting an accurate function value. The quality of a hypothesis for approximating a specific function is measured by a so called loss function, increasing as the differences between predicted and true function values increase. It is also taken into account, that it is more important to predict well on frequently observed instances.

Methods: Neural Nets, Lazy Learning, Top-Down Induction of Regression Trees, Support Vector Machine (SVM)

Theories: Probably Approximately Correct (PAC-) Learning, Statistical Learning

Interesting Subgroups

The task of finding interesting subgroups is related to the task of characterization. We are looking for subsets of an instance space, with interesting properties. This differs from the tasks of concept learning and function approximation, because we are not trying to find a hypothesis, that globally describes the data and enables to predict unseen instances, but we focus on subsets of the data.

On-Line Learning

The learning task here is how to combine advice from several experts.

Methods: WEIGHTED-MAJORITY

Reinforcement Learning

In the reinforcement learning scenario the task is to find a policy that maximizes the reward of an agent by experimentation.

Methods: Q Learning

Theories: Learning as Optimization

Revision

The learning task is to revise a given theory, in order to no longer imply a given set of unwanted facts. In other words, a theory has become inconsistent and shall be adjusted to some new insights.

Theories: Gaerdenfors' Postulates

Sequence Prediction

The target is to find rules, accurately predicting the occurence of future events, given the event sequence up to the actual point in time.

Methods: Best Region Rules

Syntax/Grammar/Automata - Learning

Given an infinite stream of examples, where examples are either well-formed sentences of a language, or sentences classified as well-formed or not well-formed, and a class of languages, where for each language there exists a procedure to decide whether a sentence is member of this language, or a class of automata, where each automata can decide whether a given sentence is accepted the target is to find the language or the automaton that generates/accepts all well-formed sentences.

Theories: Identification In The Limit, Inductive Logic Programming (ILP)

Time Series Analysis

The goal is to predict the behaviour at point n+1 in time, which possibly may be determined by a function.

A 6.2.1.2 Learning Algorithms

The presented learning algorithms address different learning tasks. Some of them are based on the same more general learning method.

Some learning algorithms are: *APRIORI, AQn, AdaBoost, Best Region Rules, CANDIDATE-ELIMINATION, CART, COBWEB, FOIL, GOLEM, ID3, PROGOL, Q Learning, RDT, WEIGHTED-MAJORITY, k-NEAREST NEIGHBOR.*

A 6.2.1.3 Learning Methods

Several learning algorithms share the same general learning method. Each method addresses one or more learning tasks.

On the abstraction level of methods, the representation of examples and hypotheses will usually be determined. The appearance of examples is formally defined by an example language. The set of possible results of a method is given by its so called hypothesis language.

In learning tasks requiring hypotheses to generalize (sub-)sets of examples, there has to exist an efficient test of whether an arbitrary hypothesis from the hypothesis language covers an arbitrary example that is part of the example language. In this setting, each hypothesis defines a subset of the example language.

Some learning methods are: *Boosting, Bottom-Up Induction of Horn Clauses, First Order Regression, Incremental decision tree learning, Lazy Learning, Least Squares Linear Regression, Logistic Regression, Multivariate Adaptive Regression Splines, Naive Bayes, Neural Nets, Non-parametric Regression (also known as local modelling), Projection Pursuit Regression, Regression Rules, Searching Version Spaces, Star, Support Vector Machine (SVM), Top-Down Induction of Decision Trees (TDIDT), Top-Down Induction of Horn Clauses.*

A 6.2.1.4 Example Languages

To choose a hypothesis from their so called hypothesis language, most learning algorithms read a set of classified or unclassified examples. The representation of examples is determined by the choice of an example language.

Example languages are: *Attribute-Value, Clauses, Ground Facts, Numerical Values.*

A 6.2.1.5 Hypothesis Languages

For most learning algorithms there exists a set of possible results, the algorithm chooses from. The result is chosen such, that it fits the input data, provided by the user.

The set of possible results, the algorithm is allowed to choose from, is called its hypothesis language or hypothesis space. The choice of an adequate hypothesis language can be essential for the following reasons: 1) restricting the hypothesis language too much can decrease the quality of learning results, because it bears the risk of the learner not having good hypotheses left in its search space, for specific inputs, and 2) choosing a large set of possible results increases the chance, that the learner decides for a concept explaining the specific input data well, but simply for reasons of "statistical fluctuation", i.e. the data is not representative enough for the learner to distinguish between the set of most promising hypotheses. The problem in this case is, that the chance of performing worse on new, yet unseen data, relying on the result the learner presented, is higher.

Hypothesis languages are: Decision Trees, Functions, Probabilistic Categorization Tree, Regression Functions, Regression Trees, Restricted First Order Logic, Version Space.

A 6.2.1.6 Theoretical Settings

Theoretical settings are the theoretical scenarios and postulates used by machine learning researchers. Here scenarios and postulates are presented as well as some of the main questions research focusses on. The theoretical settings are closely related to learning methods and tasks.

Some theoretical settings are: Bayesian Learning, Gaerdenfors' Postulates, Identification In The Limit, Inductive Logic Programming (ILP), Learning as Optimization, Learning as Search, Probably Approximately Correct (PAC-) Learning, Statistical Learning.

A 6.2.2 Concept Learning Algorithms

Research in the field of symbolic machine learning has resulted in the development of a wide range of algorithms. Typically, learning in these algorithms is accomplished by searching through a space of possible hypotheses to find an acceptable generalisation of a concept. However, machine learning algorithms vary in their goals, learning strategies, the knowledge representation languages they employ and the type of training data they use [Smith, 1999].

Concept learning or Induction is the task of constructing a class definition. Concept learning algorithms often generate class definitions in the form of decision trees which are able to solve difficult problems of practical importance. A decision tree is a representation of a procedure for determining the class of a given instance [Utgoff, 1989] and consists of the following: a) Leaf or answer nodes that indicate a classification either positive or negative, and b) Non leaf or decision nodes which contain an attribute name and branches to other decision trees or leaf nodes, one for each value of the attribute.

The top-down induction of decision trees is an approach to decision tree building in which classification starts from a root node and proceeds to generate sub trees until leaf nodes are created. It is possible to categorise conjunctive and disjunctive descriptions of concepts with decision trees, and *if-then* rules can easily be lifted from the trees.

Some concept learning algorithms are described next.

A 6.2.2.1 Classification Algorithms

The classification algorithm is a non-incremental, supervised concept learning method that produces a hypothesis in the form of a decision tree. The algorithm accepts a training set of attribute based on positive and negative examples of a concept which must all be presented before learning commences, hence the algorithm is non-incremental. The classification algorithm proceeds by randomly selecting an attribute to add to the decision tree and branches are grown for each possible value of the attribute. The training examples are added to the tree and the classification of the examples at the node checked. If all the training examples at a node are positive or negative the node is labelled with that classification and becomes a leaf node of the tree. Otherwise this process is repeated recursively until all nodes are leaf nodes.

The ID3 algorithm

The classification algorithm randomly selects the order that attributes are added to the decision tree. By selecting attributes in a different order, different trees can be produced and some of these trees could be shallower than others. Shallower trees are ones in which the classification is reached in fewer levels. These trees are said to be more efficient as the classification is reached quicker. This point is addressed by Quinlan's ID3 algorithm [Quinlan, 1983], which is an enhancement of the classification algorithm previously described, which similarly produces a hypothesis in the form of a decision tree. However, the ID3 algorithm adds two new features to the basic classification algorithm. These are windowing and the information theoretic heuristic.

Windowing can be used if the training set is very large. A subset of the training set called the window is chosen randomly to build an initial tree. The remaining input cases are then classified using the tree. If the tree gives correct classification for these input cases then it is accepted for the entire training set and the process ends. If this is not the case then a selection of incorrectly classified instances are appended to the window and the process continues until the tree gives correct classification for the whole set.

The information theoretic heuristic is used to produce shallower trees by deciding the order in which to select attributes.

The C4.5 algorithm

C4.5 [Quinlan, 1993] is an advanced and incremental version of the ID3 algorithm that caters for unavailable values, continuous attribute value ranges, and decision tree pruning. A new features of C4.5 is the Gain ratio criterion, that allows the algorithm not only to select an attribute containing maximum information but also an attribute with minimal partitioning.

A 6.2.2.2 The Version Space algorithm or focusing algorithm

The focusing algorithm [Young, Plotkin and Linz, 1977] is considered to be a powerful technique to learn concepts. The algorithm aims to produce a definition that is consistent with all given positive training data, but none of the negative. The focusing algorithm uses a version space search though the concept space. The concept space covers all the possible concept descriptions. The version space only covers concept descriptions which are consistent with the given training instances. The focusing algorithm is similar to Mitchell's Candidate Elimination algorithm [Mitchell *et al*, 1986].

In the focusing algorithm the concept being learned is represented by a set of trees. There is one tree for each attribute used to describe the concept. An upper boundary is initially placed above each tree and a lower boundary is placed below. The version space lies between the upper and lower boundaries and constitutes the search space for the concept. Concepts are learned my moving the lower and upper boundaries upward and downwards respectively. Initially the version space covers the entire concept space.

A 6.2.2.3 Clustering Algorithms

Clustering algorithms are unsupervised concept learning algorithms. The learning process does not require pre-classified training data, and produces classifications based on a measurement of the degree of similarity between objects. This type of learning system organises unclassified objects into a hierarchy of classes by measuring the similarities between objects and gathering maximally similar objects into the same group or cluster.

The process of determining clusters can be done in either a bottom-up or top-down fashion. The bottom-up (hierarchical) method of clustering recursively combine single objects or groups of objects into larger groups until the original set of objects merge into one single category which located at the top of the hierarchy. The top-down (non-hierarchical) method recursively splits the original set of objects into subcategories until each object is assigned to a subcategory.

There are a number of techniques for calculating similarities between objects. The use of any of these methods depends on the relation between objects and groups of objects. This relationship might be object-to-group or group-to-group. If the clustering system is built on numerical similarity then the system cannot give any conceptual interpretation of the obtained categories; that is, they cannot give any explanation about clusters in human terms. Conceptual clustering systems attempt to overcome this.

As with other learning algorithms, clustering algorithms can be subdivided into non-incremental algorithms that rely on all the object descriptions being available prior to the clustering process and incremental algorithms that process objects as they become available.

Clustering algorithms are usually categorised according to the type of cluster structure they produce e.g. hierarchical or non-hierarchical, or according to the type of data description language they use e.g. statistical, based on numeric descriptors, or conceptual, based on symbolic descriptors.

The non-hierarchical clustering methods divide a set of N objects into M clusters; no overlap is allowed. These are also known as partitioning methods. The hierarchical clustering methods produce a nested data structure by recursively splitting groups of objects into subcategories until each object is assigned to a subcategory. The hierarchical methods can be either agglomerative or divisive. Agglomerative clustering places each object in its own cluster and gradually merges these atomic clusters into larger clusters until all objects are in a single cluster. In contrast, divisive methods start with all objects in one cluster and subdivide into smaller clusters.

Clustering methods have been employed to identify users' interests [Crabtree et al., 1998] and to collect evidence about user behaviour in various agent modelling systems (DOPPELGÄNGER [Orwant, 1995]; Syskill and Webert [Pazzani et al., 1997]). Researchers have also investigated the potential use of clustering in information retrieval and information filtering systems [Maes 1994; Sheth and Maes, 1993].

Stepp and Michalski [Stepp and Michalski , 1986] note that clustering algorithms based on numeric taxonomy fail to take into account background knowledge and also fail to provide meaningful semantic explanations for the resulting categories. That is, they can only produce extensional definitions and cannot produce intentional definitions of the resulting categories.

Conceptual clustering [Michalski, 1980; Stepp and Michalski, 1986] addresses these problems. Conceptual clustering algorithms can be used with objects represented by symbolic descriptors and produce simple concept descriptions by applying background knowledge in the formation of categories [Jain and Dubes, 1988].

The objective of conceptual clustering is to group objects into conceptually similar classes. Objects to be clustered are described by a number of attributes and the values of these attributes are textual descriptions.

Examples of clustering algorithms are:

- Statistical Clustering: as the name suggests this form of clustering has its historic roots in the field of statistical analysis. This simple algorithm produces a class hierarchy based on numerical similarity within object descriptions. Each object is described by a number of attributes, the values of these attributes are numerical values.
- Conceptual Clustering: this algorithm developed by Michalski [Michalski, 1980] overcomes the restriction of statistical clustering algorithms, which are limited to numeric values and allows contextual information to be taken into account.
- Kohonen Net Clustering: the Kohonen net has much in common with connectionist learning and can be thought of as two layers of nodes. The input layer has a node for each input, but there are a larger number of output nodes. Each input node is fully connected to every output node. There is a weight associated with each connection from the input layer to the output layer. The positions of the output nodes are determined by the values of the weights.

A 6.2.3 Inductive Learning of Classifiers

The result of applying the learning algorithm to a set of training examples is a classifier.

[Dumais, Platt, Heckerman and Sahami, 1998]

Classifiers can be learned by using inductive learning methods, such as Find Similar,Decision Trees, Naïve Bayes, Bayes Nets and Support Vector Machines.

These methods require only a small amount of labeled training data (i.e., examples of items in each category) as input. This training data is used to "learn" parameters of the classification model. Learned

classifiers are easy to construct and update. They require only subject knowledge and not programming or rule-writing skills. Inductively learned classifiers make it easy for users to customize category definitions, which is important for some applications. In addition, these learning methods provide graded estimates of category membership allowing for tradeoffs between precision and recall, depending on the task.

Find Similar: Our Find Similar method is a variant of Rocchio's method for relevance feedback [Rocchio, 1971] which is a popular method for expanding user queries on the basis of relevance judgements. Test instances are classified by comparing them to the category centroids using the Jaccard similarity measure. If the score exceeds a threshold, the item is classified as belonging to the category.

Decision Trees: A Decision Tree can be constructed for each category using the approach described by Chickering et al. [1997], being grown by recursive greedy splitting

Naïve Bayes: A Naïve-Bayes classifier can be constructed by using the training data to estimate the probability of each category given the values of a new instance, using the Bayes theorem to estimate the probabilities. The assumption of conditional independence is generally not true in real problems, nevertheless the Naïve Bayes classifier is surprisingly effective.

Bayes Nets: Bayesian nets are the consequence of the recently interest in learning more expressive Bayesian networks [Heckerman et al., 1995] as well as methods for learning networks specifically for classification [Sahami, 1996]. Sahami, for example, allows for a limited form of dependence between feature variables, thus relaxing the very restrictive assumptions of the Naïve Bayes classifier.

Support Vector Machines: Vapnik proposed Support Vector Machines (SVM) in 1979 [Vapnik, 1995], but they have only recently been gaining popularity in the learning community. In its simplest linear form, an SVM is a hyperplane that separates a set of positive examples from a set of negative examples with maximum margin.

A 6.2.4 Ensembles of classifiers

When a wide variety of data exists, single classifiers would fail in some regions of the training data or would be accurate in others (comptence region). Classifiers with different competence regions can be combined in order to enhance performance. This combination is usually called ensemble of classifiers [Boticario et al., 2001; Gaudioso and Boticario, 2002b].

An ensemble consists of a set of individually trained classifiers whose predictions are combined when classifying novel instances. Many researchers have investigated the technique of combining the predictions of multiple classifiers to produce a single classifier [Breiman, 1996; Clemen, 1989; Perrone, 1993; Wolpert, 1992]. The resulting classifier (the ensemble) is generally more accurate than any of the individual classifiers making up the ensemble. Both theoretical [Hansen and Salamon, 1990; Krogh and Vedelsby, 1995] and empirical [Hashem, 1997; Opitz and Shavlik, 1996a; Opitz and Shavlik, 1996b] research has demonstrated that a good ensemble is one where the individual classifiers in the ensemble are both accurate and make their errors on different parts of the input space.

Combining the output of several classifiers is useful only if there is disagreement among them [Opitz and Maclin, 1999]. As a result, methods for creating ensembles center around producing classifiers that disagree on their predictions. Generally, these methods focus on altering the training process in the hope that the resulting classifiers will produce different predictions. Two popular methods (Bagging [Breiman, 1996] and Boosting [Freund and Schapire, 1996; Schapire, 1990]) try to generate disagreement among the classifiers by altering the training set each classifier sees. These methods rely on "resampling" techniques to obtain different training sets for each of the classifiers.

Constructing esembles of classifiers

The task of constructing ensembles of classifiers [Dietterich, 1997] can be broken down into two sub-tasks, first to generate a diverse set of base-level classifiers and, once the base-level classifiers have been generated, the issue of how to combine their predictions arises [Zenko, Todorovski, and Dezeroski, 2001].

Several approaches to generating base-level classifiers are possible. One approach is to generate classifiers by applying different learning algorithms (with heterogeneous model representations) to a single data set (see, e.g., [Merz, 1999]). Another possibility is to apply a single learning algorithm with different parameters settings to a single data set. Finally, methods like bagging [Breiman, 1996] and boosting [Freund and Schapire, 1996] generate multiple classifiers by applying a single learning algorithm to different versions of a given data set.

Bagging and boosting apply the same learning algorithm to different versions of a given data set in order to obtain a set of diverse base-level classifiers. Bagging uses random sampling with replacement in order to

obtain different versions of a given data set. The size of each sampled data set equals the size of the original data set. On each of these versions of the data set the same learning algorithm is applied. Classifiers obtained in this manner are then combined with majority voting.

Boosting fist builds a classifier with some learning algorithm from the original data set. The weights of the misclassfied examples are then increased and another classifier is built using the same learning algorithm. The procedure is repeated several times. Classifiers derived in this manner are then combined using weighted voting.

Techniques for combining the predictions obtained from the multiple base-level classifiers can be clustered in three combining frameworks: voting (used in bagging and boosting), stacked generalization or stacking [Wolpert, 1992] and cascading [Gama, 1998].

In voting, each base-level classifier gives a vote for its prediction. The prediction receiving the most votes is the final prediction.

In stacking, a learning algorithm is used to learn how to combine the predictions of the base-level classifiers. The induced meta-level classifier is then used to obtain the final prediction from the predictions of the base-level classifiers.

Cascading is an iterative process of combining classifiers: at each iteration, the training data set is extended with the predictions obtained in the previous iteration. The basic idea of this method is to use the learning algorithms in sequence. At each iteration a two step process occurs. In the first step, a model is built using a base classifier. In the second step, the instance space is extended by the insertion of new attributes, generated by the base model. The constructive step generates terms in the representational language of the base classifier. If the high level classifier chooses one of these terms, its representational power has been extended. The *bias* restrictions of the high level classifier is relaxed by incorporating terms of the representational language of the base classifiers. This is the basic idea behind Ltree, a multivariate decision tree.

Dynamic integration of classifiers

[Puuronen, Terziyan and Tsymbal, 1999]

The integration of multiple classifiers has been under active research in machine learning and neural networks, and different approaches have been considered for example in [Chan and Stolfo, 1997; Dietterich, 1997; Kohavi, 1995; Koppel and Engelson, 1996; Merz, 1996; Merz, 1998; Ortega, Koppel and Argamon-Engelson, 1998; Puuronen, Terziyan, Katasonov and Tsymbal, 1999; Schapire, 1997; Skalak, 1997; Terziyan, Tsymbal and Puuronen, 1998; Terziyan, Tsymbal, Tkachuk and Puuronen, 1996; Tsymbal, Puuronen and Terziyan, 1998; Wolpert, 1992]. The challenge of integration is to decide which classifier to rely on or how to combine classifications produced by several classifiers.

Two main approaches have lately been used: selection of the best classifier and combining the classifications produced by the basic classifiers (these ones previosuly mentioned). One of the most popular and simplest selection approaches is CVM (Cross-Validation Majority) [Merz, 1996], which estimates the accuracy of each basic classifier using cross-validation and selects a classifier with the highest accuracy. More sophisticated selection approaches use estimates of the local accuracy of the basic classifiers by considering errors made in similar instances [Merz, 1996] or the meta-level classifiers (“referees”), which predict the correctness of the basic classifiers for a new instance [Ortega, Koppel and Argamon-Engelson, 1998]. Classifier selection methods can also be divided into two subsets: static and dynamic methods. A static method propose one “best” method for the whole data space (as for example CVM), while a dynamic method takes into account characteristics of a new instance to be classified (as for example the more sophisticated selection methods above).

The most popular and simplest method of combining classifiers is voting (also called majority voting and Select All Majority, SAM) [Merz, 1996]. The classifications produced by the basic classifiers are handled as equally weighted votes for those particular classifications and the classification with most votes is selected as the final classification. More sophisticated classification algorithms that use combination of classifiers include the stacking (stacked generalization) architecture [Wolpert, 1992], SCANN method that is based on the correspondence analysis and the nearest neighbor procedure [Merz, 1998], combining minimal nearest neighbor classifiers within the stacked generalization framework [Skalak, 1997], different versions of resampling (boosting, bagging, and crossvalidated resampling) that use one learning algorithm to train different classifiers on subsamples of the training set and then simple voting to combine those classifiers [Dietterich, 1997; Schapire, 1997]. Two effective classifiers’ combining strategies based on stacked generalization (called an arbiter and combiner) were analyzed in [Chan, 1997]. Nevertheless, there are still many open questions even with the widely used stacked generalization architecture, as which basic

classifiers should be used, what attributes should be used at the meta-level training, and what combining classifier should be used. Different combining algorithms have been considered by various researchers, as classic boosting with simple voting [Schapire, 1997] ID3 for combining nearest neighbor classifiers [Skalak, 1997], and the nearest neighbor classification in the space of correspondence analysis results (not directly on the predictions) [Merz, 1998].

Another ensemble of classifier is MAI (Model Applicability Induction) approach [Ortega, 1994; Ortega, 1996].

A 6.2.5 Machine Learning Software in Java

The Weka Machine Learning Project has built a software workbench called WEKA (Waikato Environment for Knowledge Analysis) that incorporate several standard machine learning techniques.

Weka is a collection of machine learning algorithms for solving real-world data mining problems. The Java-based Weka system (Weka 3) is written in Java and runs on almost any platform. The algorithms can either be applied directly to a dataset or called from any Java code. Weka is also well-suited for developing new machine learning schemes. Weka is open source software issued under the GNU General Public License. Apart from the following learning schemes, Weka also contains a large variety of tools that can be used for pre-processing datasets.

Implemented schemes for classification include:

- decision tree inducers
- rule learners
- naive Bayes
- decision tables
- locally weighted regression
- support vector machines
- instance-based learners
- logistic regression
- voted perceptrons
- multi-layer perceptron

Implemented schemes for numeric prediction include:

- linear regression
- model tree generators
- locally weighted regression
- instance-based learners
- decision tables
- multi-layer perceptron

Implemented "meta-schemes" include:

- bagging
- stacking
- boosting
- regression via classification
- classification via regression
- cost sensitive classification

Also included are clustering methods (Cobweb and an EM algorithm), and an association rule learner (Apriori).

A 6.2.6 Machine Learning Tasks already implemented in the aLF platform

To achieve adaptation, interaction data have to be accesible. User interaction items can be stored separately in database tables, as it is done in the aLF platform (see Appendix 2). However, there are some features of the user model that are not directly observable from the data in the database, but they can be inferred by the system [Kobsa et al, 1999]. This inference can be done by means of both several predefined rules and some machine learning techniques. The main disadvantage of the rule based approach is that the rules have to be predefined so all the process is too static. On the contrary, the machine learning techniques process training input and infer the required values based on this input. However, an hybrid approach combining these two different techniques is desired [Boticario et al., 2000; Pohl and Nick, 1999].

Focusing on machine learning, there are some learning tasks that can be used by the system to acquire the value of some attributes to perform tasks that promote collaboration among users [Preguia et al., 2000], for instance. The following tasks [Gaudioso y Boticario, 2002a] have already been implemented in the aLF platform, and each of them uses different machine learning techniques.

Prediction of the levels of activity

The objective of these learning tasks is to predict each user's level of activity in general and in specific individual services. For any given learning task, an algorithm processes an input consisting of a set of training examples (selected from the examples collected) and keeps the results in their specific form. Then the system uses the values learnt in the adaptation task for the training examples under consideration.

Usually, the learning tasks posed are classification tasks, where the objective of the task is to assign a specific class value to a specific example. Like any learning task [Webb et al., 2001], it is necessary to determine what the input attributes are going to be for each instance in the training examples. By extracting and preparing data from the database, a set of training instances as input data to the learning algorithm can be obtained. In this case, each instance corresponds to a user and includes information about his/her behaviour in his/her interactions, and from this process a dataset of instances labeled with a class value is obtained, where the training data was constructed taking into account data labeled manually.

However, when a wide variety of interaction data exists, like in a web environment, single classifiers would fail in some regions of the training data but would be accurate in others (their competence region). Single classifiers with different competence regions can combine to enhance performance, using ensemble of classifiers [Boticario et al., 2001; Gaudioso and Boticario, 2002b].

A decision tree learner (C4.5), a Bayesian learner (Naive Bayes) and two ensembles (MAI [Ortega, 1995] and Ltree [Gama and Brazdil, 1999]) have been used.

Automatic allocation of categories on forum messages

The automatic classification of texts [Dumais et al., 1998] is an important component in any automatic knowledge management task. To date it has usually been applied in the automatic classification of documents for supporting extraction and information filtering tasks, as well as in many other personalisation tasks for information management: electronic mail message classification, classification of files in folders, etc.

To perform automatic classification of texts, it is necessary to predefine some vocabulary containing all the possible categories, and these categories are used to classify the input documents. In this case, the learning tasks posed consist of learning the existing relationship between a specific document and its corresponding category. This task can be resolved by learning a classifier for each defined category. More specifically, for each category and classifier to be learnt, the documents belonging to a specific category are taken as positive examples and the other documents are taken as negative examples.

There are text classification systems that incorporate the data preprocessing for facilitating the final learning mechanisms such as the Rainbow system [McCallum, 1996], which has been used in aLF for automatic allocation of categories on forum messages. Although aLF forums provide categories for organising the forum messages, users very often do not assign these categories to their messages. The direct consequence of this situation is that there are more and more sets of unclassified messages (in this instance the messages would belong to the category no category). The purpose of this learning task is therefore to provide certain categories (from those existing) for unclassified messages.

Automatic classification text systems have used numerous automatic learning and classification techniques, such as Find Similar, Naive Bayes, Bayesian networks, decision trees and support vector machines.

Automatic grouping of users in collaboration subgroups

The objective of this task is to group users into subgroups to facilitate collaboration among them with the course tools. There are two alternatives which do not require manually processing [Gaudioso and Boticario, 2002a]: 1) Using a supervised learning task, taking as input a series of students previously grouped in subgroups to learn a classifier that is assigned to each new student in a specific subgroup; and 2) Automatically grouping students just taking into account their interaction profile by constructing heterogeneous workgroups (as regards knowledge) where the group members' different levels of knowledge will lead to greater collaboration.

To perform this task in aLF platform, the second alternative has been implemented. This task has been posed as a non-supervised learning task, where there are no labeled training examples. Examples of unsupervised learning user modelling can be found in the system Doppelgänger [Orwant, 1991].

A 6.3 Assessment

A 6.3.1 Conclusions for ALFANET

To achieve adaptation in the system to be developed it seems necessary to have dynamic user models that adapt to the user's interaction with the system [Garrido and Gea, 2001; Hoppe, 1995]. These adapted models allow the system to provide the user with an adapted response to these models. Machine learning techniques can be used for user modeling, and therefore, can be used in the development of aLFanet.

Nevertheless, machine learning field is very rich in research and lots of theory exists. Time and effort will be needed to spent in order to select the best techinique to be used for each particular case.

The system to be developed will be used by wide and heterogenous users, whose needs will be completely different, and it is impossible to know beforehand which is the appropiate task in each case. The solution is using machine learning techniques to combine different strategies.

Some learning tasks have already been implemented in the aLF platform, and can serve as the basis for others to be performed by the system to be developed in aLFanet Project.

A 6.4 References

BOTICARIO, J. G., GAUDIOSO, E., and HERNANDEZ, F. 2001. A machine learning architecture for user model acquisition. In *Proceedings of the AIED 01. Frontiers in Articial Intelligence and Applications*. IOS Pres, San Antonio, USA.

BOTICARIO, J. G., GAUDIOSO, E., and CATALINA, C. 2001. Towards personalised learning communities on the web. In *European Perspectives on Computer-Supported Collaborative Learning. Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, K. H. PIERRE DILLENBOURG, A. EURELINGS, Ed. Maastricht McLuhan Institute, 115-122.

BOTICARIO, J. G., GAUDIOSO, E., and HERNANDEZ, F. 2000. Adaptive navigation support and adaptive collaboration support in webdl. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. Number 1892 in Lecture Notes in Computer Science (LNCS). Springer Verlag, Trento, Italy, 51-61.

Breiman, L. (1996) Bagging Predictors. Machine Learning 24(2): 123-140.

Chan, P., Stolfo, S.: On the Accuracy of Meta-Learning for Scalable Data Mining. Intelligent Information Systems, Vol. 8 (1997) 5-28.

Chickering D., Heckerman D., and Meek, C. A Bayesian approach for learning Bayesian networks with local structure. In *Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

Clemen, R. 1989. Combining forecasts: A review and annotated bibliography. Journal of Forecasting, 5, 559-583.

Crabtree B. I., Soltysiak S. J. (1998) 'Identifying and tracking changing interests' International Journal on Digital Libraries 1998.

Dietterich, T. G. (1997) Machine-Learning Research: Four Current Directions. AI Magazine 18(4): 97-136.

DUMAIS, S., J, P., HECKERMAN, D., and SAHAMI, M. 1998. Inductive learning algorithms and representations for text categorization. *In Proceedings of the 1998 ACM CIKM International Conference on*

Information and Knowledge Management. G. GARDARIN, J. FRENCH, N. PISSINOU, K. MAKKI and L. BOUGANIM, Eds. ACM Press, Bethesda, MD. 148-155.

Freund, Y. and Schapire, R. E. (1996) Experiments with a New Boosting Algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, pages 148-156. Morgan Kaufmann.

Gama, J. (1998) Combining Classifiers by Constructive Induction. In Proceedings of the Ninth European Conference on Machine Learning.

GAMA, J. and BRAZDIL, P. 1999. Linear tree. *Intelligent Data Analysis* 3, 3.

Gama, J. (2000) Combining Classification Algorithms. Universidade do Porto.

Gaudioso, E. y Boticario, J. (2002a) Collection, Analysing and Interpreting Data for Modelling Collaborative Tasks in an Adaptive On-line Learning Community.

GAUDIOSO, E. and BOTICARIO, J. 2002b. User data management and usage model acquisition in an adaptive educational collaborative environment. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-based* Systems. Number 2347 in Lecture Notes in Computer Science (LNCS). Springer Verlag, Malaga, Spain, 143-152.

GARRIDO, J. and GEA, M. 2001. Modelling dynamic group behaviours. In *8th International Workshop of Interactive Systems: Design, Specication and Verication* . Number 2220 in Lecture Notes in Computer Science (LNCS). Springer Verlag, Glasgow, UK. 128-143

Gilmore D. and Self J. (1988)The application of machine learning to intelligent tutoring systems. *Artificial Intelligence and Human Learning. Intelligent Computer-Aided Instructions,* Self J. (Eds).

Hansen, L. Salamon, P. 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12, 993-1001.

Hashem, S. 1997. Optimal linear combinations of neural networks. Neural Networks, 10(4), 599-614.

Heckerman, D. Geiger, D. and Chickering, D.M. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning, 20,* 131-163, 1995.

HOPPE, U. 1995. Use of multiple student modeling to parametrize group learning. In *Proceedings of 7th International Conference on Articial Intelligence in Education (AI-ED95)*. AACE, Washington, DC., 234-249.

Jain, A., and Dubes, R., "*Algorithms for Clustering Data*," published by Prentice-Hall, 1988.

KOBSA, A., KOWNEMANN, J., and POHL, W. 1999. Personalized hypermedia presentation techniques for improving online customer relationships. *Technical Report 66*, German National Research Center for Information Technology, St. Augustin, Germany.

Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of IJCAI'95 (1995).

Koppel, M., Engelson, S.P.: Integrating Multiple Classifiers by Finding their Areas of Expertise. In: AAAI-96 Workshop On Integrating Multiple Learning Models (1996) 53-58.

Krogh, A. Vedelsby, J. 1995. Neural network ensembles, cross validation, and active learning. In Tesauro, G., Touretzky, D., Leen, T., Advances in Neural Information Processing Systems, 7, 231-238 Cambridge, MA. MIT Press.

Langley, P. (1996) *Elements of Machine Learning*. Morgan Kaufmann Publishers, Inc.

Maes P. (1994) 'Agents that Reduce Work and Information Overload' *Communications of the ACM*, July 1994.

MCCALLUM, A. K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classication and clustering. http://www.cs.cmu.edu/ mccallum/bow.

Merz, C.: Dynamical Selection of Learning Algorithms. In: D.Fisher, H.-J.Lenz (Eds.), Learning from Data, Artificial Intelligence and Statistics, Springer Verlag, NY (1996).

Merz, C.J.: Combining Classifiers Using Correspondence Analysis. In: Advances in Neural Information Processing Systems 10, M.I.Jordan, M.J.Kearns, S.A.Solla, eds., MIT Press, 1998.

Merz, C. J. (1999) Using Correspondence Analysis to Combine Classifiers. Machine Learning 36(1/2): 33-58. Kluwer Academic Publishers.

Michalski R.S. (1980) Pattern recognition as rule-guided inductive inference. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-2, pp.349-361.

Mitchell T.M., Keller R.M. and Kedar-Cabelli S.T. (1986)Explanation-based generalization: A unifying view. Machine Learning, vol.1, pp. 47-80.

MInet. The Machine Learning network. http://kiew.cs.uni-dortmund.de:8001/ Training Information Server.

Opitz, D. and Maclin, R. (1999) Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research* 11 (1999), pp. 169-198.

Opitz, D. and Shavlik, J. 1996a. Actively searching for an effective neural-network ensemble Connection Science, 8(3/4), 337-353.

Opitz, D. and Shavlik, J. 1996b. Generating accurate and diverse members of a neural-network ensemble. In Touretsky, D., Mozer, M., Hasselmo, M., Advances in Neural Information Processing Systems, 8, 535-541 Cambridge, MA. MIT Press.

Ortega, J. (1994) Making the most of what you've got: using models and data to improve learning rate and prediction accuracy. Technical Report Computer Science Dept., Vanderbilt University. Abstract appears in Proceedings of the Twelfth National Conference on Artificial Intelligence, p. 1483, Seattle, Wa.

ORTEGA, J. 1995. Exploiting multiple existing models and learning algorithms. In *AAAI 95- Workshop in Induction of Multiple Learning Model.*

Ortega, J. (1996) Making the most of what you've got: using models and data to improve learning rate and prediction accuracy. Ph. D. Dissertation, Vanderbilt University, Nashville, TN.

Ortega, J., Koppel, M., Argamon-Engelson, S.: Arbitrating Among Competing Classifiers Using Learned Referees, Machine Learning (1998) to appear.

ORWANT, J. 1991. *Doppelgänger goes to school: Machine learning for user modeling*. M.S. Thesis, Program in Media Arts and Sciences, MIT, 1991. 121

Orwant J. (1995) 'Heterogeneous Learning in the Doppelgänger User Modeling System' *User Modeling and User-Adapted Interaction*, 4: 107-130.

Pazzani M., Muramatsu J. and Billsus D. (1997) 'Syskill & Webert: Identifying interesting web sites' http://128.195.1.46/~pazzani/RTF/AAAI.html.

Perrone, M. 1993. Improving Regression Estimation: Averaging Methods for Variance Reduction with Extension to General Convex Measure Optimization. Ph.D. thesis, Brown University, Providence, RI.

POHL, W. and NICK, A. 1999. Machine learning and knowledge representation in the labour approach to user modeling. In *Proceedings of the 7th International Conference on User Modeling*. Banff, Canada, 179--188.

PREGUIA, N., MARTINS, J. L., DOMINGOS, H. J., and DUARTE, S. 2000. Data management support for asynchronous groupware. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW'00)*. ACM DL, Philadelphia, PA, USA, 69-78.

Puuronen S., Terziyan V., Tsymbal A. (1999), A Dynamic Integration Algorithm for an Ensemble of Classifiers. In: Z.W. Ras, A. Skowron (Eds.), *Foundations of Intelligent Systems, 11th International Symposium ISMIS'99*, Warsaw, Poland, Lecture Notes in Artificial Intelligence, Vol. 1609, Springer-Verlag, pp. 592-600. (Copyright © Springer-Verlag)

Puuronen, S., Terziyan, V., Katasonov, A., Tsymbal, A.: Dynamic Integration of Multiple Data Mining Techniques in a Knowledge Discovery Management System. In: SPIE Conf. on Data Mining and Knowledge Discovery, 5-9 April 1999, Orlando. Florida.

Quinlan I.R. (1983)Learning efficient classification procedures and their application to chess end-games. *Machine Learning: An Artificial Intelligence Approach,* Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds), Morgan Kaufmann, pp. 463-482.

Quinlan I. R. (1993)*C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Rocchio, J.J. Jr. Relevance feedback in information retrieval. In G.Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing,* 313-323. Prentice Hall, 1971.

Sahami, M. Learning Limited Dependence Bayesian Classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 335-338, AAAI Press, 1996. http://robotics.stanford.edu/users/sahami/papers-dir/kdd96-learn-bn.ps

Schapire, R. 1990. The strength of weak learnability Machine Learning, 5(2), 197-227.

Schapire, R.E.: Using Output Codes to Boost Multiclass Learning Problems. In: Machine Learning: Proceedings of the Fourteenth International Conference (1997) 313-321.

Sheth B. and Maes P. (1993) 'Evolving Agents for Personalised Information Filtering' *IEEE Conference on Artificial Intelligence for Applications*.

Simon H. (1983)Why Should Machine Learn?. *Machine Learning: an Artificial Intelligence Approach*, Michalski R., Carbonell J. and Mitchell T.M., Palo Alto:Tioga.

Skalak, D.B.: Combining Nearest Neighbor Classifiers. Ph.D. Thesis, Dept. of Computer Science, University of Massachusetts, Amherst, MA (1997).

Smith, A. (1999) *Application of Machine Learning Algorithms in Adaptive Web-based Information Systems*: http://www.cs.mdx.ac.uk/staffpages/serengul/Pdf/thesis.htm

Stepp and Michalski (1986) 'Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects' in Michalski R.S., Carbonell J. G. and Mitchell T.M. (eds) *Machine Learning: An Artificial Intelligence Approach* Vol.2 Los Alto, CA: Morgan Kaufmann.

Terziyan, V., Tsymbal, A., Puuronen, S.: The Decision Support System for Telemedicine Based on Multiple Expertise. Int. J. of Medical Informatics, Vol. 49, No. 2 (1998) 217-229.

Terziyan, V., Tsymbal, A., Tkachuk, A., Puuronen, S.: Intelligent Medical Diagnostics System Based on Integration of Statistical Methods. In: Informatica Medica Slovenica, Journal of Slovenian Society of Medical Informatics,Vol.3, Ns. 1,2,3 (1996) 109-114.

Tsymbal, A., Puuronen, S., Terziyan, V.: Advanced Dynamic Selection of Diagnostic Methods. In: Proceedings 11th IEEE Symp. on Computer-Based Medical Systems CBMS'98, IEEE CS Press, Lubbock, Texas, June (1998) 50-54.

Utgoff, P. E. (1989)Incremental Induction of Decision Trees. *Machine Learning*, vol. 4, pp. 161-186.

Vapnik, V., *The Nature of Statistical Learning Theory,* Springer-Verlag, 1995.

WEBB, G. I., PAZZANI, M. J., and BILLSUS, D. 2001. Machine learning for user modeling. *User Modelling and User Adapted Interaction* 11, 1, 19-29.

Weka Machine Learning Project. http://www.cs.waikato.ac.nz/~ml/index.html The University of Waikato.

Wolpert, D. 1992. Stacked generalization. Neural Networks, 5, 241-259.

Young R.M., Plotkin G.D. and Linz R.F. (1977).Analysis of an extended concept learning task. *IJCAI-77 Proceedings*, Los Altos, CA: Morgan Kaufmann. Inc.

Zenko, B., Todorovski, L. and Dezeroski, S. (2001) A comparison of stacking with MDTs to bagging, boosting, and other stacking methods. Jozef Stefan Institute. Ljubljana, Slovenia. IDDM 2001.

Appendix 7 User modelling

A 7.1 Overview

Adaptation is essential in any e-learning environment because its users have a wide variety of backgrounds, interests, level of experience on the use of resources, etc. and they demand an environment that adapts to each individual user needs. To carry out this adaptation a user model representing the user's knowledge state, preferences and goals has to be constructed.

The adaptation is done through user model acquisition from the user data available and from traces of interaction of that user with the system, combining the manual and the automatic techniques used in Intelligent Tutoring systems, Adaptive Hypermedia Programs and Learning Apprentice Systems. As it will be exposed in this appendix, different kinds of data have to be considered in order to build the user model: data about the user, data about the computer usage and data about the user's hardware, software and physical environment. Different techniques to acquire these data have to be employed, depending on the nature of each data.

Two approaches can be used in user modelling representation and inference, one based on knowledge representation that uses a representation formalism and can perform reasoning tasks and the other one based on machine learning techniques.

User modelling is a huge field in research, and there are many kinds of models, many input data and many ways of representation, but the main points are the data themselves, the way to acquire them and their representation and inference.

Another issue to be taken into account is whether the system will be run in a web-based environment or not. In a web-based environment, there is less control about the interaction of the user, and therefore it is more difficult to obtain the data to construct the user model.

A 7.1.1 User Models

A user model is an explicit representation of the properties of an individual user and can be used to reason about the needs, preferences or future behavior of that user. Most computer systems that interact with humans contain some kind of implicit model of the user, but there are some difficulties when incorporating a user model into the design of existing systems, namely development resources and performance.

There can be a wide variety of user model types, and models can be classified along these four dimensions [Ross, 2000]:

- **What is modeled:** Canonical user or Individual user
- **Source of modelling information:** Model constructed explicitly by the user or Model abstracted by the system on the basis of the user's behavior
- **The time sensitivity of the model:** Short-term (highly specific information) or Longer-term (more general information)
- **Update methods:** Static model or Dynamic model

The update methods often follow from the other three dimensions; Individual User models, Models abstracted on the basis of user behavior and Short-term models generally require dynamic update.

If the model contains very short-term information then it can become a task model, since it is relevant to the task in hand, and the individual user is not important. This is because the model will update immediately to reflect any task which a new user undertakes.

The most basic type of model is static and contains a canonical user. This type of model can be embedded within a system and almost does not need to be stored explicitly. In contrast, if the individual user is modeled, then dynamic update is required, and explicit methods are necessary to describe how the user model state affects the system performance.

The forms that a user model may take are as varied as the purposes for which user models are formed. User models may seek to describe [Webb, Pazzani and Billsus, 2001]:

- the cognitive processes that underlie the user's actions

- the di¡erences between the user's skills and expert skills
- the user's behavioral patterns or preferences
- the user's characteristics

Another important dimension along which it is important to distinguish approaches is with respect to whether they model individual users or communities of users.

A 7.1.2 Evaluation of User Models

[Chin, 2001]

User models cannot and should not be separated from the software systems that use them. After all, what good is a user model if it will not be used for anything? A system with an unused user model might as well not have a user model at all. If the user model is indeed being used to make a difference in a software system such as adapting the software system to the user, then one should ask whether the user model adaptations actually improve the software system. Also, what types of users benefit from the adaptations? It may very well be that some user model adaptations are less beneficial or even detrimental to some classes of users.

In general, adding a user model to any software system will most likely make it more complex, less predictable, and more buggy. Consequently, it is a very reasonable question to ask whether or not the user model will actually improve the system. Even when a user model adapts a system to follow the users explicit wishes exactly, there is still a question as to whether this is a good idea. For example, the users preferred configuration may actually be slower or more error-prone than an ideal configuration. Or, the surplus of different configurations may make it difficult for users in a group to cooperate, thus decreasing overall efficiency, even though efficiency for individual users working alone may be improved. A common adaptation for user models is information filtering, which may seem to be always helpful, especially in todays information-overloaded society. However, eliminating seemingly irrelevant information can confuse users, thus decreasing performance. For example, eliminating irrelevant streets from maps may make them more readable unless the system has eliminated some of the user's landmarks (e.g., the user may look for a particular unique crosshatch pattern of streets to quickly locate a particular neighborhood). Likewise, eliminating irrelevant links from a web page may confuse the user's navigation when one of the irrelevant links is part of a path that the user relies on. There may be a more direct path elsewhere in the page, but that does not help because the user does not know it. So, we must test the usefulness of user model adaptations through experiments before we can claim that they are helpful.

Student modelling has had a long history of empirical evaluation stemming from the educational psychology roots of intelligent tutoring and computer-aided instruction systems. Student modelling systems are typically evaluated by comparing systems with and without student models. As a preliminary step, the accuracy of student models can also be tested. For example, one can compare predicted student actions/results with actual actions/results or compute the percentage of recognized bugs. Systems that use machine learning methods to acquire user models in any area can evaluate the acquired user models using standard machine learning measures that compare the user model against a reserved set of test data that was not used for training (typically an 80/20% split for training/testing).

A 7.1.3 Techniques for User Modelling

The personalization process can be divided into three major tasks, Acquisition, Representation and Secondary Inference, and Production [Kobsa, Koenemann and Pohl, 2001]. Since we are focusing on User Modelling, we will only describe the first two in this section. However, a brief description of all of them is given next:

- **Acquisition:** identifies the information that is available about users' characteristics and computer usage behavior as well as the usage environment, either by monitoring the computer usage or by obtaining this information from external sources. This information has to be made accessible to the adaptation component of the application to construct initial models of the user, the computer usage and/or the usage environment.
- **Representation and Secondary Inference:** expresses the content of the user and usage models appropriately in a formal system, to allow access and futher processing and to draw further 'secondary' assumptions about users and/or user groups, their behavior and their environment, thereby integrating information from various sources.

- **Production:** generates the adaptation of content, presentation and modality, and structure based on a given user, usage and environment model.

A 7.1.3.1 Input Data for User Modelling

Personalization is often a data-intensive task. Some data can typically be observed by the system directly, while most others may require one or more additional acquisition steps. This section deals with the different kinds of data that user-adaptive systems may need to consider when adapting the system to the current user individual needs, namely data about the user, data about computer usage and data about the user's hardware, software and physical environment. These data can be behavior-related or related mental notions.

The following table sumarizes the different kinds of data that will be described next [Kobsa, Konnemann and Pohl, 2001].

USER MODEL			
User Data (mentalistic)	**Usage Data (behavior-related)**		**Environment Data**
	Observable Usage	**Usage Regularities**	
Demographic	Selective actions	Usage frequency	Software
Knowledge	Temporal viewing behavior	Situation-action correlations	Hardware
Skills & Capabilities	Ratings	Action sequences	Locale
Technological experience	Other confirmatory actions		
Individual traits			
Interests & Preferences			
Goals & Plans			

User Data

User data denote information about personal characteristics of the user, while usage data is related to a user's (interactive) behavior. However, there are overlaps between these two categories, because although some user data can be directly supplied by the user, most data must normally be inferred from usage observations. Several categories of user data have been found:

- Demographic data about the user are objective facts, like record data (name, address, phone number), geographic data (area code, city, state, country), user characteristics (age, sex, education, disposable income), psychological data (data indicating lifestyle).
- Data relating to user knowledge (*knowing what*), which implies assumptions on user's knowledge about concepts, relationship between concepts, facts and rules with regard to the domain of the application system (Adaptive Hypermedia Systems such as **Sales Assistant** [Popp and Lödel, 1996], **MetaDoc** [Boyle and Encarnacion, 1994], **KN-AHS** [Kobsa, Müller and Nill, 1994], **SETA** [Ardissono and Goy, 1999; Ardissono, Goy, Meo and Petrone, 1999; Ardissono and Goy, 2000] and Intelligent Tutoring Systems such as **ISIS-Tutor** [Brusilovsky and Pesin, 1994], **Hypadapter** [Hohl, Böcker and Gunzenhäuser, 1996], **ELM-ART II** [Weber and Specht, 1997]).
- Data relating to user skills and capabilities (*knowing how*), which can be distinguished between the actions a user is familiar with and the actions he or she is actually able to perform (Adaptive Help Systems such as **Unix Consultant** [Chin, 1989] and **AVANTI** [Fink, Kobsa and Nill, 1998]).
- Data relating to individual traits, which means features that together define a user as an individual: personality factors, cognitive factors and learning styles, and have to be extracted by specially designed psychological tests.
- Data relating user interests and preferences, which is the central notion for so-called recommender systems.
- Data relating user goals and plans to support users in achieving their goals by plan recognition. (**PUSH** [Höök, Karlgren, Waern, Dahlbäck, Jansson, Karlgren and Lemaire, 1996], **HYPERFLEX** [Kaplan, Fenwick and Chen, 1993])

Usage Data

Usage data is related to a user's (interactive) behavior and to the usage context, and may be directly observed and recorded, or acquired by analyzing observable data. The extent to which observation is technically possible varies considerably. Hypermedia systems that are exclusively based on HTML will only be able to record what pages and files have been requested from the server (which includes a trace of the navigation paths for links on the same site). Systems that have more control about the interaction by, e.g., using Java applets, can record usage data on the level of mouse clicks and movements (**ELFI** [Pohl and Nick, 1999] and **TELLIM** [Joerding, Michel and Popella, 1998]).

In addition to interaction behavior, the usage context may also be considered as a source for adaptation. Among the relevant items are the current task and the interaction history. Typically, merely application-specific context is currently taken into account, although it might be desirable to also consider the user's general situation, e.g. other applications that the user currently or frequently employs.

Usage data provide useful information for building the user model since deals with the interactions of the user and that is the key point to adapt the system to the user as much as posible. However, these data are the most difficult to obtain information out of them and usually a further processing is needed before using the usage data for the user model, although some data can be directly observed and may lead directly to adaptation. In a web-based environment it is even more difficult to get usage information since the user is not in a closed environment.

- Observable usage can get data through the different ways in which users may interact with the system.
    - Selective actions, which can be regarded as indicators for interest (**WebWatcher** [Joachims, Freitag and Mitchell, 1997], **Letizia** [Lieberman, 1995], **HIPS** [Oppermann and Specht, 1999, 2000]), unfamiliarity with technical terms (**MetaDoc** [Boyle and Encarnacion, 1994], **KN-AHS** [Kobsa, Müller and Nill, 1994], **Sales Assistant** [Popp and Lödel, 1996]) and preferences (**Adaptive Graphics Analyser** [Holynski, 1988]). These actions need not only be mouse clicks, and may only be weak indicators and lead to positive evidence only.
    - Temporal viewing behavior, although in most cases is a weak indicator because it is impossible to tell whether the user has been looking at a specific item within a specific time interval.
    - Ratings are used to indicate how relevant or interesting an object is to the user, or how relevant or interesting the user thinks it is to other users (**Firefly** [Shardanand and Maes, 1995], **Syskill & Webert** [Pazzani and Billsus, 1997], **GroupLens** [Konstan, Miller, Maltz, Herlocker, Gordon and Riedl, 1997]). However, the relevance of information is always relative to the information need of a user, is not independent and explicit user input is not very reliable, particulary on negative ratings.
    - Other confirmatiory and disconfirmatory actions, such as further processing of an object (e.g. saving a document).
- Usage regularities are obtained by further processing of usage data. Examples of usage information that is acquired from observed usage data are:
    - Usage frequency, done by categorizing events and counting their frequencies (**Word** [Debevc, Meyer, Donlagic and Svecko, 1996], **Flexcel** [Thomas and Krogsaeter, 1993; Krogsaeter, Oppermann and Thomas, 1994], **AVANTI** [Fink, Kobsa and Nill, 1998]).
    - Situation-action correlations, which express generalized correlations between previous situations and how the user processed them.
    - Action sequences, which are analyzed to recommend macros for frequently-used action sequences, predict future user actions on the basis of past actions and recommend actions based on frequent action sequence of other users.

For instance, in our domain the usage data may include all those attributes that allows to determine certain characteristics in the user interaction with the system, such as whether a particular resource has been used or not (has_bookmarks?), counters for the use or access to each service or resource (files_personal_storage_area, num_msgs_forum), boolean indicators to know whether a service is activated or not (email_alerts?), time spent using a resource (average_session_duration), ... and some others to determine the usefullness of the contribution of the user to each of the resources (fellows_replayed_user_message).

Environment Data

In a web-based environment these data have also to be taken into account. Web-usage may be influence by the software and the hardware of the individual user since there are many web-capable appliances with limited abilities, and by the characteristics of the user's current locale.

- The software environment deals with the browser version and platform, the availability of plug-ins for media contents and the Java and Javascript compliance.
- The key points in the hardware environment are bandwidth, processing speed, display devices and input devices.
- Information about usage locale includes users' current location and characteristics of usage locale.

A 7.1.3.2 Acquisition Methods

Once reviewed the types of user, usage and environment data that are potential input to user models, we will described methods that can be used to obtain these data [Kobsa, Koenemann and Pohl, 2001].

User Data Acquisition Methods

Acquisition methods are employed for acquiring explicit assumptions about user data, and to store these data in an initial user model. These methods can be based on explicit user input and active acquisition behavior of the system, or on passive acquisition, namely domain-dependent acquisition rules, plan and goal recognition and stereotypes for user classification.

- **Active acquisition**, by letting the user supply the necessary data through questions posed by the system in the initial phase of system usage, through interviews and through reports about their usual activities (**AVANTI** [Fink, Kobsa and Nill, 1998]**, SATELIT** [Akoulchina and Ganascia, 1997], **ELM-ART II** [Weber and Specht, 1997]). Users do not like to spend time on set-up to optimize the system, so the acquisition phase should therefore be minimized or automatized by means of a central user model.
- **Passive acquisition** is less disturbing and annoying to the user than active acquisition since it does not initiate any interaction with the user. Techniques for passive acquisition are:
    - **Acquisition rules**, which are inference rules that are typically executed when new information about the user is available, and usually refer to observed user actions or straightforward interpretation of user behavior. Some domain-independent acquisition systems are **KNOME** [Chin, 1989] and **GUMAC** [Kass, 1991].
    - **Plan recognition** deals with reasoning about the goals that the user may pursue and the action sequence he/she performs to achieve them, and consists of a task knowledge base that models possible user actions and relationships between these actions, and of mechanisms that identifies the current plan and the associated goals of the user from the observed interactions.
    - **Stereotype reasoning** is the simplest method and consists of classifying users into categories based on certain basic attributes of the users, and then making predictions about them based on the stereotype that is associated to each category. A set of activation conditions are needed for applying the stereotype to a user.

However, none of these techniques has been chosen as the best one so far, but the choice depends on the domain where the technique is to be applied. In any case, some usability criteria have to be taken into account, such as bothering the user as less as possible.

Usage Data Acquisition Methods

These methods construct aggregated information about a user's interactive behavior from individual observations and from learning correlations between situations that the user encounters and the actions he/she performes to predict user behavior in future situations, to suggest appropiate actions to the user, to automatically perfom actions on the user's behalf, to predict action sequences or to reccomend actions based on action sequences or other users (**Flexcel** [Thomas and Krogsaeter, 1993; Krogsaeter, Oppermann and Thomas, 1994], **Basar** [Thomas and Fischer, 1996], **Doppelgänger** [Orwant, 1995]).

Although already mentioned before, obtaining these data require a lot of work and effort, but they are also the richer data for adaptation.

Environment Data Acquisition Methods

The key problem for methods for acquiring information about user's environment is the mapping of a physical device to an individual person in a multi-user context. In this case, environment data must be represented separated from user data.

For the software environment, browser constrains are taken into account. However, hardware constrains are often difficut to asses, and can, at the most, be guessed (**Hyperspace Agent** [Fuller and de Graaff, 1996], **AVANTI** [Fink, Kobsa and Nill, 1998], **TELLIM** [Joerding, Michel and Popella, 1998], **AWCD** [Chen, Yang and Zhang, 2000]). Locality information can be acquired by a wide range of methods, both for stationary network devices and mobile ones, and a combination of methods has to be used for best results.

A 7.1.3.3 Representation and Secondary Inferences

When user and usage models have been acquired, they need to be represented in order to be available for further exploitation. Systems also need to employ inferences to further augment the user and usage model based on initial acquisition results, domain knowledge and knowledge about other users, but do not consider the current input anymore (and that is why they have been called secondary inferences). Three kinds of reasoning techniques can be distinguished, Deductive, Inductive and Analogical reasoning.

There exist two main approaches for user modelling representation and inference, the one based on knowledege representation (predefined rules) and the one based on machine learning techniques (learned rules from the observed data). The first one makes uses of deductive reasoning, while the second one uses both inductive and analogical reasoning.

The most common representation approaches and the inference techniques associated with them are described next. These methods are used for representing the acquired data as models in formal systems and for making generalizations and predictions about the user.

Three kinds of reasoning can be distinguished [Kobsa, Kownemann and Pohl, 2001]:

- **Deductive reasoning** (from the more general to the more specific) can be obtained by:
    - Logic-based representation and inference, using concept formalisms, propositional calculus or modal logics (**KN-AHS** [Kobsa, Müller and Nill, 1994]). The shortcomings of this approach to user model representation is its limited ability to deal with uncertainty and with changes to user model, unless non-monotonic logics are used to deal with changes and revisions.
    - Representation and reasoning with uncertainty is used in systems like **HYDRIVE** [Mislevy and Gitomer, 1996], **EPIAIM** [De Rosis, Pizzutilo, Russo, Berry and Molina, 1992], **PRACMA** [Jameson, Schäfer, Simons and Weis, 1995] and **Microsoft Office Assitant** [Horvitz, 1997; Horvitz, Breese, Heckerman, Hovel and Rommelse, 1998]. Evidence-based techniques used are linear parameters together with feature-value pairs and fuzzy logic (**Sales Assistant** [Popp and Lödel, 1996]).
- **Inductive reasoning** (from specific cases to the general case) involves monitoring user's interaction with the application and drawing general conclusions based on a series of observations, and are usually used for constructing users' explicit or implicit interest profiles that will be used for feature-based filtering (**Syskill & Webert** [Pazzani and Billsus, 1997], **Fab** [Balabanovic, 1997; Balabanovic and Shoham, 1997], **Letizia** [Lieberman, 1995]). Several machine learning algorithms can be applied.
- **Analogical reasoning** (from similar cases to the present case) exploits the fact that web-based systems have a large number of users and tries to recognize similarities between them. Two approaches can be used:
    - **Clique-based filtering**, that for a given user try to find other users who show similar interaction behavior, instead of analyzing the features of the objects in which the user has expressed an interest like in inductive learning, which is not easy (**GroupLens** [Konstan, Miller, Maltz, Herlocker, Gordon and Riedl, 1997]). To get the recommendations, similar neighbors have to be found, the set of closest users has to be selected and the prediction based on weighted representation of selected neighbors has to be computed.
    - **Cluster User Profiles** allows the system to form explicit user profiles using machine learning techniques. If profiles of different users are stored, clustering algoritms are applied to find similar users and to form group profiles, so individual profiles can be compared to these profiles. In this way, reclassification can be supported (**Doppelgänger** [Orwant, 1995]).

A 7.2 Solutions

In early user-adaptive systems, the use of knowledge representation methods for user modelling has often been the focus of research. However, the application of machine learning techniques to control user-adapted interaction has become popular in recent years. Both approaches have significant drawbacks, and therefore, an hybrid approach may be a better choice. In this section, we will give an overview of the three of them [Pohl and Nick, 1999; Pohl, Schwab and Koychev, 1999].

A 7.2.1 Knowledge Representation for User Modelling

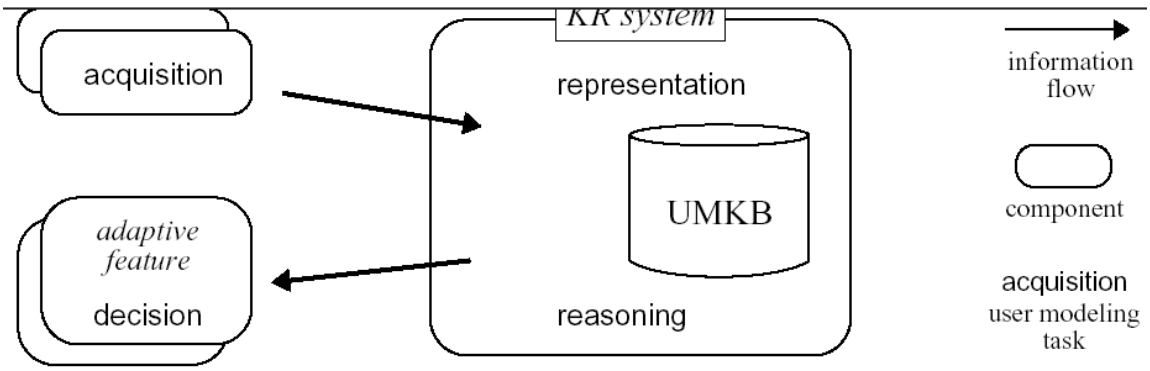
Knowledge representation based user modelling originated in research on (natural-language) dialog systems. Traditional user modelling systems often make use of knowledge representation techniques. Knowledge representation formalism offer facilities for maintaining knowledge bases (using representation formalisms) and for reasoning (using the inference procedures of representation formalisms).

Assumptions about individual characterisitcs of the user are maintained in a knowledge base, using a representation formalism. Since this knowledge base may additionally contain system knowledge about the application domain or meta-knowledge for inferring additional assumptions about the user from his/her current model (including stereotypes=predefined group models), it has been called user modelling knowledge base (UMKB [Pohl, 1998]). If available, inference procedures of the representation formalism or meta-level inferences can be used to expand the user model.

Some of the issues typical of systems that use knowledge representation for user modelling are the following:

- Acquisition of assumptions can be based on how user interacts with the system and can be controlled by heuristics like “if the user does ... then he/she is ...”
- User modelling shell systems usually represent both assumptions and domain knowledge using a concept formalism
- Once being added to the user model, an assumption may trigger meta-level reasoning that is based on concept relationships represented as domain knowledge in the UMKB
- The systems access both explicit and implicit user model contents to make its adaptivity decisions
- Acquisition and decision are performed outside the knowledge representation system, which is responsible for representation and reasoning
- The separate acquisition components often employ procedures or rules which are triggered by one or a few observations to construct an assumption about the user that is to be entered into the UMKB
- Acquisition process is not incremental (does not take observation history into account), which may lead to conflicts in the user model and, as a consequence, truth maintenance techiniques are needed to resolve these conflicts
- This user models mostly contain assumptions which are related to mental notions (knowledge, belief, goals, interests)

The following figure illustrates the application of knowledge representation methods to user modelling.



An example of this approach is **KN-AHS** [Kobsa, Müller and Nill, 1994], an adaptive hypertext system which makes use of the knowledge representation methods offered by the user modelling shell system BGP-MS to maintain assumptions about user knowledge.

Thus, the knowledge representation based approach uses a representation formalism to maintain assumptions about a user within a knowledge base. The reasoning mechanisms of the formalism are used to extend this user model and the (explicit or implicit) contents of the knowledge base can be accessed by an application and be used to support its adaptivity decisions. Usual knowledge representation based systems accept and deliver assumptions, make non-incremental acquisition, use explicit representation and are mainly mentalistic.

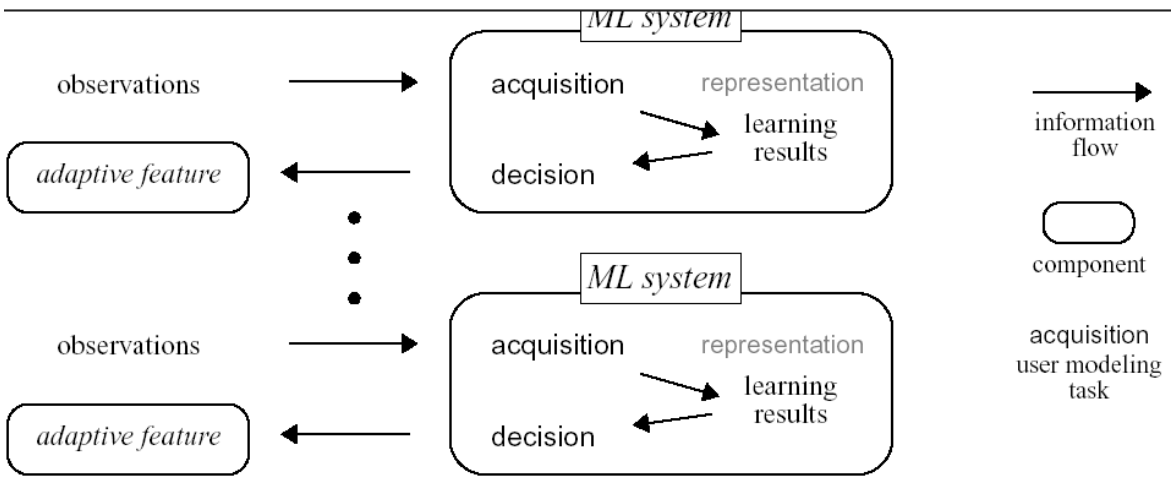
A 7.2.2 Machine Learning for User Modelling

This strand of research on user-adapted interaction related to the area of intelligent user interfaces started in the early nineties, with systems like **Flexcel** or interface agents and personal assistants, and more recently, systems using machine learning for personalized information filtering such as **Syskill & Webert** [Pazzani and Billsu, 1997], **Letizia** [Lieberman, 1995] or **Amalthaea** [Moukas, 1996]. Systems developed in this area typically observe system usage and determine usage regularities to form a usage profile that is to support adaptivity decisions. This procedure can be regarded as 'learning from observation'. Hence, it is no surprise that machine learning techniques have often been used to form usage profiles and support user-adapted interaction.

Machine learning methods process training input and offer support for decision (mainly classification) problems based on this input. Instead of a knowledge base like knowledge representation based systems, learning results are the central source of information about the user. The key issues are:

- Observations of user behavior are used to form training examples
- Learning components do acquisition by running their algorithms on these examples
- Learning results typically serve one specific decision process, which implies different learning processes for different adaptive features
- Representation is implicit, because format of learning results are specific to the learning algorithm used, which makes them difficult to be reused for other purposes
- Due to the lack of an independent representation formalism, there is no further reasoning based on already acquired data
- Decisions are directly supported
- Learning components can predict the user's reaction to new situations and use this prediction for their individualized suggestions
- Acquisition is incremental (history-aware), since it takes the history of interactions into account by processing a set of training examples
- It carries behavior-related information about the user (usage profile), except information filtering systems, which contain mentalistic assumptions

Usual machine learning based systems accept observations and deliver decisions, make incremental acquisition, have implicit representation and use behavior-related information (except information filtering systems that are mentalistic). The next figure illustrates the use of machine learning for user-adapted interaction.



If systems can maintain several user models, machine learning techniques can futher be used for group modelling, such as clustering user models into user group models. Then, individual user models may be complemented by suitable group information.

A 7.2.3 A Hybrid Approach for User Modelling

The hybrid approach to user modelling attemps to take an ideal by combining machine learning techniques and knowledge representation techniques. Machine learning techniques have the ability to support history-aware acquisition and decisions dynamically, but it is no easy for different decision processes to take advantage of learning results when these results only reflect usage regularities but do not explicitly represent individual user characteristics, nor communicate learning results to the user for inspection and explanation purposes.

An ideal user modelling system should have the following features:

- The possibility to report observations to the user modelling system, so applications would not be forced to form assumptions about the user on their own
- Acquisition should be incremental and not ignore interaction history, although it may be complemented by heuristic acquisition if quick results are needed or the number of observations is small
- Representation should be explicit, specially if assumptions about the user may be useful to more than one adaptive feature
- Assumptions about the user should not be restricted to be either bahavior-related or mentalistic
- The system should be able to support decisions, but also allow for direct access to the user model

A first step to integrate knowledge representation based and machine learning based user modelling was made by the user model server **Doppelgänger** [Orwant, 1995], which uses learning methods to process information from several sources. Learning results are represented explicitly in a standarized format and, as a consequence, several learning components can work on the acquisition of the same kind of data.

LaboUr [Pohl, 1997] is another user modelling architecture that integrates knowledge representation and machine learning mechanisms. This system accepts observations about the user, from which learning components or acquisition components may choose appropiate ones. Learning components, which are machine learning based, internally generate usage-related results that will be transformed into explicit assumptions, if possible, which are passed to a knowledge representation based user modelling subsystem. Acquisition components directly generate user model contents, which may be behavior-related or mentalistic, can implement heuristic acquisition methods but do not support decision. Therefore, acquisition components can allow for ‘quick-and-dirty’ acquisition from a small number of observations, in contrast to learning components which typically need a significant number of observations to produce learning results with sufficient confidence. There are also decision components that refer to user model contents, but the system offers also direct access to user models due to its use of explicit representation facilities.

Thus, LaboUr defines a framework for a hybrid approach that exploits the advantages of both explicit and implicit user profiles and uses clustering techniques for constructing user group models.

Bayesian Networks are another type of user modelling systems that is neither a knowledge representation nor a machine learning based system.

A 7.3 Assessment

A 7.3.1 Conclusions for Alfanet

Section A6.2 gives an overview of the techniques that can be used in user modelling, and as it has already been discussed, both approaches lack of important features (summarized next) and therefore a combined approach seems to be the best choice.

- The approach based on knowledge representation does not benefit from contextual and user interaction information, since it is centered on the application domain and on the personal characteristics of the user. In theory this approach can represent anything, but it is basically static and therefore, data from interaction are are difficult to update and rules that make inference taking these data into account are difficult to define a priori.
- The knowledge is explicitely declared in the first approach, and as a consequence, the model can be easily observed by the user. However, since this knowledge is based on a priori predefined acquisition rules which are usually complex, it is difficult to create the models, but it is even more difficult to keep the models updated, and user characteristics and needs evolve in time. On the other hand, user models constructed with machine learning techniques can be dynamically updated simply by using new training examples obtained from new user interactions. But because this model has to be learned, the model obtained is quite simple, and the results obtained are stored in the specific format of the learning algorithm used.
- The machine learning based one has its own drawbacks, too. The four critical issues that are limiting applications of machine learning in user modelling are the need for large data sets, the need for labeled data, concept drift and computational complexity. If supervised learning is used (which is the one actually used in most cases – clustering is also used, and it is non-supervised) lots of training examples are needed in order to produce learning results with sufficient confidence. To cope with this, theory refinement can be used to create a model by modifying an initial model. It is also possible to structure the task so that a learned model need not exactly replicate the user's decision by having more than one option available and not hindering the user from taking actions that were not anticipated, and as a result, the system does not have to have an accurate model to be useful.

  Supervised machine learning approaches also require explicitly labeled data, but the correct labels may not be readily apparent from simple observation of the user's behavior. The user must perform additional work to provide explicit feedback to the system but is not provided with an immediate reward. Besides, only the possitive examples obtained are usually reliable (determining negative examples is very difficult since e.g. if a user does not visit a link does not necessary mean that he/she is not interested in that link).

  Concept drift deals with the fact that the attributes that characterize a user are likely to change over time, and learning algorithms should be capable of adjusting to these changes quickly. An straightforward approach is placing less weight on older observations from the user, or limiting training data to an adjustable time window. Dual models can also be used, which classify instances by first consulting a model trained on recent data, and delegating classification to a model trained over a longer period if the recent model is unable to make a prediction with sufficient confidence.

  Computational complexity has also to be considered due to the sheer amount of information available as well as the number of users online. Many approaches proposed in academic research only care about improving predictive accuracy, but approaches that can be used in high-volume real-word scenarios are more useful, although they achieve worst accuracy. Anyway, computationally expensive algorithms can still be utilized in they can be applied in scenarios where models can be learned offline.

Therefore, we consider that a hybrid approach (knowledge representation – machine learning) should be chosen when defining the user model of the system to be developed in aLFanet Project, trying to get an effective integration of both machine learning and knowledge representation techniques. This proposal has to be further investigated in order to confirm its viability, but anyway, we will point out now some features that will have to be analysed:

- Explicit knowledge representation in the system to allow direct access to the user model and the use by different learning processes.
- No domain-dependent knowledge in the system, only generic acquisition rules and some reasoning based on the already acquired data to extend user model (the objective of aLFanet Project is self-learning for work in any subject; therefore, no domain-dependent knowledge can be used).
- Observations about user's behavior should be coded in the available representation formalism and used to form training examples.
- Incremental acquisition, taken into account past interactions using different weights depending on the antiquity of each interaction.
- Learning components used to learn from interaction tasks so the user model holds the changing needs.
- The learning results obtained by running the learning algorithms on the training examples should be transformed into explicit assumptions.
- Heuristic acquisition methods for quick and dirty acquisition from a small number of observations and short usage periods should be used.
- Model contents should be both behavior-related and mentalistic
- Decision should be supported based on learning results and on the user model contents themselves
- All user models should be kept in the system, and machine learning techniques should be used for group modelling by clustering user models into user group models. Collaborative-filtering can be done to reduce the amount of time to construct a reliable model of user interests. Perhaps an initial stereotyping could be done to provide some adpatation at the beginning, when the user is not known yet.

Because a hybrid approach is used for user modelling, reasoning will be done using the three possible techniques described: deductive reasoning, inductive reasoning and analogical reasoning.

A 7.4 References

Akoulchina, I. and Ganascia, J.-G. (1997). SATELIT-Agent: An Adaptive Interface Based on Learning Interface Agents Technology. In Jameson, A., Paris, C., and Tasso, C., eds.: *User Modelling: Proceedings of the Sixth International Conference*, Wien New York: Springer-Verlag, 21-32. http://www.um.org/um_97/gz/Akoulchinal.ps.gz

Ardissono, L. and Goy, A. (1999). Tailoring the Interaction with Users in Electronic Shops. In J. Kay, ed.: UM99 User Modelling: Proceedings of the Seventh International Conference. Wien New York: Springer-Verlag, 35-44. http://www.cs.usask.ca/UM99/Proc/ardissono.pdf

Ardissono, L., Goy, A., Meo, R. and Petrone, G. (1999). A Configurable System for the Construction of Adaptive Virtual Stores. *World Wide Web* 2(3), 143-159.

Ardissono, L., and Goy, A. (2000b). Tailoring the Interaction with Users in Web Stores. *User Modelling and User-Adapted Interaction* **10**(4): 251-303.

Balabanovic, M. (1997). An Adaptive Web Page Recommendation Service. In: *Proceedings of the 1st International Conference on Autonomous Agents*, Marina del Rey, CA, 378-385.

Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66-72.

Boyle, C. and Encarnacion, A. O. (1994). Metadoc: An Adaptive Hypertext Reading System. *User Modelling and User-Adapted Interaction*, 4(1):1-19.

Brusilovsky, P. and Pesin, L. (1994) ISIS-Tutor: An Intelligent Learning Environment for CDS/ISIS Users. Online Proceedings of CLCE'94, Joensuu, Finland, http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html.

Chen, J., Y. Yang and H. Zhang (2000). An Adaptive Web Content Delivery System. In: P. Brusilivsky, O. Stock and C. Strappavara, eds.: Adaptive Hypermedia and Adaptive Web-Based Systems. Berlin etc.: Springer, 284-288.

Chin, D. N. (1989). KNOME: Modelling what the User Knows in UC. In Kobsa, A. And Wahlster, W., editors, *User Models in Dialog Systems*, pages 74-107. Berlin, Heidelberg: Springer Verlag.

Chin, D.N. (2001) *Empirical Evaluation of User Models and User-Adapted Systems.* User Modelling and User-Adapted Interaction 11, 181-194.

De Rosis, F., Pizzutilo, S., Russo, A., Berry, D. C., and Molina, F. J. N. (1992). Modelling the user knowledge by belief networks. *User Modelling and User-Adapted Interaction*, 2(4):367-388.

Debevc, M., Meyer, B., Donlagic, D., and Svecko, R. (1996). Design and evaluation of an adaptive icon toolbar. *User Modelling and User-Adapted Interaction*, 6(1):1-21.

Fink, J.; Kobsa, A.; Nill, A. (1998): Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People. *The New Review of Hypermedia and Multimedia* 4, 163-188.

Fuller, R., and de Graaff, J. J. (1996). Measuring User Motivation from Server Log File. In Proceedings of the 'Designing for the Web: Empirical Studies', Redmond, WA. Microsoft Usability Group. http://www.microsoft.com/usability/webconf/fuller/fuller.htm

Gaudioso E., Boticario, J.G. *"User data management and usage model acquisition in an adaptive educational collaborative environment"*. To appear in Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems . Lecture Notes in Computer Science (LNCS), Málaga, Spain, 2002.

Gutiérrez, J., Pérez, T.A., Lopistéguy, P. and Usandizaga, I. (1995) *Sistemas Tutores Inteligentes: una Forma de Conseguir Sistemas Hipermedia Educativos*. CAEPIA'95.

Hohl, H., Böcker, H.-D., and Gunzenhäuser, R. (1996). HYPADAPTER: An Adaptive Hypertext System for Exploratory Learning and Programming, *User Modelling and User-Adapted Interaction*, 6 (2-3), 131-155.

Holynski, M. (1988). User-adaptive Computer Graphics. *International Journal of Man-Machine Studies*, 29:539-548.

Höök, K., Karlgren, J., Waern, A., Dahlbäck, N., Jansson, C., Karlgren, K., and Lemaire, B. (1996). A Glass Box Approach to Adaptive Hypermedia, *User Modelling and User-Adapted Interaction,* 6(2/3), 157-184.

Horvitz, E. (1997). Agents with Beliefs: Reflections on Bayesian Methods for User Modelling. In Jameson, A., Paris, C., and Tasso, C., eds.: *User Modelling: Proceedings of the Sixth International Conference*, Wien, New York. Springer-Verlag, 441-442. http://www.um.org/um_97/gz/HorvitzE.ps.gz

Horvitz, E., Breese, , J., Heckerman, D., Hovel, D. and Rommelse, K. (1998). The Lumière Project: Bayesian User Modelling for Inferring the Goals and Needs of Software Users. Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 256-265.

Jameson, A., Schäfer, R., Simons, J., and Weis, T. (1995). Adaptive Provision of Evaluation-oriented Information: Tasks and techniques. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada. San Mateo, CA: Morgan Kaufmann, 1886-1893.

Joachims, T., Freitag, D., and Mitchell, T. (1997). Webwatcher: A Tour Guide for the World Wide Web. In: *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*. Palo Alto, CA: Morgan Kaufmann Publishers.

Joerding, T., Michel, S., and Popella, M. (1998). Tellim - Ein System für Adaptive Multimediale Produktpräsentationen im World Wide Web. In Timm, U. J. and Rössel, M., eds.: *ABIS-98 – 6. Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen*, 29-40, Erlangen. FORWISS. http://www-mmt.inf.tudresden.de/joerding/abis98/abis98.html

Kaplan, C., Fenwick, J., and Chen, J. (1993). Adaptive Hypertext Navigation Based on User Goals and Context, *User Modelling and User-Adapted Interaction*, 3(3): 193-220.

Kass, R. (1991). Building a User Model Implicitly from a Cooperative Advisory Dialog. *User Modelling and User-Adapted Interaction*, 1(3):203-258.

Kobsa, A., Müller, D., and Nill, A. (1994). KN-AHS: An Adaptive Hypertext Client of the User Modelling System BGP-MS. In*: Proceedings of the Fourth International Conference on User Modelling*, Hyannis, MA, 99-105. Reprinted in: M. T. Marbury and W. Wahlster, eds. (1998): Intelligent User Interfaces. San Mateo, CA: Morgan Kaufman, 372-378. http://ics.uci.edu/~kobsa/papers/1994-UM94-kobsa.ps

Kobsa, A., Koenemann, J. and Pohl, W. (2001) *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*. The Knowledge Engineering Review 16(2), 111-155. Cambridge University Press.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3): 77-87.

Krogsæter, M., Oppermann, R., and Thomas, C. G. (1994). A user interface integrating adaptability and adaptivity. In: Oppermann, R. (Ed.) (1994). *Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software.* Hilldale, New Jersey: Lawrence Erlbaum, 97-125.

Langley, P. (1999). User modelling in adaptive interfaces. *Proceedings of the Seventh International Conference on User Modelling* (pp. 357-370). Banff, Alberta: Springer.

Lieberman, H. (1995). Letizia: An agent that assists web browsing. In: *Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Canada*. San Mateo, CA: Morgan Kaufmann.

Mislevy, R. J. and Gitomer, D. H. (1996). The Role of Probability-based Inference in an Intelligent Tutoring System. *User Modelling and User-Adapted Interaction*, 5(3-4), 253-282.

Moukas, A.G. (1996) Amalthaea: Information discovery and filtering using a multi-agent evolving ecosystem. In *Proceedings of the Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*.

Oppermann, R. and Specht, M. (1999). Adaptive Information for Nomadic Activities. A Process Oriented Approach. In: Arend, U., Eberleh, E., Pitschke, K. (eds.): Software-Ergonomie '99. Design von Informationswelten. Stuttgart, Leipzig: B. G. Teubner, 256 - 264.

Oppermann, R. and Specht, M. (2000): A Context-sensitive Nomadic Information System as an Exhibition Guide. Proceedings of the Handheld and Ubiquitous Computing Second International Symposium, HUC 2000, Bristol, UK 127 - 142.

Orwant, J. (1995). Heterogeneous Learning in the Doppelgänger User Modelling System. *User Modelling and User-Adapted Interaction*, 4(2): 107-130.

Pazzani, M. and Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27:313-331.

Pohl, W. (1997) LaboUr – machine learning for user modelling. In Smith, M.J. , Salvendy, G. And Koubek, R.J. (Ed.) *Design of Computing Systems: Social and Ergonomic Considerations (Proceedings of the Seventh International Conference on Human-Computer Interaction)* vol. B, pags. 27-30. Amsterdam. Elsevier Science.

Pohl, W. (1998) *Logic-Based Representation and Reasoning for User Modelling Shell Systems.* Number 188 in Dissertationen zur künstlichen Intelligenz (DISKI) infix, Sankt Augustin.

Pohl, W. and Nick, A. (1999). Machine learning and knowledge-based user modelling in the LaboUr approach. In J. Kay, ed.: UM99 User Modelling: Proceedings of the Seventh International Conference. Wien New York: Springer-Verlag,179-188. http://www.cs.usask.ca/UM99/Proc/pohl.pdf

Pohl, W. y Nick, A. (1999) *Machine Learning and Knowledge Representation in the LaboUr Approach to User Modelling*. GMD FIT, HCI Research Department. Germany.

Pohl, W., Schwab, I. and Koychev (1999) *Learning About the User: A General Approach and Its Application*. In Proceeding of IJCAI'99 Workshop "Learning About Users", Stockholm, Sweden.

Popp, H. and Lödel, D. (1996). Fuzzy techniques and user modelling in sales assistants. *User Modelling and User-Adapted Interaction*, 5(3-4): 349-370.

Ross, E. (2000) *Intelligent User Interfaces: Survey and Research Directions*. University of Bristol. United Kingdom.

Sánchez Villalobos, F. (2000) *Modelado del usuario para ambientes de aprendizaje colaborativo en Internet.* Tesis. Universida de las Américas-Puebla. México.

Shardanand, U. and Maes, P. (1995). Social Information Filtering: Algorithms for Automating Word of Mouth. In; Proceedings of the Human Factors in Computing Systems Conference (CHI-95). New York, NY: ACM Press, 210-217.

Thomas, C. G. and Fischer, G. (1996). Using Agents to Improve the Usability and the Usefulness of the World-Wide Web. Fifth International Conference on User Modelling, Kailua-Kona, HI, 5-12.

Thomas, C. G. and Krogsæter, M. (1993). An Adaptive Environment for the User Interface of Excel. In: *Proceedings of Intelligent User Interfaces '93*. ACM Press, 123-130.

Webb, G.I., Pazzani, M.J. and Billsus, D. (2001) *Machine Learning for User Modelling.* User Modelling and User-Adapted Interaction 11,19-29.

Weber, G. and Specht, M. (1997). User Modelling and Adaptive Navigation Support in WWW-based Tutoring Systems. In Jameson, A., Paris, C., and Tasso, C., editors, *User Modelling: Proceedings of the Sixth International Conference*. Wien, New York: Springer-Verlag, 289-300. http://www.um.org/um_97/gz/WeberG.ps.gz

Appendix 8 Intelligent agents and multi-agent architectures

A 8.1 Overview

A 8.1.1 Agents

A 8.1.1.1 Definitions

It can be said that *Autonomous agents are computational systems that inhabit some complex, dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks that they are designed for* [Maes, 1995].

However, researchers do not share the same vision or notion of what agents are. Two meanings of the term 'agent' can be distinguished [Wooldrige and Jennings, 1995], a weaker one and a stronger one.

A Weak Notion of Agency [Wooldrige and Jennings, 1995]

The most common way in which the term 'agent' is used is to denote a hardware or (more usually) a software-based computer system with the following properties:

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

A Stronger Notion of Agency [Wooldrige and Jennings, 1995]

For some researchers —particularly those working in Artificial Intelligence —the term 'agent' has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualized or implemented using concepts that are more usually applied to humans. For example, it is quite common in Artificial Intelligence to characterize an agent using mentalistic notions, such as knowledge, belief, intention, and obligation. Some Artificial Intelligence researchers have gone further, and considered emotional agents. Another way of giving agents human-like attributes is to represent them visually, perhaps by using a cartoon-like graphical icon or an animated face — for obvious reasons, such agents are of particular importance to those interested in human-computer interfaces.

A 8.1.1.2 Classifications

There exist many ways to classify software agents. A classification can be made using the following categories [Nwana, 2000]:

- Collaborative agents
- Agents of Internet/Information
- Intelligent Agents
- Reactive Agents
- Agents of Learning
- Agents of Interface
- Movable Agents
- Hybrid Agents
- Agents of Recommendation
- Filtrate Agents

A 8.1.2 Multi-Agent System (MAS)

A 8.1.2.1 Definition

As seen from Distributing Artificial Intelligent, a multi-agent system is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity. More recently, the term multi-agent system has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [Flores-Méndez, 1999]:

- each agent has incomplete capabilities to solve a problem
- there is no global system control
- data is decentralized
- computation is asynchronous

Therfore, a multi-agent system should have the following skills [Camacho, 2002]:

- Social Organization
- Coordination
- Cooperation
- Negotiation
- Communication

A 8.1.2.2 Classification

A possible classification of the architectures used for building agents-based systems can be obtained from the reasoning model used by agents [Camacho, 2002]:

- **Deliberative**: uses a symbolic knowledge representation model
- **Reactive:** no internal symbolic reasoning model exist, but the system acts according to a stimulus-answer behavior model.
- **Hybrid:** both previous architectures are combined, so that the reasoning is done in a symbolic way and the interaction with the outside is done in a reactive way.

Multi-agent systems can be also classified according to their organization, that is, according to the way they structure themselves when interacting among them [Sycara, 1998]:

- **Hierarchy**: The authority for decision-making and control is concentrated on a single problem solver (or a specialized group) at each level in the hierarchy. Interaction is through vertical communication from superior to sub-ordinate agents, and vice versa. Superior agents exercise control over resources and decision-making.
- **Community of experts**: This organization is flat, where each problem solver is a specialist in some particular area. The agents interact by rules of order and behavior [Lewis and Sycara, 1993; Lander, Lesser, and Connell, 1991]. Agents coordinate though mutual adjustment of their solutions so that overall coherence can be achieved.
- **Market**: Control is distributed to the agents that compete for tasks or resources through bidding and contractual mechanisms. Agents interact through one variable, price, which is used to value services [Müllen and Wellman, 1996; Davis and Smith, 1983; Sandholm, 1993]. Agents coordinate through mutual adjustment of prices.
- **Scientific community**: This model shows how a pluralistic community can work [Kornfeld and Hewitt, 1981]. Solutions to problems are locally constructed, then they are communicated to other problem solvers that can test, challenge and refine the solution [Lesser, 1991].

Another classification distinguishes collaboration rules among agents. Although different agents community management policies exist, they can be summarized into the next to [Giráldez, 1999]:

- **Negotiation**: the agents have their own interests and must negotiate with the rest of agents to obtain aid or collaboration. The main characteristics of negotiation that are necessary for developing applications in the real world [Sycara, 1998] are: (1) the presence of some sort of conflict that must be resolved in a decentralized manner by (2) self-interested agents under

conditions of (3) bounded rationality and (4) incomplete information. Different theories exist to make the communication possible, like the theory of games, or different types taken from market laws.

- **Collaboration**: the agents help any other agent who asks for it.

Multi-agent systems have to be characterized by learning, since, by definition, a multi-agent system has to learn of its own experience [Giráldez, 1999].

A 8.1.2.3 Middle agents [Flores-Méndez, 1999]

There is a need for mechanisms for advertising, finding, fusing, using, presenting, managing, and updating agent services and information. To address these issues, the notion of *middle agents* [Decker, Sycara and Williamson, 1997] was proposed. Middle agents are entities to which other agents advertise their capabilities, and which are neither requesters nor providers from the standpoint of the transaction under consideration. The advantage of middle agents is that they allow a multi-agent system to operate robustly when confronted with agent appearance, disappearance, and mobility.

There are several types of agents that fall under the definition of middle agents. Note that these types of agents, which are described below, are defined so vaguely that sometimes it is difficult to make a clear differentiation between them.

- **Facilitators**: agents to which other agents surrender their autonomy in exchange for the facilitator's services [Bradshaw, 1997]. Facilitators can coordinate agents' activities and can satisfy requests on behalf of their subordinated agents.
- **Mediators**: agents that exploit encoded knowledge to create services for a higher level of applications [Wiederhold, 1992].
- **Brokers**: agents that receive requests and perform actions using services from other agents in conjunction with their own resources [Decker, Williamson and Sycara, 1996].
- **Matchmakers** and **yellow pages**: agents that assist service requesters to find service provider agents based on advertised capabilities [Bradshaw, Dutfield, Benoit and Woolley, 1997] [Decker, Williamson and Sycara, 1996].
- **Blackboards**: repository agents that receive and hold requests for other agents to process [Nii, 1987][Cohen, Cheyer, Wang and Baeg, 1994].

A 8.1.3 Intelligent systems oriented to the education

The use of Artificial Intelligence techniques in programs aimed at supporting education-learning environments goes back at the beginning of the decade of the 70, like the SCHOLAR system [Carbonell, 1970]. Carbonell's proposal created an historical framework for ITS (Intelligent Tutorial Systems). The goal was to construct a group of programs where a model of the users was used to adapt the presentation of contents to the profile of each student.

According to Oliveira [1994], ITSs have a basic organization, where the functional components can be divided into:

- *Student Module:* stores the specific information of each student, such how is the student working the course materials.
- *Tutorial Module (or Pedagogical Module),* offers a methodology for the learning process. It knows the strategy to follow based on the student characteristics. Therefore, this module has to interact with the Student Module to get this information.
- *Dominion Module:* stores the information that will be taught to the learner. To have a model of the knowledge to be given to each student at any time is very important for the success of the system.
- *Interface Module:* intermediate module for the interaction between the system and the user.

Up to now, these systems were applications that integrated all these modules in a sequential way. Nevertheless, agent technology can provide a better way of conceptualizing and/or implementing ITS applications. Three important domain characteristics for adopting agent technology are [Bond and Gasser, 1988]: (1) data, control, expertise, or resources are inherently distributed, (2) the system is naturally

regarded as a society of autonomous cooperating components, or (3) the system contains legacy components, which must be made to interact with other, possibly new software components.

As the reader can see, ITSs match very well with agent technologies, where each module could be an agent. Other advantages could be [Giraffa, 1998] the following ones:

- to maintain a distributed knowledge in the system among several tutors, each one with his/her own beliefs, desires, objectives, emotions or plans of performance
- the student would interact with each tutor in a more flexible way
- the student could pass knowledge to the tutors that could be passed to other students later

As it is said in [Webber, 2001], Web-based technologies in conjunction with multi-agent methodology form a new trend in modelling and development of learning environments. Multi-agent methodology has recently appeared as an alternative to conceive distributed learning applications. The main reasons of this are the evolution of multi-agent technology itself and the fact that multi-agent methodology deals well with applications where crucial issues, such as distance, cooperation among different entities and integration of different components of software, are found.

A 8.2 Solutions

A 8.2.1 Categorization of existing systems

Deliberative systems are [Wooldrige and Jennings, 1995]:

- STRIPS [Fikes and Nilsson, 1971]. This system takes a symbolic description of both the world and a desired goal state, and a set of action descriptions, which characterize the pre- and post- conditions associated with various actions.
- *Intelligent Resource-bounded Machine Architecture* (IRMA) [Bratman, Israel and Pollack, 1988]. This architecture has four key symbolic data structures: a plan library, and explicit representations of beliefs, desires, and intentions.
- HOMER [Vere and Bickmore, 1990]. This agent is a simulated robot submarine, which exists in a two-dimensional 'Seaworld', about which it has only partial knowledge.
- GRATE [Jennings, 1993]. GRATE is a layered architecture in which the behavior of an agent is guided by the mental attitudes of beliefs, desires, intentions and joint intentions.

Reactive systems are [Wooldrige and Jennings, 1995]:

- PENGI [Agre and Chapman, 1987] Agre observed that most everyday activity is 'routine' in the sense that it requires little —if any — new abstract reasoning. Most tasks, once learned, can be accomplished in a routine way, with little variation. PENGI is a simulated computer game, with the central character controlled using a scheme such as that outlined above.
- Agent Network Architecture [Maes, 1989; Maes, 1990; Maes, 1991]. Maes has developed an agent architecture in which an agent is defined as a set of competence modules.

Hybrid systems are [Wooldrige and Jennings, 1995]:

- Procedural Reasoning System (PRS) [Georgeff and Lansky, 1987]. The PRS is a belief-desire-intention architecture, which includes a plan library, as well as explicit symbolic representations of beliefs, desires, and intentions. A PRS plan library contains a set of partially elaborated plans, called knowledge areas (KAs), each of which is associated with an invocation condition. This condition determines when the KA is to be activated. KAs may be activated in a goal-driven or data-driven fashion; KAs may also be reactive, allowing the PRS to respond rapidly to changes in its environment. The set of currently active KAs in a system represent its intentions.
- TOURINGMACHINES hybrid agent architecture [Ferguson, 1992a; Ferguson, 1992b]. The architecture consists of perception and action subsystems, which interface directly with the agent's environment, and three control layers, embedded in a control framework, which mediates between the layers. Each layer is an independent, activity-producing, concurrently executing process.

- INTERRAP [Müller and Pischel, 1994; Müller et al., 1995; Müller, 1994], is a layered architecture, with each successive layer representing a higher level of abstraction than the one below it.

The agents of a multi-agent system can collaborate before a request of aid of another agent takes part. They are arranged to collaborate without no type of negotiation exists. Some systems that present/display this characteristics are MMS [Brauer and Weiss, 1998] and LOPE [García-Martinez and Borrajo, 1998].

The negotiation among agents depends on the organization of the multi-agent systems and the rules of collaboration that exist among them. An example of organization and negotiation by the market is I-Help [Vassileva et al., 2001], which is described in the next subsection.

A 8.2.2 Multi-agent systems for the development of Intelligent Systems oriented to education

This section briefly describes some of the most recent approaches of Intelligent Systems oriented to education:

- White Rabbit [Thibodeau et al, 2000]. The White Rabbit system intends to enhance cooperation among a group of people by analysing their conversation. Each user is assisted by an intelligent agent, which establishes a profile of his or her interests. Next, with its autonomous and mobile behaviour, the agent will reach the personal agents of other users to be introduced and presented to the ones that seem to have similar interests. A mediator agent is used to facilitate communication among personal agents and to perform clustering on the profiles that they have collected. Conversation between users takes place in a chat environment adapted to the needs of the system.
- LeCS [Learning from Case Studies]. This is an intelligent system for remote education that has, according to Rosatelli et al. [2000], an architecture based on a Federal System of agents. LeCS supports web-based distance learning from case studies, allowing collaborative learning between a group of learners that is geographically dispersed. It provides the necessary tools to carry out the case solution development and accomplishes functions that altogether assist the learning process. LeCS is used to give CSCL (Computer Supported Collaborative Learning) through the Web. The method of machine used Learning is based on CBR (Case Based Reasoning).
  In this agent-agent architecture direct communication does not exist, but all is done by a special agent called Facilitator. This Facilitator is in charge to store all the information needed for the communication. Three types of agents exist in the system:
  - Agent Interface: stores the individual interactions of each user
  - Information Agent: stores information regarding didactic materials (HTML page, images, interactions on the chat, etc) and keeps a knowledge base for the solutions of the developed cases
  - Advisor Agent: has mechanisms to guess situations in which an aid to the user is needed
- Baghera [Webber et al., 2001]. The Baghera platform is founded on the principle that the educational function of a system is an emerging property of the interactions organized between its components: agents and humans, and not a mere functionality of one of its parts. Their first achievements include a web-based multi-agent architecture for learning environments and an operational prototype for the learning of geometry. Students and teachers interact with different agents, according to the activities they will carry on and the educational approach of Baghera. Each student is supported by three artificial agents:
  - Student's Personal Interface Agent: associated with the student's interface
  - Tutor Agents: can interact with mediator agents, assistant agents and other tutors
  - Mediator Agent: the aim of this agent is to choose an appropriate problem solver to send the student's solutions

  In a similarly way, two artificial agents gives support to each teacher:
  - Teacher's Personal Interface Agent: associated with the teacher's interface
  - Assistant Agent: a kind of personal agent whose goals include assisting the teacher with the creation and distribution of new activities, which are kept in the teacher's electronic folder.
- I-Help [Vassileva et al., 2001]. I-Help is based on a multi-agent architecture, consisting of personal agents (of human users) and application agents (of software applications). These agents use a common ontology and communication language. Each agent manages specific resources of the user (or application) it represents, including for example, the knowledge resources of the user about certain concepts, or the instructional materials belonging to an application. The agents use their resources to achieve the goals of their users, their own goals, and goals of other agents. Thus all the agents are autonomous and goal-driven. In their goal pursuit the agents can also use resources borrowed from other agents, i.e. they are collaborative. For this they have to negotiate. Each agent

possesses a model of its user and of other agents it has encountered and negotiated with. The agents communicate with each other and with matchmaker agents to search for appropriate help resources for their users, depending on the topic of the help-request. If an electronic resource is found (represented by application agents), the personal agent "borrows" the resource and presents it to the user in a browser. However, if a human helper is located, the agents negotiate the price for help, since human help involves inherent costs (time and effort) for the helper. Help is arranged (negotiated) entirely by the personal agents, thus freeing the learner from the need to bargain and think about the currency spent / earned. In this way the personal agents trade the help of their users on a virtual help market. Thus the multi-agent architecture involves various levels of organization, including the negotiation between agents, an economical model and control / policing institutions. In this way, we achieve a distributed (multi-user, multi-application) adaptive (self-organized) system that supports users in locating and using help resources (other users, applications, and information) to achieve their goals.

- AME-A [D'Amico et al., 1997; D'Amico et al., 1998; Pereira et al., 2001]. AME-A is an education-learning multi-agent which sets out the study and the development of an interactive educational system for education. The proposal is generic and adapts education to the psico-pedagogical characteristics of the student.

  The system uses both static learning and dynamic one. The static learning corresponds to the first interaction of the student with the environment, where an agent models the apprentice according to his/her affective characteristics, motivation and level of knowledge. The dynamic learning takes places during the interaction, when the student model (like the pedagogical strategies in force) is validated.
- Electrotutor [D'Amico et al., 1997; D'Amico et al., 1998; Pereira et al., 2001]. Electrotutor III implements distributed evironments of intelligent education-learning based on a multi-agent architecture for teaching Physics. Agents dynamic perceive the conditions of their environment and make decisions to change it. Seven agents, each one of them with a specific function compose the society of agents. In order to be able to act on the environemt, each agent has an internal partial representation of the world that surrounds it. The metaphor of mental states is used to model this way the knowledge base that represents the states of the environment where the agent is living. The seven agents are:
  - Dominion Managing Agent: recovers information referring to the dominion on which the student is going to work
  - Exercises Managing Agent: provides exercises and their answers to the student
  - Examples Managing Agent: provides examples to the student
  - Activities Managing Agent: in charge to provide extra activities to the student
  - Student Model Agent: in charge to construct and to maintain a knowledge base that models the state of the students who are or been have connected to the system
  - Agent Interface: controls what appears on the Navigator (an agent interface exist per student).
  - Communication Managing Agent: in charge of the communication of each agent Interface with the others.
- JADE [Silveira, 2000]. This environment contains a special agent responsible for each teaching strategy developed, that is, for the domain knowledge retrieval over each point to be presented to the student, for the task of proposing exercises and evaluating proposals, examples and extra activities. JADE architecture encompasses, therefore, a Multi-Agent environment composed of an agent responsible for the system general control (Student s Model), and a Communication Manager and other agents (Pedagogical Agents), which are responsible for tasks related to their teaching tactics, where each agent may have its tasks specified according to its goal. All actions of student's data accessing are taken by the Student s Model, thus when a pedagogical agent is required to update the student's historic, this agent sends to the Student Model data to be updated, as well as any other change in the student s state of teaching.
- GRACILE [Ayala-Yano, 1996; Ayala-Yano, 1998]. For Computer-Supported Collaborative Learning (CSCL) environments they propose intelligent agents that assist the learners and cooperate in order to create possibilities of effective collaboration in a virtual community of practice. They have developed two kinds of software agents: mediator agents which play the role of facilitators that support the communication and collaboration among learners, and domain agents, which provide assistance concerning the appropriate application of domain knowledge in the network. Mediator agents cooperate exchanging their beliefs about the capabilities, commitments and goals of the learners. Doing this each mediator agent is able to construct a representation of its learner's collaboration possibilities in the group (referred as the learner's group-based knowledge frontier), considering the social and structural aspects of knowledge development. The mediator agent

proposes the learner to commit to tasks that require the application of knowledge elements in the learner's group-based knowledge frontier, which results in an increment of the collaboration possibilities between learners, the creation of zones of proximal development, and therefore more learning possibilities.

- A Computational Model of Distance Learning Based on Vygotsky's Social-Cultural Approach [Andrade et al., 2001]. This framework is based on Vygotsky's social-cultural theory and is designed as a multi-agent society supporting distance learning. The goal of this research is to propose an environment that privileges collaboration as form of social interaction, through the use of language, symbols and signs. To support collaborative learning, they present a society formed of the following artificial agents: ZPD agents, mediating agents, semiotic agent and social agent; it also involves human agents who have either the role of tutors or learners.
- ABITS: An Agent Based Intelligent Tutoring System for Distance Learning [Capuano, 2000]. ABITS is able to support a Web-based Course Delivery Platform with a set of "intelligent" functions providing both student modelling and automatic curriculum generation. Such functions found their effectiveness on a set of rules for knowledge indexing based on Metadata and Conceptual Graphs following the IEEE Learning Object Metadata (LOM) standard. Moreover, in order to ensure the maximal flexibility, ABITS is organized as a multi-agent system composed by pools of three different kind of agents (evaluation, pedagogical and affective agents). Each agent is able to solve in autonomous way a specific task and they work together in order to improve the WBT learning effectiveness adapting the didactic materials to user skills and preferences.

A 8.3 Assessment

A 8.3.1 Conclusions for Alfanet

Multi-agent systems can be used to reach the adaptation and personalization in the platform. In order to achieve good personalised and adaptive e-learning, the system could make use of heterogeneous agents that combine the solutions learned with different biases corresponding to different machine learning methods. It is thus possible to apply different approaches to different tasks and clarify which option is the most appropriate for each instance.

The flexibility and robustness obtained from multi-agent systems makes them suitable to be used in Web environments. Multi-agent systems oriented to Web will be able to learn using machine learning tools and could simulate an expert when making pedagogical tasks.

Based on the above considerations, multi-agent systems improve the system by doing it viable in opened knowledge domains, where ITS were earlier applied with less success.

The use of machine learning tools is needed since the environment will evolve both with time and users. The system will have to personalize the interactions with the users using these techniques.

A 8.4 References

Ayala, G., Yano, Y. Intelligent Agents to Support the Effective Collaboration in a CSCL Environment, Proceedings of the ED-TELECOM 96 World Conference on Educational. 1997. Communications, June 17 - 22, 1996, Boston, Mass. AACE, Patricia Carlson and Fillia Makedon (Eds.) pp. 19-24 (Ayala-Yano 1998) Ayala, G., Yano, Y. A Collaborative Learning Environment Based on Intelligent Agents, Expert Systems with Applications, Pergamon Press, pp. 129-137. 1998

Andrade, Adja F. de; Jaques, Patrícia A.; Jung, João Luiz; Bordini, Rafael H.; Vicari, Rosa M.. A Computacional Model of distance Learning Based on Vygotsky´s Socio-cultural Approach. Workshop Multi-Agent Architectures for Distribuited Learning Environments. Procedings International Conference on AI and Education, San Antonio, Texas, May, 2001. P.33-40.

Bond, A. H., Gasser, L. (Eds.) (1988) Readings in Distributed Artificial Intelligence. Morgan Kaufmann.

Bradshaw, J.M. An Introduction to Software Agents. In: *Software Agents*, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 3-46.

Bradshaw, J.M., Dutfield, S., Benoit, P. and Woolley, J.D. KAoS: Toward An Industrial-Strength Open Agent Architecture. In: *Software Agents*, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 375-418.

Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355.

Brauer, W. and Weiss, G. Multi-machine scheduling, a multi-agent approach. In *Proceedings of the Third International Conference on Multi Agent Systems*, pages 42-48, 1998.

Camacho Fernández, D. Recuperación e Integración de Información disponible en Web para la Resolución de Problemas, Universidad Carlos III de Madrid, 2002.

Capuano, N.; Marsella, M.; Salerno, S. ABITS: An Agent Based Intelligent Tutoring System for Distance Learning. In : Proceedings of the International Workshop in Adaptative and Intelligent Web-based Educational Systems. Available at http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000/.

Carbonell, J. R. AI in CAI: na artificial intelligence approach to computer assisted instruction. IEEE Transactions on Man Machine Systems, v.11, n.4, p.190-202, 1970.

Cohen, P.R., Cheyer, A., Wang, M., and Baeg, S.C. An open agent architecture. In: *Proceedings of the AAAI Spring Symposium*. 1994.

Decker, K., Sycara, K. and Williamson, M. Middle-Agents for the Internet. In: *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI-97)*, January, 1997.

Decker, K., Williamson, M. and Sycara, K. Matchmaking and Brokering. In: *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, December, 1996.

D'Amico, C.B.; Viccari, R.M.; Alvares, L.O. A Framework for Teaching and Learning Environments. In: Simpósio de informática Na educaçao, VIII, 1977, Sao Paulo, SP.

D'Amico, C.B.; Pereira, A.S.; Geyer, C.F.R.;Viccari, R.M. Adapting Teaching Strategies in a Learning Environment on WW. In: Proceedings of the WebNet World Conference of the WWW, Internet & Intranet, Florida, USA. 1998.

Ferguson, I. A. (1992a). *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agent*s. PhD thesis, Clare Hall, University of Cambridge, UK. (Also available as Technical Report No. 273, University of Cambridge Computer Laboratory).

Ferguson, I. A. (1992b). Towards an architecture for adaptive, rational, mobile agents. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91*), pages 249–262. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Fikes, R. E. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189–208.

Flores-Mendez R.A. Standardization of Multi-Agent System Frameworks, University of Calgary 1999.

Frasson, Claude; Martin, Louise; Gouarderes, Guy; Aimeur, Esma. LANCA: A Distance Learning Architecture Based on Networked cognitive Agents. In lecture Notes in Computer Science. Intelligent Tutoring Systems. Proceedings of 4th International Conference, ITS 1998, San Antonio, Texas, August 1998. P. 594-603.

Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87*), pages 677–682, Seattle, WA.

Giráldez Betrón, J.I. Modelo de Toma de Decisiones y Aprendizaje en Sistemas Multi-Agente, Universidad Politécnica de Madrid, 1999.

Giraffa, L. M. M. Uma arquitetura de tutor utilizando estados mentais. Porto Alegre: CPGCC da UFRGS, 1998. Tese de Doutorado.

Gracía-Martínez, R. and Borrajo, D. Planning, learning and executing in autonomous systems. In S. Steel, editor, *Recent Advances in AI planning. 4th European Conference on Planning ECP'97*, number 1348 in Lecture Notes in Artificial Intelligent, pages 208-220. Springer-Verlag, 1997.

Jennings, N. R. (1993). Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information System*s, 2(3):289–318.

Maes, P. (1989). The dynamics of action selection. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), pages 991–997, Detroit, MI.

Maes, P. (1990). Situated agents can have goals. In Maes, P., editor, *Designing Autonomous Agent*s, pages 49–70. The MIT Press: Cambridge, MA.

Maes, P. (1991). The agent network architecture (ANA). *SIGART Bulleti*n, 2(4):115–120.

Maes, P. "Artificial Life meets Entertainment: Interacting with Lifelike Autonomous Agents." Special Issue on New Horizons of Commercial and Industrial AI, Vol. 38, No. 11, pp. 108-114, Communications of the ACM, ACM Press, November 1995.

Müller, J. P. (1994). A conceptual model for agent interaction. In Deen, S. M., editor, *Proceedings of the Second International Working Conference on Cooperating Knowledge Based Systems (CKBS-94*), pages 213–234, DAKE Centre, University of Keele, UK.

Müller, J. P. and Pischel, M. (1994). Modelling interacting agents in dynamic environments. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94*), pages 709–713, Amsterdam, The Netherlands.

Müller, J. P., Pischel, M., and Thiel, M. (1995). Modelling reactive behaviour in vertically layered agent architectures. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890*), pages 261–276. Springer-Verlag: Heidelberg, Germany.

Nii, H.P. Blackboard Systems. In: *The Handbook of Artificial Intelligence*, A. Barr, P.R. Cohen and E.A. Feingenbaum (Eds.), Addison-Wesley, New York, 1989, Volume IV, chapter XVI, pages 1-82.

Nwana, H. S. and Ndumu, D. T. A Perspective on Software Agents Research, Nwana, H. S. and Ndumu, D. T. IN: The Knowledge Engineering Review, 14(2), pages 125-142, Cambridge University Press 2000.

Oliveira, F. M.. Critérios de equilibração para Sistemas Tutores Inteligentes. Porto Alegre: CPGCC da UFRGS, 1994. Tese de Doutorado.

Pereira, A.S.; D'Amico, C.B.; Geyer C.F.R. Gerenciamiento do Conhecimiento do ambiente AME-A. http://www.inf.ufrgs.br/~adriana/vcied.doc

Rosatelli, M.C.; Self, J.A.; Thiry, M. LeCS: a collaborative case study system, in G. Gauthier, C. Frasson and K. VanLehn(eds.), Intelligent Tutoring Systems, Berlin: Springer-Verlag, p232-241, 2000.

SILVEIRA, Ricardo. Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino - JADE - Java Agent framework for Distance learning Environments Porto Alegre: CPGCC da UFRGS, 2000. Dissertação de Mestrado. http://www.inf.ufrgs.br/~rsilv/Jade/jade.html

Sycara, K.. Multiagent Systems. AI Magazine, vol. 18, nº 2, 1998.

Thibodeau, Marc-André; Bélanger, Simon; Frasson, Claude. Matchmaking of User Profiles Based on discussion analysis Using Intelligent Agents. In Lectures Notes in Computer Science. Intelligent Tutoring Systems. Proceedings of 5th International Conference, ITS 2000, Montreal, Canada, June 2000. P.113-122.

Vassileva, Julita; Detters, Ralph; Geer, Jim; Maccalla, Gord; Bull, Susan; Kettel, Lori. Lessons from Deploying I-Help. Workshop - Multi-Agent Architectures for Distributed Learning Environments. Proceedings of International Conference on AI and Education, San Antonio, Texas, May, 2001. P.3-11.

Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C. And Grant, S. *ARIES* Lab, A Multi-agent Approach to the Design of Peer´help Enviroments. AIED'99 Submission, 1999.

Vere, S. and Bickmore, T. (1990). A basic agent. *Computational Intelligenc*e, 6:41–60.

Webber, Carine; Bergia, Loris; Pesty, Sylvie; Balacheff, Nicolas. The Baghera project: a multi-agent architecture for human learning. Workshop - Multia-Agent Architectures for Distributed Learning Environments. Proceedings International Conference on AI and Education, San Antonio, Texas, May, 2001.

Wiederhold, G. Mediators in the Architecture of Future Information Systems. In: *IEEE Computer*, March 1992, pages 38-49.

Wooldridge, M., and Jennings, N. 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review* 10(2): 115–152.

Appendix 9 Web access and services Personalization

A 9.1 Overview

A 9.1.1. Web Personalization

Web Personalization is the technology of dynamically altering the presentation of a web site according to the preferences of the user. Web Personalization is employed today at hundreds of websites across the Internet, including many of the most trafficked. It can take many forms, and typically are used by web sites to filter or recommend the content and navigation choices displayed to each user, such as custom headers that greet the user by name, custom advertisements that target the user specifically, custom content that solidifies the user's affinity with the site and custom recommendations that encourage the user to purchase. The average Internet-surfer will encounter personalization at retail sites, at portals, and sometimes at news providers.

E-commerce web sites are racing to create dynamic, personalized experiences that accurately target individual customer needs and build user loyalty. In a world where the competition is always just a click away, online vendors search for ways to solidify customer habits.

The Web Personalization market has bifurcated into two different families of products: rules-based engines and collaborative filtering engines. The methodologies differ in the way they determine the optimal personalized content, but much of the featureset is similar. Both solutions seek to understand a user's affinities and personalize the web page accordingly. Both detect user preferences through manual means (user states preferences in an online survey) and induction (user reveals preferences through his/her actions and purchases). Both can access information from other databases to assist in the personalization process.

- **Rules-based** personalization engines use business logic embedded in conditional (if/then) statements to create content display. Under rules-based personalization, a user's known preferences fulfill certain criteria, and corresponding content is served accordingly. A manager or system administrator typically uses a visual interface to input if/then criterion, specifying each condition and the content which should be recommended in response. The primary benefit of this approach lies in its ability to directly link organizational strategy or policy to customer interactions.
- **Collaborative filtering** engines, very different from their rules-based counterparts, use the recorded preferences of all users in order to algorithmically find content that is likely to appeal to the user. Web site users are grouped by their shared preferences, and it is through use of the preferences database that the engine makes predictions. The engine locates people with preferences similar to the user's, and recommends items that those people liked. The power of this approach is its ability to constantly respond to changing user preferences and growing content catalogs without requiring the creation of new personalization rules. Collaborative filtering also allows sites to target recommendations to smaller groups than might otherwise be feasible with rules-based engines.

No single Web Personalization architecture is optimal for all Web sites. Recognizing this, collaborative filtering and rules-based personalization vendors have established partnerships offering varying levels of technical integration between the two systems.

Web personalization solutions are packaged applications, bundled with services such as consulting and training. Like other significant enterprise software applications, web personalization implementations can be time-consuming, complex, and very service-intensive. Personalization solutions come in many forms: clickstream-centric or database-driven, interface-generating or engine-based, off-the-shelf or custom.

A 9.1.2. Personalization of Web Services

Personalization of Web Services is marketing oriented, being the goal to achive customer loyalty. In the educational environment creating customer relations is not the goal, but efficient learning and tutoring, which means that the platform is useful for learning and companies will be determined to buy it. Concepts and techniques described for web services can be applied to an educational environment, since the way personalization is defined is quite similar: "*Personalization involves the process of gathering user-information during interaction with the user (from a history of previous sessions or through interaction in real time), which is then used to deliver appropiate content and services, tailor-made to the user's needs. The aim is to improve the user's experience of a service and better serve the customer by anticipating needs, building a*

relationship that encourages the customer to return for subsequent purchases" [Personalization Consortium].

Personalised service is part of user's standard definition of good service. As users become more proficient in their use of the web and are exposed to a wider range of experience, they become more demanding. Personalized services currently offered by web sites are:

- Information provision, including news and purchasing.
- Selectivity of the content, such as the one provided by *MyYahoo.*
- Purchasing assistance.
- Recommendations.

Most services provide Customisation (user can configure an interface and create a profile manually) instead of Personalization (the website monitors, analyses and reacts to behaviour), and their personalized service is not based on individual user behavior or use input, but contents are tailored for a predefined audience.

Some of the features that appear in Web Personalized Services are:

- Customization of layaout and selection of contents from a choice of modules, the arrangement of contents on screen and the frequency of updating such contents.
- Browsing customization
- Personalization guide to match user tastes by learning the preferences over time
- Alerts on demand (daily email or email reminders), which give users more control over the service
- User's rating of services and products
- Criteria for selecting a product to give specific information afterwards, and allowing comparisons with other products
- Saving information obtained while accesing the service
- List of favourite items
- Group profiles for organizations
- Suggestions based on previous purchases, ratings and by comparing with other customers
- Mailing lists
- Selection and filtering information to provide the users with what they need most:
    - Group favourite, most used resources and links
    - Up-to-date view of recommended resources
    - Personalised information dissemination to users by alerting when new information is available corresponding to a paricular profile

There exist several technologies to gather information to provide personalized services.

- **Fill-in Profile**: user fills in forms to control the type of content provided and the look and feel of the interface (generic information, choice of specific content, general preferences for interaction).
- **Click-stream Analysis / Web Usage Mining Systems**: technique of collecting data about user movements on a website (route followed, possition on the screen of links clicked, time spent, connections between links visited and consequences).
- **Collaborative Filtering**: compares a user's tastes with those other users in order to build a picture of like-minded people.
- **Cookies**: data packets sent by the web site and stored on the browser side that can be used for tracing users.

Resource discovery and learning services have to deal with the following challenges:

- An **information architecture framework** based on user, content and business needs. The business rules determine which content to display and how to present it, by matching attributes of

the content with attributes captured in the user profile. A vocabulary layer for both content and user profiles can be used, which determine the acceptable attributes and their possible values. Vocabularies for describing the content are well studied. For describing users profiles, see the IMS Learner Information Packaging specification.

- Establishing **user's needs** to be able to apply rules and techniques to decide what content might be appropriate. In a marketing environment customers do not need to be individually defined and it is enough to determine target groups and what contents are appropriate for those groups.
- **Usability**: do not use personalization as a excuse for poor design. Besides, sometimes personalization is not used because it may be more useful to let users use their natural intelligence, users do not need complex customizations or lack the confidence to experiment, and other times users use tools in unexpected ways.
- **User privacy and protection of information** supplied by users is an issue of importance since intensive collection and use of personal information is required for personalization (encryption and privacy statement). P3P, which is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies.
- Personalization has to help to recreate the human element and offered a personal touch, and also to support communications among users (**building relationships and communities**). The principle is that of share-and-connect.
- **Measuring success** and the effectiveness of a personalization web service by defining metrics and feedback techniques.

Some more issues about Web Personalization are [JACK AARONSON CONSULTING]:

- Personalization is not an add-on; it is not an afterthough
- The exit point is too late to retain a customer
- Never set expectations too high for the services offered
- One product should do the whole business.

A 9.1.3. Rules of Web Personalization

The following rules gather the mandates behind the design of useful personalization systems.

- **Learn from Every Move:** much must be inferred from the user's small actions. Collect everything from which a lesson could be learned, and try to learn that lesson. Study and mine clickstream data, and learn more from it than a mere row in a database is likely to tell you.

  Was the user satisfied? A satisfaction metric can drive some excellent personalized behavior: a satisfied user is best shown the things that worked last time, while an unsatisfied user should be given the opportunity to radically change the site's approach.
- **Don't Place Resistance in Front of Personalization:** do not if at all possible ask that the user register, fill surveys, pay money, or take any action which is costly. Too many potential participants will be deterred from the system. Instead, attempt to personalize from the moment the customer enters the site; use what little you know to draw them in step by step.
- **Don't Neglect any Source of Information:** filter through the noise and erraticism of user behavior in order to extract those trends that have predictive meaning (data mining upon data warehouse, interpretations upon the clickstream data, stated preferences of the user).
- **Allow the User to Prove you Wrong:** users can change their preferences, a lost user can indicate 'preferences' he/she does not hold, multiple users may share the same computer, and algorithms may incorrectly judge the true preference which motivated user action.

  An *impersonal page*, never more than a click away, is recommended to which users could revert should personalization fail them. This page should be absolutely free of personalized aspects (like most home pages today) and offer easy navigation and search paths. No matter how advanced your personalization scheme, it is always best to let the user get what they want, and to tell you that you didn't provide it.
- **Get all the Information you can, and do it as Painlessly as Possible:** *friction vs. accuracy*. One method is to gather information gradually, allowing the user to explain themselves through their

actions. It is possible to design a site so as to better gather information in this manner, by aligning clicks with meaning, and by relying on clicks (rather than scrolls, which because the take place client-side are 'silent') to access new categories of information. Include attractive 'directory pages' which allow the user, by conveniently navigating to his/her chosen topics, to also indicate a series of preferences and interests. It is easy to overstate the usefulness of this technique; suffice it to say that of the three primary ways in which users find information (click, scroll, search) clicks are the best for preference-comprehension purposes because they record partiality for specific categories.

- **Emphasize Privacy:** customers have two fears: that you will know them too well, and that you will tell others about them. Allow users at any point to erase the variables and methods by which you determine their personalized content, reassure that their categorized preferences are safe with you, that you will not sell or share them and protect your personalization databases from hacking, both external and (more common and more dangerous) internal.
- **Sell the User on the Value of your Personalization:** Always let a user know that what they see is personalized. Encourage the user to take value in the understanding you have regarding their preferences. He/she may wish to aid it, adding value to your service.
- **Observe what Users Dislike in Addition to what they Like:** an oft-ignored special feature on your home page might in the future be dropped, to make way for another sort of attraction, one which had better chance of success.
- **Make it Easy for the User to Tell you what they Like and Dislike:** there are two popular myths which falsely guide the construction of many personalized sites: 1) every user wants to explain themselves to the personalization algorithm at the same time (usually the first time they log in) and 2) every user has the same tolerance for preference-revealing questionnaires and activities.
    - make it easy for users to tell you what they want to say, whenever they want to say it
    - instant feedback features should be embedded into the website (more specific than simple surveys and more quickly than a poll-form)
    - a "go-away button" and a little "x" hidden in the corner for a one-click personalization-correction mechanism
    - an extended interview-in-pieces, that a user could fill out whenever fancy and that be readily available
- **Never Leave the User Waiting for a Personalized Response:** have a canned response ready, and offer it should the personalization process take more than the expected period. This method also provides a good fail-safe for the personalization system. Should it fail, and as a more complex entity it is liable to fail more often than the site itself, it should drop out of the process and allow the website to continue in dis-personalized manner until the error is corrected.

A 9.1.4. Web Access and Service Personalization in Mobile Devices

[Billsus, Brunk, Evans, Gladis and Pazzani, 2002]

Web Access and Service Personalization is the key point that will make some web sites succeed to the detriment of others. However, when dealing with mobile devices, there are some other features that have to be provided by personalization techniques, too.

While browsing and searching can be aceptable methods for locating information on the wired web, those operations become inefficient in the wireless setting. Small screens, slower connections, high latency and limited input capabilites present new challenges. Agents that select information for the user are a convenience when displaying information on a 19-inch desktop monitor accessed over a broadband connection, but they are essential on a handheld wireless device. In this context, adaptive personalization technology is used to make all the information available to the mobile user. The goal of adaptive personalization here is to incresase the usage and acceptance of mobile access through content that is easily accesible and personally relevant.

Many wireless data services cope with these constraints by reducing the amount of information available to users. Many wireless carriers offer a limited degree of personalization, where users customize their menus by visiting a web site and selecting the order in which items appear or indicating topics of particular interest, but only 5% of users makes use of customization. Several factors contribute to the limited usage of such approaches:

- It is fairly complicated to create an account on the web and asóciate that account with a wireless device, limiting the audience to the technically sophisticated
- The categories tend to be very coarse-grained, allowing users to select from general categories rather than specific ones.
- Topic profiles requiere maintenance to be helpful, but few users are willing to continually maintain their customized profile on a regular web site for an optimal wireless web experience
- Such approaches may prevent wireless users from gaining access to information outside their profile without going back to a wiered web site and adjusting the settings

This clearly is a conflict with the goal of being able to access any information at any time from any location.

Explicit customization approaches such as web-based questionnaires place the burden of personalization to the user. A better approach is adaptive personalization, an automated approach that uses artificial intelligence and statistical techniques to construct a model of each user's interests. It is important that adaptive systems learn from a few examples and adapt quickly to changing user interests. To be truly useful in a mobile context, user's interests must be inferred implicitly from actions and not obtained exclusively from explicit content ratings provided by the user. Obtaining explicit feedback about user interests requires a user interface consuming screen real estate and additional data to be transmited. And adaptive interfaces should not limit the user's choices by filtering information, but rather order it, so that the most relevant information is at the top of each screen.

Requierements for adaptive personalization to meet the tough test of user acceptance are:

- **Provide a good initial experience and learn quickly for new users.** The first use should provide an acceptable, non-personalized experience, but the benefits of adaptive personalization must be apparent within the first few uses if users are to return, and this transition must be smooth.
- **Adapt quickly to changing needs.** Users change their habits due to external events.
- **Avoid tunnel vision.** Personalization should not get in the way of finding important novel information.
- **Do not require hand tagging on content with category labels.** This extra work is not feasible with thousands of new items being added daily.
- **Avoid brittleness.** A single action, such as selecting something accidentally or skipping over an article on a topic should not have a drastic und unrecoverable effect on the presentation.
- **Support multiple modes of information access**, such as content related to the current information.
- **Respect individual's privacy.** Although personalization is designed to help users by minimizing the amount of work they must do to access relevant information, different users have different trade-offs for convenience and privacy. Users can opt to have the server save identifying information to avoid repeatedly typing the same address when using features such as forwarding items by e-mail. However, users may remain anonymous while providing the benefits of personalization, turn off all personalization features, or delete their data from the server.

A combination of similarity-based methods, Bayesian methods and collaborative methods meets the above requirements by achieving the right balance of learning and adapting quickly to changing interests while avoiding brittleness. Response time and scalability are also key concerns. Because the mobile devices have limited processing power and memory and slow network access, the personalization technology resides on a server that interacts with a browser on the devices. With inverted indices, caching of similarity calculations, and distributing computation across multiple servers, it is possible to meet the performance needs of personalized content delivery. An added benefit of the inverted index is that it does support full text search for content, and the similarity cache facilitates finding related information.

Adaptive personalization in mobile device context provides benefits any time there are more options than easily fit on the screen, and each individual explores only a subset of the available options. Many mobile applications are improved by the use of this technology. Some applications with Adaptive Personalization are the following:

- **Classified Ads.** To provide the most personally relevant information to the user on the initial screens and to save users from filling out many forms to specify their interests precisely.
- **News.** Items related to articles read in the past are displayed.

- **Restaurants and Entertainment Listings.** Adaptive personalization automatically learns customers' tastes in restaurants, nightclubs, etc. and can get positive feedback by actions such as calling the restaurant or looking for directions.
- **Wireless Portals.** To log which sites the user accesses from a hierarchical menu and make the most frequently accessed site the default for that user.
- **Voice Portals.** Anyone who has interacted with an airline or bank automated phone system can understand the benefits of presenting information in an order that is relevant to the user. In a voice interface, adaptive personalization can change both the order in which categorizes are listed, as well as the order in which items within categories are spoken. By adaptively learning preferences, users can save time while accessing personally relevant information.

Concentrating on mobile access has forced to think carefully about what capabilities and interfaces elements are essential. However, adaptive personalization technology is not tied to any delivery channel and provides benefits for Internet access from a desktop computer as well. For example, it's not uncommon to have a few hundred cars, apartments, or jobs match in an online classified ad system. Adaptive methods can be used to prioritize which ads are most likely to be of interest to each shopper and display them on the first screen. In an online news site, adaptive personalization puts the news stories that are most likely to be of interest to the reader above the fold (i.e., visible at the top of the screen without scrolling). Adaptive info is currently working with a large US media company to roll out an adaptive interface for a group of wired and wireless news web sites. The list of applications for adaptive interfaces on mobile devices is by no means comprehensive. Other applications of adaptive personalization include prioritizing mail messages, finding applications to download onto 3G phones, keeping track of sales leads, shopping from product catalogs, searching through real estate listings, and browsing online dating services. Furthermore, the server solution can easily be extended to handle new devices as they become available in the rapidly evolving world of mobile communications.

A 9.2 Solutions

The main technologies currently used for web personalization are the profiling and collaborative filtering.

Profiling, which aggregates data culled from multiple Web sites, allows an online session to be tailored before a potential customer even identifies him/herself or places an order. Without the business knowing the identity of the site visitor, the business's profiling application can know about this visitor's interests and needs. Based on ID information automatically supplied to the business when the visitor arrives on a site, the site is configured to appeal to the buying habits of that customer. This previously obtained information from other sites is provided without access to the customer's identity.

Collaborative filtering (or social filtering) allows a business to utilize the e-commerce experiences of a customer or other customers in shaping its electronic responses to each individual customer. Through the use of an algorithm or equation and sophisticated data analysis that may include neural networks, the history of past interactions produces for the business a projection of a customer's future buying behavior. This enables the business to offer a unique product or service for each customer that is more likely to be attractive.

However, other solutions taken from the field of Adaptive Interfaces, can also be taken into account [Langley, 1999]:

- **Information filtering:** content-based filtering
- **Generative systems:** create new knowledge structures to satisfy the user's goals
- **Apprentices**, where each decision provides data for learning a user model

A 9.3 Assessment

A 9.3.1 Conclusions for Alfanet

Currently, most web services provide customization instead of personalization, with contents tailored for a predefined audience. In a marketing environment customers do not need to be individually defined and it is enough to determine target groups and what contents are appropriate for those groups. However, the system we are to develop in aLFanet should adapt individually to each user, by real personalization.

The rules for designing useful web personalization systems given in this appendix should be taken into account when designing the aLFanet system, specially the ones that refer to not annoying the user while collecting the data and to give the user the last word when using the system. Privacy and information protection, which is an issue of great importance in web access and service personalization, has to be also a key point in aLFanet system. Moreover, the principle of share-and-connect is also applicable.

Mobile devices demand the use of an effective adaptive personalization due to their intrinsic constraints. Requirements fulfilled by adaptive personalization in mobile devices should also be considered in the aLFanet system.

Having in mind the rules for good designing of web personalization systems and the requirements for adaptive personalization, the aLFanet system should, at least, take into account the following topics:

- Provide a good initial non-personalized experience and learn quickly, so users check the benefits of adaptive personalization soon.
- Adapt quickly to changing needs of the users
- Collect and study all data that can be used to individually adapt the system to each user
- Do not require hand tagging on content with category labels
- Do not place resistance in front of personalization by forcing actions which are costly (in effort) to users
- Allow the user to prove wrong system decisions
- Be able to recover from user's wrong actions
- Never leave the user waiting for a personalized response
- Respect individual's privacy

A 9.4 References

APPIAN CORPORATION: http://www.appiancorp.com/report/features/

Billsus, D., Brunk, C.A. Evans, C., Gladis, B. and Pazzani, M. (2002) *Adaptive Interfaces for Ubiquitous Web Access*. CACM 45(5): 34-38

Bonnet, M. (2001) *Personalization of Web Services: Opportunities and Challenges*: http://www.ariadne.ac.uk/issue28/personalization/

JACK AARONSON CONSULTING: http://ww.JackAaronson.com

Langley, P. (1999). User modeling in adaptive interfaces. *Proceedings of the Seventh International Conference on User Modeling* (pp. 357-370). Banff, Alberta: Springer.

PERSONALIZATION CONSORTIUM: http://www.personalization.org

Appendix 10 EML – Learning standard

A 10.1 Overview

EML has been developed by the Open University of the Netherlands for use in e-learning. The version 1.0 information model and XML binding has been released in December 2000. Version 1.1 is in beta. EML has been selected as the base for the IMS Learning Design specification, where it is integrated with IMS Content Packaging and IMS Simple Sequencing. It has been used and deployed in several runtime systems and a large number of courses has been modelled and delivered with it.

Current learning technology specifications allow only for some simple ordering and sequencing of resources used in e-learning (e.g. SCORM, IMS Content Packaging, IMS Simple Sequencing). EML adds to this the ability to integrate learning designs ('instructional designs') to enable more advanced e-learning applications, e.g. to model competency based education, portfolio's, collaborative learning.

EML is a semantic specification, based on a pedagogical meta-model, which describes the structure and processes in a 'units of learning'. It aggregates learning objects with learning objectives, prerequisites, learning activities, teaching activities and learning services in a workflow (or better 'learning flow'), which itself is modelled according to a certain learning design. The learning design is a concrete instance of a pedagogical model, which at its turn is an instance of the pedagogical meta-model. The meta-model doesn't force users to use a certain pedagogical model, but allows to create and describe their own models in an expressive way. The meta-model is derived from an analysis of current existing models, based on (social) constructivist's approaches, behaviourist's approaches or cognitive approaches.

EML provides a semantic information model and several bindings (in SGML and XML).

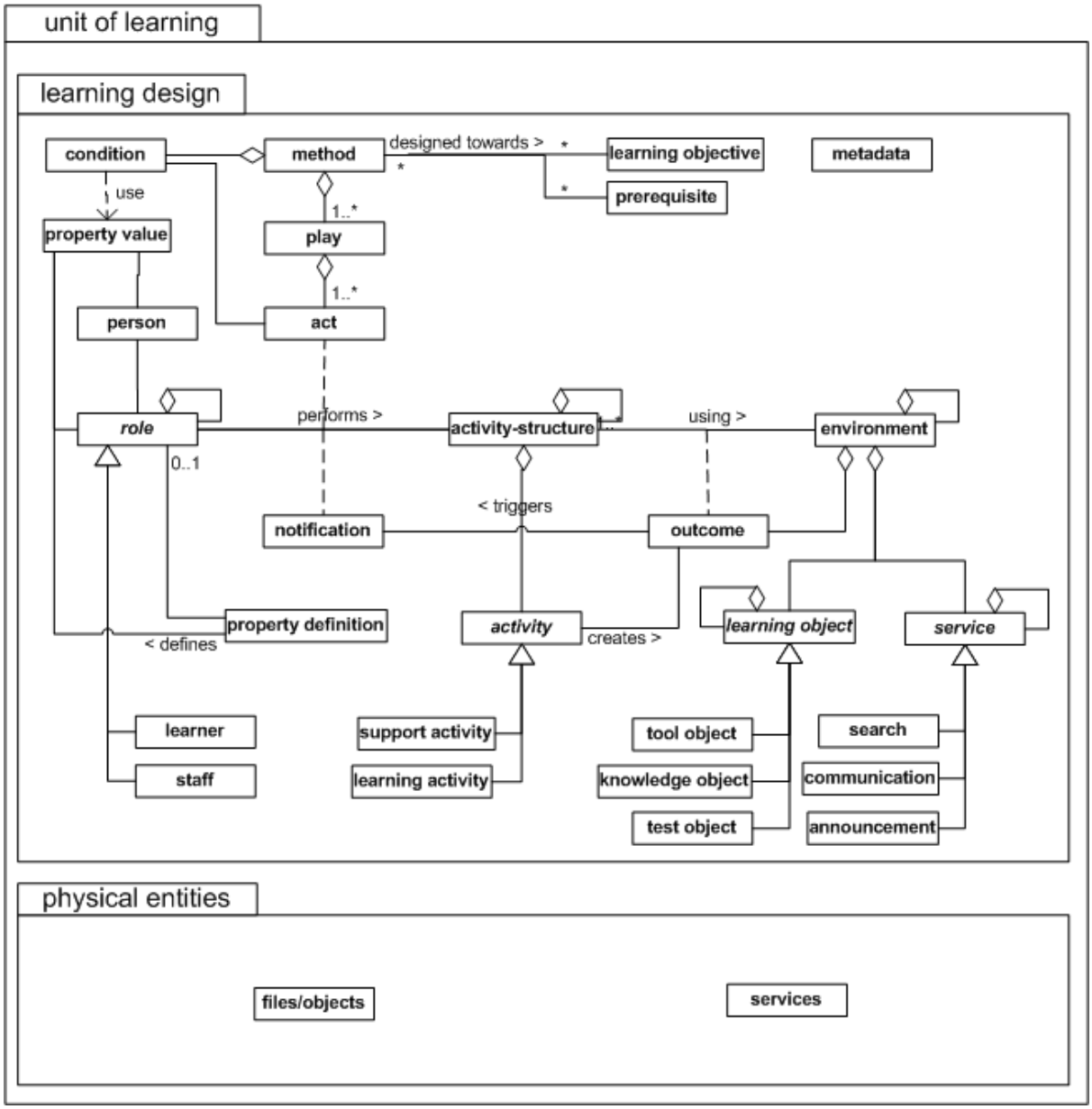
A 10.2 Information Model

The major requirements for the development of the EML information model were:

1. EML must describe units of learning in a formal way, so that automatic processing is possible (*formalisation*).
2. EML must be able to describe units of learning that are based on different theories and models of learning and instruction (*pedagogical flexibility*).
3. EML must explicitly express the semantic meaning of the different learning objects within the context of a unit of learning. It must provide for a semantic structure of the content or functionality of the typed learning objects within a unit of learning, alongside a reference possibility (*explicitly typed learning objects*).
4. EML must be able to fully describe a unit of learning, including all the typed learning objects, the relationship between the objects and the activities and the workflow of all students and staff members with the learning objects (*completeness)*. And regardless of whether these aspects are represented digital or non-digital.
5. EML must describe the units of learning so that repeated execution is possible (*reproducibility*).
6. EML must be able to describe personalization aspects within units of learning, so that the content and activities within units of learning can be adapted based on the preferences, prior knowledge, educational needs and situational circumstances of users. In addition, control must be able to be given, as desired, to the student, a staff member, the computer or the designer (*personalization*).
7. EML describes content resources in a medium neutral way, so that it can be used in different publication formats, like the web, paper, e-books, mobile, etc. and also in different settings like distance teaching, online learning, blended learning, hybrid learning, … (*medium and setting neutrality*).
8. EML files must be interoperable between different learning (content) management systems (*interoperability and sustainability).*
9. The notational system must fit in with available standards and specifications (*compatibility).*
10. EML must make it possible to identify, isolate, de-contextualize and exchange useful learning objects, and to re-use these in other contexts (*reusability*).

11. EML must make it possible to produce, mutate, preserve, distribute and archive units of learning and all of its containing learning objects (*life cycle).*

EML has currently two versions (1.0 and 1.1 beta). There is a XSLT converter available automatically converting 1.0 to 1.1. The information model of version 1.1 is depicted in the diagram below.

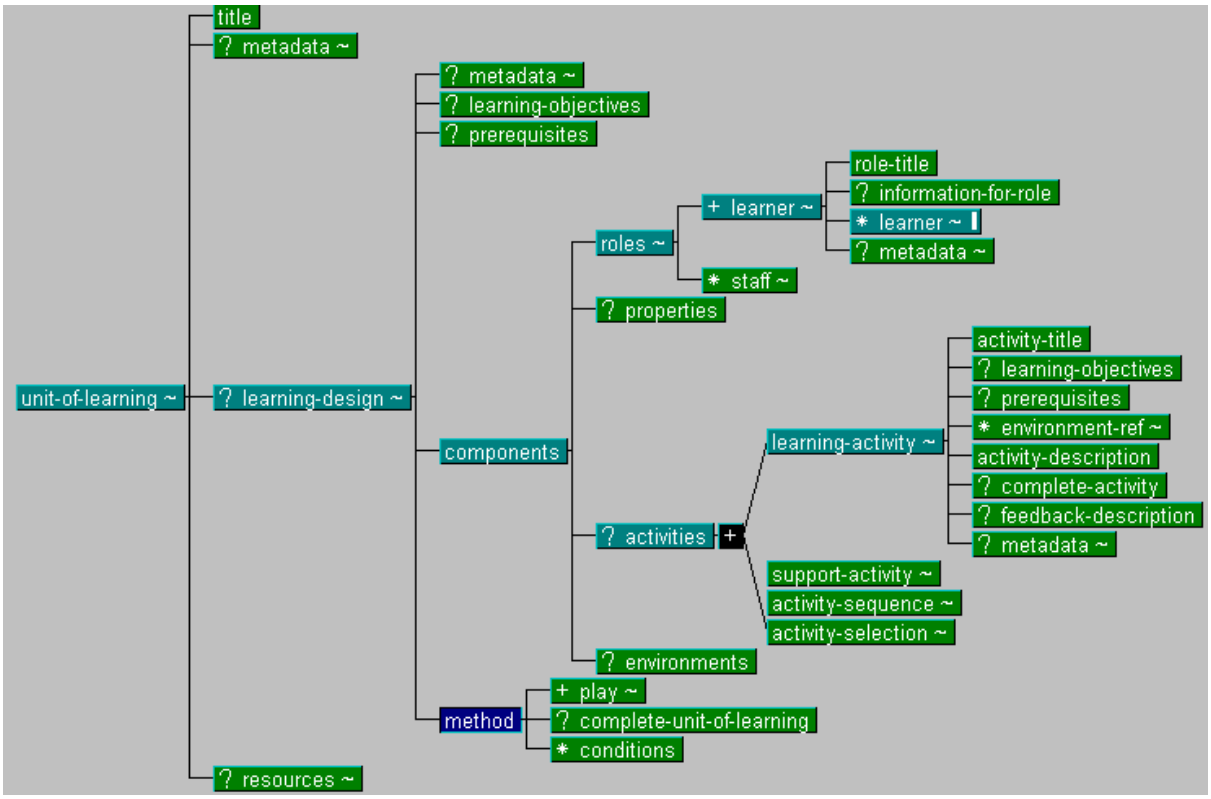


The Information model was developed by analysing all kinds of pedagogical models. In its most general form, it can describe a *role* (student or staff) who perform(s) a series of *activities* in an environment that consists of learning objects and learning services.

The information model is implemented in the EML 1.0 and 1.1 beta binding (DTD).

A 10.3 EML DTD

The EML Binding (DTD), reference manual, and articles about EML can be downloaded form: http://eml.ou.nl. Also specific sample courses are accessible at this website (Jazz course). The figure below gives the tree structure for the new version.



A 10.4 Sample of EML document instance

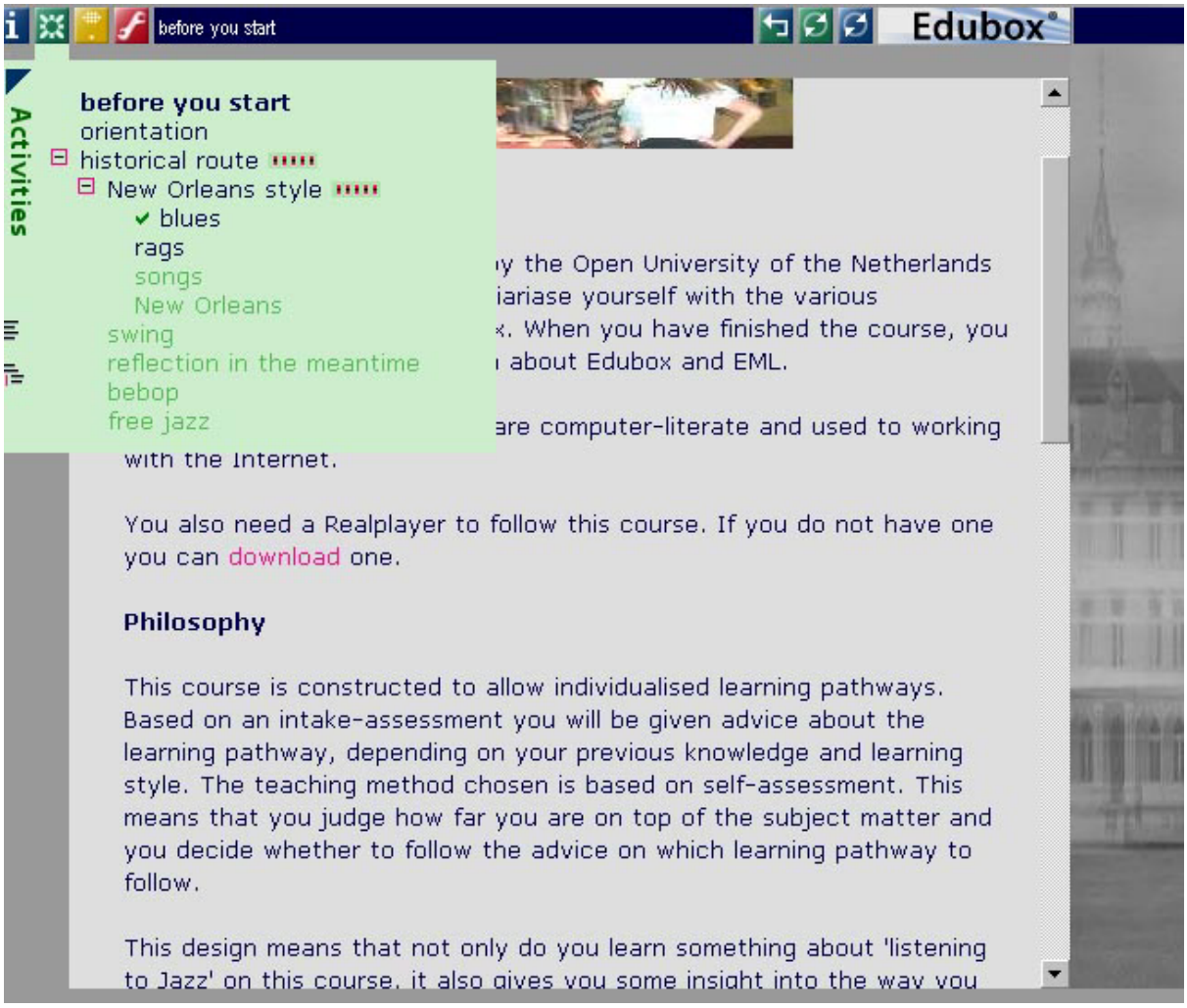
```
1. <unit-of-learning>
2.   <title>Introduction into EML</title>
3.   <metadata><<LOM schema included>></metadata>
4.   <learning-design>
5.     <components>
6.       <roles>
7.         <learner identifier="student"><title>student</title></learner>
8.         <staff identifier="tutor"><title>begeleider</title></staff>
9.       </roles>
10.      <properties/>
11. <activities>
12.   <learning-activity identifier="id-F8" isvisible="true">
13.     <title>introduction</title>
14.     <activity-description><item identifierref="resource-act1" /></activity-description>
15.     <complete-activity><user-choice /></complete-activity>
16.   </learning-activity>
17.   <learning-activity identifier="id-87" isvisible="true">
18.     <title>about elements</title>
19.          <learning-objectives><item identifierref="resource-lo1"></item>
20.          </learning-objectives>
21.     <activity-description><item identifierref="resource-act2" /></activity-description>
22.     <complete-activity><user-choice /></complete-activity>
23.   </learning-activity>
24.   <activity-selection identifier="AS-intro" number-to-select="1">
25.     <title>Overview</title>
26.     <information><item identifierref="resource-info1" /></information>
```

```
27.      <environment-ref ref="id-69" />
28.      <learning-activity-ref ref="id-F8" />
29.      <learning-activity-ref ref="id-87" />
30.  </activity-selection>
31.  <environments>
32.      <environment identifier="id-69">
33.       <title>In general</title>
34.       <knowledge-object identifier="id-B66">
35.            <title>de knoppen in Edubox</title>
36.            <item identifierref="res-1.3.1.1.2.1" />
37.       </knowledge-object>
38.    </environment>
39.  </environments>
40.  </components>
41.    <method>
42.      <play>
43.       <act>
44.        <role-part><role-ref ref="student"/><activity-selection-ref ref="AS-intro" /></role-part>
45.       </act>
46.      </play>
47.      <conditions/>
48.          </method>
49.  </learning-design>
50.  <resources>
51.  <resource identifier="resource-lo1" type="emlcontent" href="resource-121111.xml">
52.  <file href="resource-121111.xml" /></resource>
53.  <<ETC>>
54.  </resources>
55.</unit-of-learning>
```

Content is any format, however more advanced content (e.g. using the properties of the unit-of-learning file to personalize content) uses XHTML with a specific set of global elements mixed into this using namespaces.

A 10.5 Sample of course material

A screen dump from the Jazz course is shown below. Note that this is a screen dump of a course presented using Edubox, a runtime system which interprets course material described using EML. The screen dump shows the user having clicked on the Activities button and the corresponding drop-down menu.



There are many instances of EML, some of these are:

- Four fully online 200 hour courses in Business and Public Administration for OUNL
- Courses in Psychology, Geography, Law, Geographical Information Systems, Economics, etc for OUNL
- 12 dual mode courses (the whole curriculum) for the Hotel Management School, institute for higher vocational education
- UNISA (Open University of South-Africa) has developed several courses in EML
- Courses for Medienzentrum Innsbruck in Austria
- Digital University of the Netherlands (consortium of 9 universities) develops law courses in EML.
- Training modules for training departments of different companies

There have been build several runtime systems during the development of EML (Elon system, Edubox prototypes, etc.). Perot Systems is currently building a new scalable Java based runtime for EML, including hosting services (of which some service levels are free).

Some examples of instance are available in the article: Modelling units of study from pedagogical perspective: The pedagogical meta-model behind EML. This article can be downloaded form http://eml.ou.nl.

A demonstration course is available, *The Jazz Course.* It is a demonstrator of EML and the reference system (Edubox), which plays EML files. This course can be made publicly available. Accounts for this course are - on request - also available.

A 10.6 Assessment

A 10.6.1 Situation and prognosis

The interest in and use of Educational Modelling Languages in general, and in particular in the OU EML, is growing. There is a clear interest in languages capable in modeling both structure and process in 'units of learning' being able to express a variety of pedagogical models:

- EML has been selected as the base for the IMS Learning Design specification, where it is integrated with IMS Content Packaging and IMS Simple Sequencing. In the second part of 2002 the first public draft will become available.
- In a recent "Survey of Educational Modelling Languages" (Rawlings et al, October 2002) of the CEN-ISSS WSLT Learning Technologies Workshop six European Educational Modelling Languages were compared. As a basis for the comparison the following definition was used:

  An EML is a semantic information model and binding, describing the content and process within a 'unit of learning' from a pedagogical perspective in order to support reuse and interoperability

  The outcome of this comparison made clear that the scope of the EMLs differs very much. There appear to be two groups. One group restricts themselves to modelling learning content and structure. They seem to be 'ignorant' in expressing pedagogical models. They can be used within any pedagogical model, but they cannot express executable knowledge about the model. The second group consists of PALO and EML (EML-OU). They live up to the definition above. From the two, EML-OU fully lives up to the definition. It is capable to model both structure and process and it allows for the *expression of the designer's own pedagogical models*.
- Spring 2002, the 'Valkenburger group' has been established. Organisations (universities and companies) of four continents with a shared interest in the use of EML.

A 10.6.2 Conclusions for ALFANET

The objective of ALFANET to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods requires the capability to model both structure and process. EML (including Edubox) offers this capability and equally important in depth knowledge of EML is available and directly accessible, so a quick start can easily be made. In other words EML is a logical candidate for inclusion in ALFANET.

To assure a successful inclusion a number of points has to be taken care of:

- All partners of ALFANET have to acquire the appropriate level of knowledge of EML. This ranges from the basic knowledge of the language, to the design process of 'units of learning', to the technical consequences to be able to fit EML in the proposed multi agent architecture.
- The relation of EML with other standards to be included have to be investigated, in particular in the area of knowledge management and multi agent architectures, and the technical consequences have to documented.

Finally, it is important to notice that Alfanet, given its unique requirements and approach, can by using EML contribute to the further development of standards by validating the current scope and/or proposing extension(s) necessary for a multi-agent, adaptive approach. The required work to achieve this fits in WP3 'Standards'.

Appendix 11 Educational ontologies

A 11.1 Overview

A 11.1.1 Relevance of educational ontologies

An important impulse for the recent activities in the development of ontologies has been given by the initiator of the World Wide Web, Tim Berners-Lee. In an article in Scientific American [Berners-Lee 2001] he promoted the concept of the Semantic Web, a mixture of ontology descriptions and agents that can use the ontologies to perform tasks for the webusers and for websystems. For the authors of this article an ontology is simply stated a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules.

Information on existing general ontology standards and definitions are available in deliverable D31. Here we concentrate on the implementation of ontologies within an educational setting.

In the past decade research efforts have been directed to the problem to unambiguously model educational structure, content and processes. Solutions like EML are products of these endeavours.

However educational processes are now rapidly extending to open environments where materials, actors and processes cannot be modelled completely in advance.

For example in cases where competencies have to be trained in realistic environments, based on constructivist learning principles, more open learning structures are often needed.

This means that support is needed for a higher level of semantic typing of relevant objects in the study processes.

Educational ontologies should facilitate the meta-modelling of the relevant entities in the educational processes to:

- facilitate broader reuse;
- give access tot content that is emerging in the actual study process;
- anticipate on unexpected and unanticipated use;
- adapt to changing learning-needs of the student;
- adapt assignments to changing conditions by defining assignments on a meta-level;
- give access to various data (personal and non-personal) for evaluation purposes to analyse learning activities on a meta-level;
- facilitate automatic processing of separately developed materials;
- facilitate integrated analysis of separately modelled processes.

To fulfil these demands shared educational ontologies can help to create ‘networks of meaning’.

A 11.1.2 Current developments

Methodologies for ontology construction

A distinction has to be made between the ontology itself, which specifies the concepts in a domain (whose existence and relationships are true by definition or convention) and the empirical facts which are not part of the ontology but can be structured by the ontology. Commitment of all partners using the ontology to store and exchange empirical data is important (Holsapple 2002).

Holsapple advocates the need for ontological engineering as a collaborative approach to ontology design and five approaches for ontology design are mentioned in the article:

- inspirational approach

  The individual developer, using his personal imagination, creativity and views on the domain of interest, designs an ontology that will meet the recognized needs. Commitment will generally be

narrow and evidence for the adoption can come from the extent of its use, longevity and the degree to which it leads to more extensive ontologies that adopt the basic features.

- inductive approach

  Adopting this approach an ontology is developed by observing, analysing a specific case. With this approach the commitment can be concluded form the extent to which the ontology will be applied in other (related) contexts.

- deductive approach

  In this approach an ontology is constructed using some general principles and is then adaptively applied to a specific case.

- synthetic approach

  A unified ontology can be synthesised by analysing existing ontologies, integration of their concepts, elimination of sketchier characterizations in favour of more fully developed ones and reconciliation of different terminologies.

- collaborative approach

By choosing a collaborative approach, development is a joint effort reflecting experiences and viewpoints of persons who intentionally cooperate to produce it.

The usefulness of the different approaches is dependant on the problem at hand and the environmental setting.

For building educational ontologies the synthetic approach seems interesting when starting from well known content domains and well defined educational models. The collaborative approach seems also interesting because of the inherent need for cooperation between different disciplines while designing and building educational environments.

Holsapple distinguishes four phases in ontology design:

- preparation phase;
- anchoring phase;
- iterative improvement;
- application.

Ontologies compared to more common ways to share information

In cases where complexity or uncertainty occur semantics should be adopted (Kim 2002).

Compared to standards like XML complexity can be reduced by ontologies and by using to the three complexity reduction principles the need for applying ontologies can be determined:

- bounded reality

  The expression of the meaning of terms in formal languages is complex, but missing meaning can lead to misinterpretation of crucial information. In cases where shared understanding is implicitly present or are informally applied (for example manuals), the uncertainty of meaning can be managed in most cases.

- division of labour

  For humans manipulating content with XML, a distinction is often made, where the DTD is manipulated by professionals whereas the data manipulation is performed by computers with limited manual intervention by humans. The manipulation of XML can then be more efficient than the manipulation of ontologies which require skills beyond the merely clerical. By applying semantics using ontology based systems, the risks of misinterpretation of data however are smaller, especially in the case of automatic data sharing.

- near decomposability

  Within a unit or community information often can be interpreted correctly, so in organisations with near decomposability (In The science of the artificial Simon [Simon 1996] puts forward the proposition that basically all viable systems, be they physical, social, biological, artificial, share the

property of having a *near decomposable* architecture: they are organized into hierarchical layers of parts, parts of parts, parts of parts of parts and so on, in such a way that interactions among elements belonging to the same parts are much more intense than interactions among elements belonging to different parts.) XML can reduce uncertainty by applying bounded rationality and division of labour. Where near decomposability is not the case ontology use increases the likelihood that data can still be shared.

To summarize: XML use can reduce complexity as proven data-sharing platforms are already available, whereas using ontologies will reduce uncertainty.

Assessment of standards

Nilsson (Nilson 2001) attempts to answer the question which standards for encoding learning technology specifications are most useful and representatives of the future of meta-data encoding.

Traditional meta-data approaches take the view of meta-data as being a digital indexing schema to use for cataloguing digital libraries. The semantic web and the ontologies that are an integral part of the semantic web are distinct form these approaches in two ways. They are designed to allow reasoning and inference capabilities to be added to the pure descriptions. Then the semantic web and the ontologies are web-technologies that live on top of the existing web.

Seen as a layered structure XML forms the basis, being the transport syntax and RDF provide the information representation framework and on top of this layer schemas and ontologies provide the logical apparatus necessary for the expression of vocabularies enabling intelligent processing of information.

However there are problems with this layered model. XML/XML schema is a data modelling language whereas RDF models meta-data. The RDF model and the XML model are fundamentally different with respect to functionally, resources and semantics.

First experiences of Nilsson with RDF-based meta-data descriptions in learning technologies to produce an IMS Content Packaging RDF binding, related to version 1.2 of the meta-data specification, lead to some important positive lessons:

- Interoperability with other, separate, standards is greatly increased because RDF allows a single storage model for very different types of data and schemas
- Reuse of existing meta-data standards is greatly simplified.
- Terms that do not have exact equivalents in other meta-data standards, but relate to some existing terms by, for example, being more narrow, more broad etc, can be modelled without converting the underlying meta-data descriptions.
- There exists no standard way to encode and distribute vocabularies. In RDF however vocabularies are a fundamental part of the RDF Schema specification. Not only is there a standard way to list vocabulary items, but their interdependencies can be modelled in a standard way. Efforts such as DAML and OIL provide means to model vocabularies as full-fledged ontologies expressed in RDF.
- While extending the XML binding is certainly possible using XML Schemas, the process easily creates interoperability problems. In RDF, several independent means of meaningful extensions are available, none of which cause interoperability problems.

But also some drawbacks were recognised in this process:

- The underlying standards, notably the RDF Schema specification, but also the Dublin Core Qualifiers RDF binding, are still young, intensively discussed and possibly subject to change. The specifications underlying the XML binding are much more stable, but this is, of course, a temporary problem.
- Tool support for RDF is very immature at this point, and integration of Semantic Web technologies into the current Web is still only starting. XML support can be said to be mature in most respects. However, with the current pace of RDF adoption, tool support is rapidly increasing.
- Designing an RDF binding makes it necessary to revisit many of the assumptions in the underlying information model, which often is designed with an XML binding in mind. As the semantics of XML elements is not explicitly stated, much of the work in designing an RDF binding goes into defining the semantics of the elements.

From a more strategic point of view, the emerging Semantic Web presents exciting new possibilities for uses of learning technology specifications. While XML standards are very good tools for enabling interoperability

by specifying import and export formats for LMS's, they tend to favour large, centrally managed, monolithic systems. By enabling the use of learning technology specifications in Semantic Web technologies, a much wider range of applications are imaginable:

Nilson considers the future for RDF for learning technology specifications to be bright, and the possibilities opened up by RDF and Semantic Web technologies promise to take learning technology project to a new level of applications.

Dhraief [Dhraief 2001] have build an open learning repository system (OLR) and discuss the relation to the LOM-RDF binding. The OLR system uses a database which introduces advantages because of the maintainability of the data instances.

The authors also compare RDF/RDFS to the O-Telos modelling language which is strictly axiomatized and is very suitable for modelling and meta-modelling tasks and they propose a RDF-varian: O-Telos-RDF.

Student models

In a position paper on ontologies and student models, Kay (Kay 1999) stresses the importance of ontologies for reuse and scrutable student models.

The requirement for reuse means in this case that the different teaching systems should be able to make use of the same student model. Scrutability implies that the model can be used by the learner to find out what it models about them and how the modelling process operates.

Barros et al (Barros 2001) propose an integrated ontology for defining collaborative learning experiences and for analysing their processes and results and they conclude that ontology engineering seems to be an effective approach for collaborative learning that needs comprehensive and comprehensible vocabulary for better understanding of the current state of the art and for identifying the directions to go.

Research in CSCL, by combining social with cognitive perspectives, shows the potential to make important strides forward in understanding how we might facilitate learning in real situations. The concept of *activity* plays a central role in the study of Barros which builds upon Activity Theory. Relevant terms in this context are learning goals, group structure, the tools that are available; the roles taking account of the tasks, the goal, the groups and the restrictions of the use of the system; and all of them related with the context and the domain that involves each collaborative learning experience. Other relevant elements are the *community* involved, the *social norms* that govern it; the *division of labour* to be followed; the *tools* to be used for working; the *subject* of the activity; the *object* of the activity; and, finally, the *outcome* produced by the group.

Community support

Educational communities on the web can also benefit from ontologies. Especially interoperability can be improved through the use of common, shared ontologies.

The xTalks project for example profits from the use of ontologies. xTALKS (http://www.ittalks.org/jsp/Controller.jsp) is a portal of information about talks, seminars and colloquia related to Information Technology. It is organized around "domains" which typically represent talk hosting organizations such as universities, research laboratories or professional groups. It is also possible to create domains which are focused on a particular topic or set of topics. For xTALKS, a number of specific ontologies have been developed:

- talk-ont: a basic ontology for talks, seminars and colloquia;
- topic-ont: an ontology for representing topic hierarchies;
- profile-ont: an ontology for representing user profiles;
- classification-ont: an ontology build on topic-ont for describing computer science talks;
- calendar-ont: a simple ontology for representing people's schedules.

This example also shows that resources that are modelled in ontologies can also be processes along with documents, people etc.

A 11.2 Solutions

The number of tools to build ontologies is growing rapidly. At the moment there are no specific implementations to build RDF or DAML models interactively in an intuitive way.

There are two main directions that are followed to create tools with still limited functionality and/or limited adherence to standards:

- starting from existing solutions and integrating new ontology modelling standards;
- creating specific building blocks.

The Protégé system [Noy 2001] (see also http://protege.stanford.edu) is an example which started more than a decennium ago as a tool for knowledge-acquisition and incorporates now the Open Knowledge Base Connectivity (OKBC). The extensible and flexible knowledge model and the open plug-in architecture of Protégé-2000 constitute the basis for developing a suite of conceptual-level editors for Semantic Web languages. Protégé can now be used as a RDFS and OIL editor.

An different approach is chosen in case of the Briefing Associate [Tallis 2002]. The prevalent approach is to create specialized tools that support the association of semantic mark-ups with pre-existing document content These tools provide a GUI that lets authors browse ontologies, find appropriate terms, generate syntactically correct mark-ups, and associate them with portions of a document or, more often, with the document as a whole. This activity remains an extra effort that does not directly reward authors. The Briefing Associate operates through the PowerPoint graphical user interface. The native GUI of Power Point is extended with a toolbar for adding graphics that represent a particular ontology's classes and properties.

Topic Maps (http://www.topicmaps.org/) are often used to describe knowledge structures and associate knowledge structures with resources. The XTM standard (XML Topic Maps) is defined and there are now initiatives to clarify the relationships between the different knowledge representation such as Topic Maps and RFD (http://www.y12.doe.gov/sgml/sc34/document/0323.htm).

A lot of work is devoted to the automatic or semi-automatic building of ontologies form existing documents and web sites.

Desmontils (Desmontils, 2001) for example uses ontologies and natural language processing to build a structured index of websites. This structure is given in advance by a terminology oriented ontology of a domain which is chosen a priori according to the content of the Web site. By using improved natural language techniques in combination with a thesaurus candidate concepts can be associated with each term. It makes it possible to reason at a conceptual level. An elaborate list of related work is part of this article.

There are a lot of activities in the direction of creating new building blocks which is impressively underlined by a list of tools on the DAML website (http://www.daml.org/tools/). The tools for DAML are divided in categories like: browser, parser, editor, Graph Visualization, import/export, Inference Engine, database query.

A 11.3 Assessment

A 11.3.1 Situation and prognosis

The use of ontologies within an educational context can be considered to be a strongly developing field of interest. There are now a number of standards that could be adopted to build educational ontologies. RDF and DAML+OIL are the most promising. A lot of work however has still to be done before non-specialist in the ontology field could model educational materials and educational processes to produce state-of-the-art, generally usable, ontologies.

More advanced methods and methodologies, which aren't available at the moment, are also required to model specific educational ontologies.

Also on the conceptual level a lot of work needs to be done before educational ontologies will become a central perspective when making educational models.

Tools are becoming available now at high speed, but the functionalities are still limited.

But the central question remains: are ontologies worth the extra investments, both initial and structural?

A 11.3.2 Conclusions for ALFANET

Educational ontologies are a very interesting field to study within the scope of the Alfanet project.

To build the intelligent tutoring systems of the future, knowledge at a meta-level of the different actors and resources has to be available and intelligent agents are necessary to interpret meta-information by using various inference mechanisms.

So:

- incorporate ontologies to describe materials, people and processes at a meta-level;
- differentiate when considering the incorporation of educational ontologies; do not consider ontologies as a central design principle, but adopt ontologies in cases where the advantages are obvious;
- develop software agents with various inference mechanisms that use meta-information to help people and systems make use of all the relevant available information, both inside and outside the educational system.

A critical point has to be taken into account as well: an open learning environment is a prerequisite for the meaningful use of ontologies.

A 11.3.3 References

Barners-Lee, T. , Hendler, J. and Lassilla O. (2001) The Semantic web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, *Scientific American, May 2001,* http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Barros, B.M. Mizoguchi, R. and Verdejo, M.F. (2001) A platform for collaboration analysis in CSCL. An ontological approach, *submitted to AIED01, St. Antonio, USA, May 19-23, 2001*. http://www.ei.sanken.osaka-u.ac.jp/pub/miz/BbRmFv.htm

Desmontils E., and Jacquin, C.(?), Indexing a website with a Terminology Oriented Ontology, *Semantic Web Working Symposium (SWWS, 2001, Stanford University, California, USA*, http://www.semanticweb.org/SWWS/program/full/paper5.pdf

Dhraief, H. Nejdl, W., Wolf, D. and Wolpers, M. (2001) Open Learning Repositories and Metadata Modeling, *Semantic Web Working Symposium (SWWS, 2001, Stanford University, California, USA*, http://www.semanticweb.org/SWWS/program/full/paper38.pdf

Holsapple, C.W. and Joshi, K.D. (2002) A collaborative approach to ontology design, *Communications of the ACM*, 45,2 (pp 42-47)

Kay J. (1999) Ontologies for reusable and scrutable student models, *AI-ED 99 Workshop on Ontologies for Intelligent Educational Systems* http://www.ei.sanken.osaka-u.ac.jp/aied99/a-papers/J-Kay.pdf

Kim H. (2002) Predictions how ontologies for the semantic web will evolve, *Communications of the ACM*, 45,2 (pp 48-54)

Nilsson, M. (2001) The semantic web: how RDF will change learning technology standards, http://www.cetis.ac.uk/content/20010927172953

Noy, N., Sintek, M., Decker, S., Crubezy, M. Fergerson, R. W. and Musen, M. A. (2001) Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2):60-71, 2001.

Simon H. A. (1981), The sciences of the artificial, 3 rd edition, *Cambridge MA, MIT Press*.

Tallis, M., Goldman, N. and Balzer, R. (2002) The Briefing Associate: Easing Authors into the Semantic Web, *Computer Society, Volume 17, Number 1, January/February 2002 http://mr.teknowledge.com/daml/publications/IEEE-IS-02-article.pdf*.

Appendix 12 XML and related technologies

This appendix presents the market tendency in the XML area. Besides, it provides a general overview of the main concepts related with XML. First, we presents the solutions and tendencies in the XML field, and then we relate them with ALFANET project.

A 12.1 XML and related concepts

A 12.1.1 Standard Generalised Markup Language (SGML)

SGML is a Standard Generalized Markup Language defined in ISO standard 8879:1986. SGML takes the concept of descriptive markup beyond the level of other markup languages [1]. It is a standard for how to specify a document markup language or tag set. Such specification is itself a document type definition. SGML is not in itself a document language, but a description of how to specify one.

SGML is based on the idea that documents have structural and other semantic elements that can be described without reference to how such elements should be displayed. The actual display of such a document may vary, depending on the output medium and style preferences. Some advantages of documents based on SGML are:

- They can be created by thinking in terms of document structure rather than appearance characteristics (which may change over time).
- They will be more portable because an SGML compiler can interpret any document by reference to its document tag definition (DTD).
- Documents originally intended for the print medium can easily be re-adapted for other media, such as the computer display screen.

The language that Web browser uses, Hypertext Markup Language (HTML), is an example of an SGML-based language (there is a document type definition for HTML).

SGML is based somewhat on earlier generalized markup languages developed at IBM, including General Markup Language (GML) and ISIL. [2]

A 12.1.2 Extensible Markup Language (XML)

XML (Extensible Markup Language) is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere. For example, computer makers might agree on a standard or common way to describe the information about a computer product (processor speed, memory size, and so forth) and then describe the product information format with XML. Such a standard way of describing data would enable a user to send an intelligent agent (a program) to each computer maker's Web site, gather data, and then make a valid comparison. XML can be used by any individual or group of individuals or companies that wants to share information in a consistent way.

XML, a formal recommendation from the World Wide Web Consortium (W3C), is similar to the language of today's Web pages, the Hypertext Markup Language (HTML). Both XML and HTML contain markup symbols to describe the contents of a page or file. HTML, however, describes the content of a Web page (mainly text and graphic images) only in terms of how it is to be displayed and interacted with. For example, the letter "p" placed within markup tags starts a new paragraph. XML describes the content in terms of what data is being described. For example, the word "phonenum" placed within markup tags could indicate that the data that followed was a phone number. This means that an XML file can be processed purely as data by a program or it can be stored with similar data on another computer or, like an HTML file, that it can be displayed. For example, depending on how the application in the receiving computer wanted to handle the phone number, it could be stored, displayed, or dialled.

XML is "extensible" because, unlike HTML, the markup symbols are unlimited and self-defining. XML is actually a simpler and easier-to-use subset of the Standard Generalized Markup Language (SGML), the standard for how to create a document structure. It is expected that HTML and XML will be used together in many Web applications. XML markup, for example, may appear within an HTML page. [3]

The design goals for XML are:

1. XML shall be straightforwardly usable over the Internet.

2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

This specification, together with the associated standards, provides all the information necessary to understand XML and construct computer programs to process it.

An XML document could be valid and well-formed:

- A *textual object* is an *XML document* if it is either valid or well-formed.
- A *textual object* is said to be a *well-formed XML document* if, first, it matches the production labelled document, and if for each entity reference which appears in the document, either the entity has been declared in the document type declaration or the entity name is one of: amp, lt, gt, apos, quot.

  Matching the document production implies that:
  1. It contains one or more elements.
  2. It meets all the well-formedness constraints (WFCs) given in the grammar.
  3. There is exactly one element, called the root, or document element, for which neither the start-tag nor the end-tag is in the content of any other element. For all other elements, if the start-tag is in the content of another element, the end-tag is in the content of the same element. More simply stated, the elements, delimited by start- and end-tags, nest within each other.

A 12.1.3 XSL

XSL (Extensible Stylesheet Language, formerly called Extensible Style Language) is a language for expressing stylesheets. XSL is being developed under the auspices of the World Wide Web Consortium (W3C) and is currently in the working draft stage. It consists of three parts:

- XSL Transformations (XSLT): a language for transforming XML documents,
- the XML Path Language (XPath), an expression language used by XSLT to access or refer to parts of an XML document. (XPath is also used by the XML Linking specification).
- The third part is XSL Formatting Objects: an XML vocabulary for specifying formatting semantics.

An XSL stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses the formatting vocabulary [5].

That is, XSL is a language that describes how is going to be presented to the user the information stored in the XML document. XSL is a language oriented to presentation aspects such as titles, data organization in the page, etc. XSL is based on and extends the Document Style Semantics and Specification Language and the Cascading Style Sheet standards.

Think of an XML page as similar to an HTML page, but containing data in identified fields rather than text and graphics. XSL gives a developer the tools to describe exactly which data fields in an XML file to display and exactly where and how to display them. Like any style sheet language, XSL can be used to create a style definition for one XML document or reused for many other XML documents.

XSLT is a language for transforming XML documents into other XML documents.

XSLT is designed for use as part of XSL, which is a stylesheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary.

XSLT is also designed to be used independently of XSL. However, XSLT is not intended as a completely general-purpose XML transformation language. Rather it is designed primarily for the kinds of transformations that are needed when XSLT is used as part of XSL [4].

XSLT is used to describe how to transform the *source tree* or data structure of an XML document into the *result tree* for a new XML document, which can be completely different in structure. The coding for the XSLT is also referred to as a style sheet and can be combined with an XSL style sheet or be used independently.

The XSLT v1.0 document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from other documents.

Currently, they are working in the XSLT Version 2.0, that is a Working Draft dated on 14 February 2001 [5].

XPath is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer.

The XPath document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from other documents [5].

A 12.2XML databases

The term "native XML database" (NXD) is deceiving in many ways. In fact many so-called NXDs aren't really standalone databases at all, and don't really store the XML in true native form (i.e. text). To get a better idea of what a NXD really is, the NXD definition offered by the “XML:DB Initiative” says that a native XML database:

- Defines a (logical) model for an XML document -- as opposed to the data in that document -- and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of such models are the XPath data model, the XML Infoset, and the models implied by the DOM and the events in SAX 1.0.
- Has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage.
- Is not required to have any particular underlying physical storage model. For example, it can be built on a relational, hierarchical, or object-oriented database, or use a proprietary storage format such as indexed, compressed files.

The three main points can be summarized as the following:

- The database is specialized for storing XML data and stores all components of the XML model intact.
- Documents go in and documents come out.
- A NXD may not actually be a standalone database at all.

As should be clear from this definition, NXDs don't really represent a new low-level database model, and aren't intended to replace existing databases. They're simply a tool intended to assist the developer by providing robust storage and manipulation of XML documents.

Native XML databases fall into two broad categories:

- **Text-based storage** Store the entire document in text form and provide some sort of database functionality in accessing the document. A simple strategy for this might store the document as a BLOB in a relational database or as a file in a file system and provide XML-aware indexes over the document. A more sophisticated strategy might store the document in a custom, optimized data store with indexes, transaction support, and so on.
- **Model-based storage** Store a binary model of the document (such as the DOM or a variant thereof) in an existing or custom data store. For example, this might map the DOM to relational tables such as Elements, Attributes, Entities or store the DOM in pre-parsed form in a data store written specifically for this task. This includes the category formerly known as "Persistent DOM Implementations".

There are two major differences between the two strategies. First, text-based storage can exactly round-trip the document, down to such trivialities as whether single or double quotes surround attribute values. Model-based storage can only round-trip documents at the level of the underlying document model. This should be adequate for most applications but applications with special needs in this area should check to see exactly what the model supports.

The second major difference is speed. Text-based storage obviously has the advantage in returning entire documents or fragments in text form. Model-based storage probably has the advantage in combining

fragments from different documents, although this does depend on factors such as document size, parsing speed (for text-based storage), and retrieval speed (for model-based storage). Whether it is faster to return an entire document as a DOM tree or SAX events probably depends on the individual database, again with parsing speed competing against retrieval speed [8] [15].

Native XML databases differ from XML-enabled databases in three main ways:

- Native XML databases can preserve physical structure (entity usage, CDATA sections, etc.) as well as comments, PIs, DTDs, etc. While XML-enabled databases can do this in theory, this is generally not (never) done in practice.
- Native XML databases can store XML documents without knowing their schema (DTD), assuming one even exists. Although XML-enabled databases could generate schemas on the fly, this is impractical in practice, especially when dealing with schema-less documents.
- The only interface to the data in native XML databases is XML and related technologies, such as XPath, the DOM, or an XML-based API. XML-enabled databases, on the other hand, are likely to offer direct access to the data, such as through ODBC.

A 12.2.1 Solutions

The following databases are considered as “Native XML Databases”, that is databases that store the XML in “native” form, generally as indexed text or as some variant of the DOM mapped to an underlying data store. In the chapter “XML enabled databases” we include those DB that are not Native but they have extensions for transferring data between XML documents and themselves.

A 12.2.1.1 Birdstep RDM Mobile

Developer: Centor Software Corp.

URL: http://www.birdstep.com

License: Commercial

The Birdstep Raima Database Managers family gives you a choice of 3 products specifically designed for Server, Embedded or Mobile platforms. The Server and Embedded products are extremely mature and reliable, a necessity for any embedded application. The core of the Server and Embedded product has been stable in production environments for more than 15 years. They are still the best performing embedded database solutions around, distributed in numerous new high-transaction applications every year.

Just as the mobile device market has emerged, so has the Birdstep RDM Mobile Database Manager. A new product with solutions for the unique needs of a mobile application environment.

Currently, handheld devices are supplementing and replacing PCs as the preferred terminal for mobile users. The device can contain personal information, such as your contacts, your calendar and other key information to administer your everyday life. The database can also contain volatile information, such as fragments of maps, lists of restaurants, train schedules, etc. for the travelling user. Or it may contain extracts of a corporate database, for example customer information or product information for the sales rep in the field.

Expensive, unreliable and slow wireless communication makes it necessary to store information locally in the device, rather than relying on a wireless connection every time information is required. The Birdstep RDM Mobile offers all features required to store data locally in the device.

The Birdstep RDM Mobile database combines powerful mechanisms to meet the programmers' expectations, with a simple architecture adhering to the limitations found in handheld devices, Internet appliances and embedded systems. The database may run either as a pure memory-resident database, or as a database with secondary storage, such as flash and micro-disk.

The application has an object-oriented internal data model. It contains additional structural mechanisms and APIs aimed at XML-oriented, hierarchical, network-oriented and relational data structures. The database is capable of representing efficiently all data needed in a handheld wireless device (PDA) or an embedded system.

Because of its modular architecture, the database can be optimized for specific application areas, balancing small footprint, performance, robustness and programming features [9].

A 12.2.1.2 Tamino XML Server

Developer: Software AG

URL: http://www.softwareag.com/tamino/default.htm

License: Commercial

Tamino XML Server is a high performance data management platform, based upon XML and other open standard Internet technologies - Software AG is a member of the W3C Consortium - and helps with finding and managing any type of content across the enterprise. It allows to implement platform-independent B2B collaboration strategies in mission-critical environments. Its stunning performance, query capabilities and flexibility save a considerable amount of time and operational costs [11]. Tamino XML Server key features:

- **Native XML storage** to keep stored electronic documents intact and to provide persistence for transient business documents that helps keep track of business transactions for legal and governmental reasons.
- **Store for any type of data**. Tamino XML Server accepts "well-formed" XML documents and any other type of non-XML data for storage. These include office documents, audio, video, PDF files, etc. There is no need to create complicated data models before content can be stored.
- **Extensible by definition**. Adding new elements to stored XML documents is as simple as adding a new element to the associated schema which is also stored in Tamino. This leads to low administration costs.
- **Consolidates data from various sources in one place**, providing a single-server view on business data residing in distributed data sources.
- **Find-Engine for fast retrieval of XML-based content**. Tamino XML Server provides standard XPath-based query capabilities in combination with sophisticated high-speed indexing technology.
- **Built-in full-text retrieval at no extra cost**. Search type "Text" can be applied to every document stored in Tamino and is ideally suited for searching in un-indexed parts of XML documents, well-formed documents or plain text.
- **Multi-channel output formatting capabilities** are provided to support device-independent publishing of any stored content in a multitude of formats: XML documents, or such written in HTML, WML, PDF, etc.
- **Server extensions for custom functionality and application integration**. The server extensions (X-Tension) allow custom application-specific server-side features to be easily programmed in C, C++, Java or in other COM/DCOM-enabled languages, providing tight integration with EntireX Integration Server.

Tamino XML Server is a suite of products built in three layers -- core services, enabling services, and solutions (third-party applications) -- which may be purchased in a variety of combinations.

Core services include a native XML database, an integrated RDB, schema services, security, administration tools, and Tamino X-Tension allowing to write extensions customizing server functionality. More specifically:

- **XML engine** is the central and most powerful component in the Tamino XML Server. Its high performance and robustness are the basis for Tamino core services such as highly efficient storage, querying and retrieval of XML documents. The core services include *X-Query* and *full-text retrieval* functionality, based on major building blocks such as an integrated XML parser, a query processor, an object processor, an object composer, and an **integrated native XML data store**. The direct storage of XML objects without further to other structures is the main reason for Tamino's excellent performance.
- **Data Map** is the knowledge base of Tamino's server core. It contains XML metadata, such as Tamino Schemas derived from DTDs or XML Schemas(1), style sheets, relational schemas, etc., defining the rules according to which XML objects are stored and composed. The Data Map determines how XML objects, embedded in XML documents, are mapped to physical DB structures, whether they reside internally or externally, allowing existing DBs to be enabled for XML technology and the Web.

  (1)Tamino XML Server supports W3C's XML Schema based on the XML engine's and the data map's capabilities. Therefore, Tamino is very flexible in XML documents handling and supports the storage of both well-formed XML (without an explicit schema definition) and valid XML (adhering to a schema).

Among others, the **enabling service components** provide interconnection facilities with:

- **Tamino X-Port** is an integrated Web server interface connecting Tamino's server core to the Internet via standard Web servers (such as Apache, Microsoft IIS, IBM HTTP-Server or the iPlanet Web Server) without the need to write scripts or servlets. Based upon this interface, documents stored in Tamino or just parts thereof can be accessed directy via the HTTP protocol simply by applying URLs.

- **Tamino API for Java** is a object-oriented service providing functions and class libraries for Java to let applications easily communicate with Tamino XML Server. The API allows for accessing, navigating on, modifying and updating of documents stored in Tamino and supports DOM2-, JDOM-, SAX2-, as well as stream-based access to Tamino. Also HTTP Client APIs (DOM APIs) for JScript and ActiveX. The HTTP API for ActiveX allows to access Tamino XML Server via DCOM-compatible programming languages, such as C++, VisualBasic or Natural, running on Windows operating systems. An extension to the Tamino API for Java connects Tamino XML Server to major application servers / EJB servers.
- **Tamino X-Node** provides access to existing heterogeneous databases with traditional data structures, regardless of database type or location (e.g, Adabas, SQL or other data sources accessible via Open APIs). Tamino X-Node maps this data to XML structures, providing continued usability of existing DB infrastructures and thus protecting legacy IT investments, and acts as virtual DBMS, meaning a central server for existing DBs over the Web and for Web-oriented applications.

A 12.2.1.3 eXtensible Information Server (XIS)

Developer: eXcelon Corp.

URL: http://www.exln.com/products/xis/

License: Commercial

XIS stores XML documents in eXcelon's ObjectStore object-oriented database. Documents are stored in a proprietary, B-tree-like structure (for performance reasons) and can be indexed using both value and structural indexes (value indexes index element and attribute values; structural indexes index element and attribute names). Documents can be arranged in collections; these can be nested, resulting in a file system metaphor.

XIS supports queries through XPath with extension functions and a proprietary update language. Statements in the update language consist of an XPath to a node, an operation on that node (insert before/after, update, delete), and any data needed to carry out the operation. XIS also supports server-side functions, which can be used in XPath statements, as triggers associated with update statements, or as stored procedures. These are written in Java and can directly manipulate data in the database through a server-side DOM implementation.

Related to queries is a proprietary linking language. This allows users to link existing documents as well as to build virtual documents that consist of nothing but links. Links are traversed transparently during queries and update operations, which means that virtual documents can be used to perform queries and updates over multiple documents in a single operation. Note that the application must currently enforce the referential integrity of links. That is, it must ensure that the document/fragment to which a link points actually exists.

XIS can integrate backend data through the XConnects module, which uses the Data Junction Universal Translation Suite. This provides links to many different data formats, including relational databases. Because the links are two-way, it means that backend data sources can be updated through XIS.

XIS supports transactions and can participate in XA transactions. However, it cannot currently manage XA transactions, so the application must coordinate any XA transactions that include XIS and other data sources, such as backend data stores. Other database features include distributed caching, partitioning, online backup and restore, and clustering support.

Finally, XIS comes with a Java API, a built-in XSLT processor, and a set of GUI development tools. These include an XML editor, an XSLT editor, a schema editor (XML Schemas and DTDs), an XSLT/Java debugger, and tools for mapping backend data to XML documents.

A 12.2.1.4 Xindice

Developer: Apache Software Foundation

URL: http://xml.apache.org/xindice/

License: Open Source

Apache Xindice is a database designed from the ground up to store XML data or what is more commonly referred to as a native XML database.

It is written in Java that is designed to store large numbers of small XML documents. It can index element and attribute values and compresses documents to save space. Documents are arranged into a hierarchy of collections and can be queried with XPath. Collection names can be used as part the XPath query syntax, meaning it is possible to perform XPath queries across documents. For updates, Xindice supports the

XUpdate language from the XML:DB Initiative. Finally, Xindice comes with an experimental linking language that is similar to XLinks, and allows users to replace or insert content in an XML document at query time.

Xindice supports three APIs: the XML:DB API (also from the XML:DB Initiative), a CORBA API, and an XML-RPC plugin which supports access from languages such as PHP, Perl, and Applescript. In addition, Xindice provides XMLObjects, which allows users to extend the server functionality.

Xindice comes with a set of command line tools for using and administering the database [14].

Xindice is the continuation of the project that used to be called the dbXML Core. The dbXML source code was donated to the Apache Software Foundation in December of 2001 [11].

A 12.2.1.5 eXists

Developer: Wolfgang Meier

URL: http://exist.sourceforge.net

License: Open Source

eXist is an Open Source native XML database with pluggable storage backends and support for fulltext search. XML is either stored in the internal, native XML-DB or an external RDBMS. The search engine has been designed to provide fast XPath queries, using indexes for all element, text and attribute nodes. eXist is lightweight and well suited for large document collections. The server is accessible through easy to use HTTP and XML-RPC interfaces and supports the XML:DB API for Java programming.

Beginning with version 0.6, there are two different backends:

- Native backend: uses it's own internal data structures for storage and indexing. The native backend is completely written in Java and does not require any additional installation. It is very fast when retrieving larger amounts of document content. Version 0.7 comes with a redesigned indexing architecture to eliminate some performance problems and increase scalability.
- Relational backend: uses an relational database system to store documents and to build up the index structure. Currently MySQL and PostgreSQL are supported.

For both backends, the basic model employed for storing and retrieving XML documents is the same. eXist's indexing architecture is based on an alternating numbering scheme for DOM nodes, which provides fast processing of path expressions [16].

A 12.2.1.6 XML enabled databases

12.2.1.6.1 Oracle 9i XDB

Developer: Oracle

URL: http://otn.oracle.com/tech/xml/xmldb/content.html

License: Commercial

Oracle 9i XDB supports both XML-enabled and native storage of XML data. It blurs the boundaries between relational data and XML data by providing SQL features (implemented at the engine level) that allow users to view relational data as XML and XML data as relational. The main feature is the XMLType data type. This is a predefined object type that can store an XML document. Like any object type, XMLType can be used as the data type of a column in a table or view. The latter usage is important, as it means that an XML "view" -- a virtual XML document -- can be constructed over any data, regardless of whether it is relational data or XML data.

Oracle XML DB is a set of features in the Oracle9*i* database server that encompass both SQL and XML in a highly interoperable manner. It is not a separate server. Oracle XML DB adds a native XML repository to the database [12].

The XML data-model encompasses both *unstructured* content and *structured* data. Oracle XML DB provides new capabilities for both content-oriented and data-oriented access. XMLType data can be stored in either of two ways: with object-relational storage or as a CLOB. The storage options are interchangeable and XML applications use the same code regardless of which option is chosen (changing from one storage type to the other requires a database export and import.)

When using object-relational storage, users define the mapping with an annotated XML Schema. This states the name and data type (which can be an object type) of the SQL structure used to store a given element or attribute. Users can create their own mappings or use a default mapping generated by XDB. Unlike most

software that uses an object-relational mapping to store XML, XDB can round-trip XML documents at the level of the DOM. To do this, it uses hidden columns to store information that is not directly modelled by SQL, such as sibling order, processing instructions, comments, and whether a column corresponds to an element or an attribute.

In spite of the fact that applications use the same code regardless of how XMLType data is stored, there are a number of practical differences between object-relational and CLOB storage.

XMLType data can be accessed in several ways. Java Beans can be used when the data uses object-relational storage. The DOM can be used regardless of the storage option (the DOM implementation populates nodes lazily for better concurrency). In addition, data can be accessed by executing SQL statements that use the operators mentioned earlier.

For developers who see XML as documents (news stories, articles, etc.), Oracle XML DB provides an XML Repository accessible from standard protocols and queryable from SQL. This provides a file system-like view of XMLType objects in the database. That is, XMLType objects (regardless of whether they actually contain XML data or are just XML views over relational data) can be assigned a path and corresponding URL in the repository hierarchy. These can then be accessed via WebDAV, FTP, JNDI, and SQL; the latter has special operators for this purpose. In addition, the repository maintains properties for each object, such as owner, modification date, version, and access control.

12.2.1.6.2 DB2 XML Extender

Developer: IBM

URL: http://www-4.ibm.com/software/data/db2/extenders/xmlext.html

License: Commercial

The DB2 XML Extender is a DB2 UDB Extender for transferring data between XML documents and DB2. XML DTDs are mapped to relational schema (and vice versa) with the XML-based Data Access Definition (DAD) language. The language comes in two flavours: SQL mapping and RDB node mapping. SQL mapping is a template-based language and can only be used to transfer data from the database to an XML document. RDB node mapping is an object-relational mapping. It can be used to transfer data both to and from the database. A visual tool is provided for constructing DAD documents -- that is, mapping elements and attributes to tables and columns.

Applications use stored procedures to invoke the extender, which then stores or retrieves data based on the DAD document. The XML Extender manages DAD documents and DTDs in its own tables, saving applications from having to do this and optimizing access.

The DB2 Text Extender contains a variety of search technologies, such as fuzzy searches, synonym searches, and searches by sentence or paragraph that can be used when an XML document is stored in a single column.

It supports sending and retrieving XML documents from MQSeries message queues. Also supports Web services with Web services Object Runtime Framework (WORF) Beta [13].

12.2.1.6.3 SQL Server 2000

Developer: Microsoft

URL: http://msdn.microsoft.com/library/periodic/period00/sql2000.htm

License: Commercial

Microsoft SQL Server 2000 supports XML in three ways: the FOR XML clause in SELECT statements, XPath queries that use annotated XML-Data Reduced schemas, and the OpenXML function in stored procedures. SELECT statements and XPath queries can be submitted via HTTP, either directly or in a template file.

The FOR XML clause has three options, which specify how the SELECT statement is mapped to XML. RAW models the result set as a table, with one element (named "row") returned for each row. Columns can be returned either as attributes or child elements. AUTO is the same as RAW, except that: 1) the row elements are named the same as table name, and 2) the resulting XML is nested in a linear hierarchy in the order in which tables appear in the select list.

EXPLICIT allows you to model an XML document using a series of SELECT statements that are UNIONed together. In its simplest form, each SELECT statement is numbered and includes the number of its parent statement. The results of an individual statement are modelled as a table and an element is created for each

row. This is placed in the XML document beneath the appropriate parent element. Assuming there is a relation between the result sets (for example, each contains a sales order number), the children are nested as one would expect. EXPLICIT allows you to create canonical object-relational mappings from the database to an XML document, but supports more sophisticated queries as well.

The OpenXML function uses a table-based mapping to extract any part of an XML document as a table and use it in most places a table name can be used, such as the FROM clause of a SELECT statement. An XPath expression identifies the element or attribute that represents a row of data. Additional XPath expressions identify the related elements, attributes, or PCDATA that comprise the columns in each row, such as the children of the row element.

A 12.3 Assessment

A 12.3.1 XML

The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web [6]. In the last months, a new generation of languages is being used to represent the information handled in the application servers, namely XML (Extensible Markup Language) for information representation and XSLT (Extensible Style Sheet Transformations) to transform this information on the fly depending on the device accessing the information. The normal procedure is to store information in XML and present it to the users by applying "transformations" in XSLT, depending on the user requiring the information. The strong point about this procedure is that the information can be kept in one standard (XML) and depending of the user different "views" of the same information can be produced on the fly by applying XSLT.

A 12.3.2 XML databases

When we talk about XML data, the clearer solution is the use of an native XML database. The benefit of a native solution is that you don't have to worry about mapping your XML to some other data structure. You just insert the data as XML and retrieve it as XML. This is especially valuable when you have very complex XML structures that would be difficult or impossible to map to a more structured database [14].

There is only one hard requirement for any application that wants to use an NXD: the application must use XML. Beyond that there are no strict rules for what type of applications should or should not be built with an NXD, though some loose guidelines can be offered. In general, NXDs excel at storing document-oriented data (e.g. XHTML or DocBook), data that has a very complex structure with deep nesting, and data that is semi-structured in nature. Basically, if the data is represented as XML and is "kind of fuzzy" an NXD will probably be a good solution. An NXD can store any type of XML data, but probably isn't the best tool to use for something like an accounting system where the data is very well-defined and rigid.

Some potential application areas for using XML and native XML databases include Corporate information portals, Catalogue data, Manufacturing parts databases, Medical information storage, Document management systems, B2B transaction logs, Personalization databases, Human resources, and so on.

Beyond this, NXDs are simply a new tool and their ultimate utility will be determined by the creativity of the developers using them.

A 12.3.3 Conclusions for ALFANET

The use of **XML** in ALFANET will allow to separate data content from presentation. The separation of both aspects is very important. In those web applications where data and presentation are linked, the maintenance of them is very complex. Any variation in the aspect of the application imply to recompile the whole application and to modify lots of graphic elements.

In order to isolate the information structure and content from the visual presentation of these data we propose the use of XML in ALFANET project. Besides, we will need XSLT transformation patterns that will be applied to the data structure defined in XML. The output will be automatically generated in HTML, WML, … and sent to the client. With the use of these standards we will get more flexibility in our application, besides more trazability, maintainability, change resistance, and provide to the whole application an homogeneous aspect for all elements of the same type.

As a consequence of this data and presentation separation, it will possible to reuse the modules in charge of the data generation. Using different presentation templates for the same data set, ALFANET will be able of:

- Present the same content with different aspects, depending on the context where it will be presented or depending on the user profile or preferences
- Provide service from the same application to different types of client devices (PC, telephone, …). In order to do that, it is necessary to include some routines in the server that will decide depending on the type of connected client, and will apply different specialized templates.
- Provide compatibility and interchangeability by the use of approved XML standards.

All these features guide us to the use of XML in ALFANET, and it would be very important to store the data in a XML database. The **XML database** will avoid the transformation from raw data to XML and vice versa when we have to access or store to the database.

In any case, it will be in a more advanced stage when this decision will be taken, but at the moment all works are focused on this idea.

A 12.4 References

[1] http://www.personal.u-net.com/~sgml/sgml.htm

[2] http://whatis.techtarget.com/definition/0,,sid9_gci214201,00.html

[3] http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci213404,00.html

[4] http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci214420,00.html

[5] http://www.w3.org/Style/XSL/

[6] http://www.w3.org/XML/

[7] http://www.rpbourret.com/xml/ProdsNative.htm#sybase

[8] http://www.xml.com/pub/a/2001/10/31/nativexmldb.html

[9] http://www.birdstep.com/database_technology/rdm_mobile.php3

[10] http://www.softwareag.com/tamino/background.htm

[11] http://xml.apache.org/xindice/

[12] http://otn.oracle.com/tech/xml/xmldb/content.html

[13] http://www-3.ibm.com/software/data/db2/extenders/xmlext/

[14] http://www.rpbourret.com/xml/XMLDatabaseProds.htm

[15] http://www.karto.ethz.ch/neumann/caving/cavexml/xml_tools.html

[16] http://exist.sourceforge.net/

Appendix 13 Knowledge management tools

A 13.1 Overview

Knowledge Management (KM) is not easy to define. KM for an organization consists of activities focused on the organization gaining knowledge from its own experience and from the experience of others, and on the judicious application of that knowledge to fulfill the mission of the organization. These activities are executed by marrying technology, organizational structures, and cognitive based strategies to raise the yield of existing knowledge and produce new knowledge. Specific knowledge management activities help focus the organization on acquiring, storing and utilizing knowledge for such things as problem solving, dynamic learning, strategic planning and decision making. It also protects intellectual assets from decay, adds to firm intelligence and provides increased flexibility. [1] [2]

Knowledge can be classified into various categories:

Tacit knowledge	Knowledge that cannot be articulated
Implicit knowledge	Knowledge that can be articulated but has not been articulated.
Explicit knowledge	Knowledge that is articulated and more often than not, captured in the form of text, tables, diagrams etc.
Procedural knowledge	Knowledge that manifests itself in the doing of something
Declarative knowledge	Knowledge that consists of descriptions of facts and things or of methods and procedures
Strategic knowledge	Knowing when to do something and why to do it

KM tools and techniques are defined by their social and community role in the organisation in:

- facilitation of knowledge sharing and socialisation of knowledge (production of organisational knowledge);
- conversion of information into knowledge through easy access, opportunities of internalisation and learning (supported by the right work environment and culture);
- conversion of tacit knowledge into "explicit knowledge" or information, for purposes of efficient and systematic storage, retrieval, wider sharing and application. [4]

Tools and techniques are necessary and certainly useful to manage information/documentation. These tools include -- indexing, storage, retrieval and circulation systems, replication, collaborative group work on a common document, and so on. There are various degrees of integration stipulated under different types of software. One has to make an appropriate choice depending on mission-criticality, desirable degrees of flow-control, and so on. The software has also to be customised within the KM-strategic framework.

The KM tools are based on diferent technological areas:

- Search engines which are used to find documents in a database or repository by using key words or indexed topics. There are a variety of these kinds of tools. The most popular kinds are those that use either boolean operators (and, or) or natural language. Some permit the addition of language to their indices others do not.
  Other search engines, for instance Google, find results in part by looking to see how frequently a site is linked to by other sites and on that basis ranks sites by importance. It also looks for words close together rather than anywhere on a page or site.
- Neural networks which identifies patterns or ideas in text and look for similar ideas in other sources automatically. "Concept agents" can analyse new data and classify them according to the dynamic rules that they "learn". It also makes new connections with old information by continually modifying its network of relationships.

- Artificial Intelligence and Data mining apply techniques and algorithms from these areas to create systems that infer and organize information automatically. Machine reasoning, inferring, understanding and validating fall into this area.
  Data mining provides algorithms to run against unstructured content. They look beyond individual words and focus on the context created by the surrounding text. For example, the text surrounding the term "board" would identify whether its usage referred to lumber, an executive team, or ships at sea. The software automatically recognizes these nuances and organizes the documents accordingly.
- Taxonomies provides a formal system of orderly classification of knowledge. Users can browse or search the directories to find the document they are looking for.
- Knowledge Map maps the expertise of an organization is valuable for several reasons. It provides easy access to a map of expertise of the organization can connect people when they need guidance resulting in quicker response rates, reduction of re-invention of the wheel, increased employee satisfaction and more. Maps can be used then to pull people in to assist on current projects or for offering training to employees who have existing good basic skills to equip them with additional skills the organization will need for future projects. Considerations include: skills, expertise, experience, and location.

A 13.2 Solutions

A 13.2.1 Classification of KM tools

The most useful KM tools can be grouped in [6]:

- Search and Classification tools. These tools allow to search and classify information. The users can make use of all the contents in the moment they need to look up them. This kind of KM tools automatises most of the following processes:
  - indexing, in order to facilitate the access
  - cataloguing to normalise the information description
  - labelling to facilitate the classification
  - linking to establish the relationship between contents.

- Filtering and Personalized Distribution tools. These tools automatically provide information based on the user profile. So the users don't have to hunt for the information. They want it delivered to them. These tools are designed with that in mind.

- Collaboration, groupware, conferencing and mailing tools. These tools are permitting people to easily share tacit knowledge, their ideas, work together, brainstorm, collaborate. People just need to be educated that the technology is available and easy to use.
  Different technologies are combined within these tools:
  - Chat Extended conversations between two or more people who are online, typing away at their keyboards, responding to each other.
  - Data Conferencing
  - Internet telephony
  - Streaming audio and video via the internet
  - Virtual Meeting Rooms Software which enables dozens or more of people to work together over the web, hold discussions, share documents, reach decisions. With moderator, auditorium for large sessions, methodology to poll members, breakout rooms for smaller sessions, methodologies to hold one-on-one or small group conversations.

- Corporate Intranet and Extranets

  An intranet is a centralized electronic repository of information (typically accessed via computer on a company's network with a browser based for interface. There is unlimited potential for the uses of intranets including access to static information like HR forms, work product, and online resources - as

well as interactive tools for learning, collaborating and more. Listed here are solutions for small businesses who want to build an intranet solution without tremendous expense either in infrastructure (technology or staff) and time.

The same, extended to an outer ring and typically accessed via computer from a company's web site, is an extranet. The extranet users are specific clients who want an immediate tie to the company, the information it has, and their work with the company. There is unlimited potential for the uses of extranets including access to static information like advertising, newsletters, client specific work product, and online resources. There are also applications for interactive tools for collaborating and more. An extranet is frequently a portion of a company's web site that is password protected for use by authorized clients.

- Simulation tools. Simulation is one of the most potent knowledge management techniques known today. It is widely used for such critical business processes such as knowledge creation, prototyping, design, education, training, behaviour modification, and economic forecasting, to name a few. It is among the most important methods in the KM practioners' portfolio. Simulation is used widely by top universities, leading institutions and governments for development and management of their core knowledge assets.
  Simulation and Modeling Complex Adaptive Systems apply metrics and models to knowledge representations to create systems that can predict how a given process will effect an organizations ability to manage its knowledge (intellectual capital). In this way, the efficience is improved with the organisation and errors are detected before making.

A 13.2.2 Commercial tools

In the next subsection some commercial KM tools belonging to these groups will be shortly introduce. Due to their features, some of these tools could belong to several groups described above.

	Search and Classification	Filtering and Distribution	Collaboration	Corporate Portal	Simulation
SAP Knowledge warehouse	X				
Lotus Notes & Domino	X		X		
Lotus Knowledge Management	X	X	X	X	
4i ECM Platform	X			X	
Livelink Enterprise Workspace	X				
AltaVista Enterprise Search	X				
RetrievalWare	X				
Intelligent Miner for Text	X				
Verity Information Server	X				
DB/TextWorks	X				
Inxight Categorizer	X				
Hyperwave eKnowledge infraestructure	X			X	
PeopleSoft Portal Solutions				X	
Vignette Content Suite		X		X	
BroadVision		X		X	
ATG Enterprise Suite Portal		X			
Tridion Dialog Server				X	
Autonomy Software		X			
Netscape Compass Server		X			

	Search and Classification	Filtering and Distribution	Collaboration	Corporate Portal	Simulation
AWD Business Intelligence		X			
Automated Work Distributor		X			
Meta4 KnowNet			X		
Microsoft NetMeeting			X		
Teamware Office			X		
Verity K2 Enterprise				X	
Livelink: Personal Workspace				X	
Teamware Pl@za				X	
ProSim					X
ADONIS					X

A 13.2.2.1 SAP Knowledge warehouse

Developer: SAP

URL: www.sap.com/uk/education/solutions/knowledge.asp

License: Commercial

SAP Knowledge warehouse supports all stages of knowledge management, including knowledge modeling, the managment of information objects, transfer, distribution and translation of knowledge, and efficient company-wide information access. Users needing information can access the entire contents of the SAP Knowledge Warehouse either from SAP R/3 or from their personal workspace via mySAP.com, using a standard web browser.

A 13.2.2.2 Lotus Domino & Notes

Developer: IBM/Lotus

URL: http://www.lotus.com/lotus/products.nsf/fa_prohomepage

License: Commercial

Domino Family of Servers provide a multi-platform foundation for collaboration and e-business, driving solutions from corporate messaging to Web based transactions and everything in between.

Domino's integrated application services-such as security, workflow and content management-optimize the platform for rapid delivery of the collaborative Web applications you need to initiate and strengthen key business relationships. Built-in connection services provide live access to leading relational databases, transaction systems and ERP applications.

Notes is an integrated, Web-like environment that provides users with quicker access to and better management of many types of information including Domino and Internet-based e-mail, calendar of appointments, personal contacts and to-dos as well as Web pages, News Groups and intranet applications.

A 13.2.2.3 Lotus Knowledge Management Family

Developer: IBM/Lotus

URL: http://www.lotus.com/lotus/products.nsf/fa_prohomepage

License: Commercial

Discovery Server is knowledge server for e-business users. It searches or browses for information and subject matter experts from multiple locations, collaborates with colleagues instantly, and increases knowledge sharing.

K-station is a knowledge portal with out-of-the-box collaborative capabilities. It uses a Web browser user interface to access virtually any information source: from Web applications to Microsoft Office documents to back-end data.

Domino.doc provides Document and Records Management for the Distributed Enterprise. Domino.Doc delivers scalability, flexibility and low cost of ownership required to support both enterprise-wide documents and records management, while serving as a foundations for knowledge management.

Domino Workflow is a stand alone product that works on top of Domino to provide the ability to develop, manage, and monitor business processes and help eliminate the downfalls of paper-based work. Domino Workflow visually represents each step in the workflow process so you can make changes with a point and a click

Domino Extended Search performs parallel searches across heterogeneous data sources, locating and presenting relevant information in a context familiar to the user.

A 13.2.2.4 4i ECM Platform

Developer: Documentum

URL: http://www.documentum.com/products/content-management_products.html

License: Commercial

Documentum 4i Enterprise Content Management (ECM) Platform is the industry standard for managing, distributing, and exchanging large volumes of content within and beyond the enterprise -- to customers, employees, partners, and suppliers.

By integrating dynamic content, complex business processes, and people everywhere, Documentum 4i enables seamless collaboration, communication, and knowledge sharing on a global scale to truly power your e-business initiatives.

Built as an open, scalable platform, Documentum 4i enables you to leverage current technology investments with integrations to industry-leading tools while also enabling rapid application development and deployment with a robust industry-standard J2EE development environment, native XML capabilities, and a services-based architecture.

A 13.2.2.5 Livelink Enterprise Workspace

Developer: OpenText

URL: http://www.opentext.net/livelink/details/ workspace_enterprise.html

License: Commercial

The Enterprise Workspace provides users with fingertip access to corporate knowledge. Use this area to share best practices, policies, procedures, business critical news and all projects throughout the enterprise. In the Enterprise Workspace, corporate knowledge and information can be easily organized, shared, and accessed. From Marketing information to Human Resource forms, this Workspace provides the ideal forum for sharing and accessing corporate information that pertains to all members of an organization.

A 13.2.2.6 AltaVista Enterprise Search

Developer: AltaVista

URL: http://solutions.altavista.com/en/products/aventerprise.shtml

License: Commercial

AltaVista Enterprise Search brings order to diverse information environments by tapping into various sources of information, both structured and unstructured. Augmenting traditional search, advanced categorization technology helps organizations build easy-to-navigate information nodes and browsable taxonomies that provide another method for users to find relevant information. This technology enhances not only the quality of results presented, but the overall user experience as well — the process of looking for and finding needed information becomes much more intuitive and less time-intensive.

It can access information is stored in ERP, CRM or SFA systems, or any other enterprise application, providing total accessibility to all corporate information assets. In this way, all users — regardless of their level of training on individual applications — receive quick access to information.

A 13.2.2.7 RetrievalWare

Developer: Convera

URL: www.convera.com/Products/products_rw.asp

License: Commercial

RetrievalWare puts the power of knowledge behind your information portal by allowing your employees to access your organization's content using intelligent search technology or to organize that content into categories for browsing. Whether searching or browsing, they are quickly delivered complete, accurate, and relevant information. RetrievalWare creates a complete inventory of all your enterprise assets, then enables users to search more than 200 document types on file servers, in groupware systems, relational databases, document management systems, Web servers and more while respecting access rights — all from a common user interface. This access is managed by RetrievalWare's Synchronizers which recognize any changes system-wide and automatically update the RetrievalWare index.

A 13.2.2.8 Intelligent Miner for Text

Developer: IBM

URL: http://www-3.ibm.com/software/data/iminer/fortext/

License: Commercial

Intelligent Miner for Text turns unstructured information into business knowledge for organizations of any size, from small businesses to global corporations. The knowledge-discovery "toolkit" includes components for building advanced text mining and text search applications.

It provides a wide range of text analysis tools for feature extraction, clustering, categorization and summarization. It also includes full-text retrieval components, using the IBM Text Search Engine and Web-access tools, such as NetQuestion Solution and IBM Web Crawler.

A 13.2.2.9 Verity Information Server

Developer: Verity

URL: http://www.verity.com/products/infoserv/index.html

License: Commercial

Information Server indexes, searches and retrieves information on Web and file servers distributed across the enterprise and stored in virtually any format. It includes powerful search navigation facilities including document clustering, automatic summarization and query by example, and supports business rules, Verity's concept-based knowledge mapping system.

It works with a variety of companion gateway products, enabling indexing and retrieval of information from relational databases and popular repositories such as Lotus Notes and Microsoft Exchange.

A 13.2.2.10 DB/TextWorks

Developer: Inmagic, Inc

URL: http://www.inmagic.com/prod_data_dbt.htm

License: Commercial

Inmagic DB/TextWorks is a special combination of database and text retrieval software that enables you to build networked and standalone textbases to manage diverse types of information including text, images, and multimedia.It has what you want from a text retrieval system, including instantaneous keyword searches with a powerful search engine.

A 13.2.2.11 Inxight Categorizer

Developer: Inxight Software, Inc

URL: http://www.inxight.com/products/categorizer/

License: Commercial

Inxight Categorize is an enterprise-class, natural language processing system that automatically classifies documents by subjects. It has the scalability to manage thousands of categories and millions of documents. And, Categorizer's patented natural language processing and machine learning techniques offer the highest possible level of accuracy in 12 languages.

Categorizer automatically organizes and delivers timely, relevant and accurate information to corporate knowledge bases and individual users, enabling them to take immediate advantage of the most up-to-date

information available. Using Categorizer, information-intensive organizations can process, analyze and deliver their information assets while simultaneously reducing the cost of managing them.

A 13.2.2.12 Hyperwave eKnowledge infraestructure

Developer: Hyperwave

URL: http://www.hyperwave.com/e/products/ekp.html

License: Commercial

Hyperwave offers a holistic view of Knowledge Management and delivers it through the eKnowledge Infrastructure, a broad integrated product suite of applications including, content and document management, search and retrieval, workflow, collaboration and e-learning. The applications can be implemented as individual or multiple solutions

The Hyperwave eKnowledge Suite, a collaborative knowledge management solution, offers content and document management functionality for rapid deployment of corporate intra- and extranets. The Hyperwave eKnowledge Suite can easily capture and organize the information within an organization and distribute it efficiently within the enterprise, as well as to business partners and customers.

The Hyperwave eKnowledge Portal offers a comprehensive single point of information exchange, to enhance and integrate activities across the extended enterprise.

The Hyperwave eLearning Suite is the first solution to combine education with knowledge management, and guarantees an ongoing, information and knowledge transfer between employees at any time or place. It combines web-based training, specifically learning in a virtual classroom via the intranet or extranet, with the efficient use of vast amounts of corporate memory.

A 13.2.2.13 PeopleSoft Portal Solutions

Developer: PeopleSoft

URL: http://www.peoplesoft.com/corp/en/products/index.asp

License: Commercial

Portal Solutions are a framework for every type of system and content your users need. You gain a customized, role-based homepage that is the single gateway to all your critical information—such as PeopleSoft business applications, databases and applications from other vendors, external content, and more. PeopleSoft Portal Solutions provides a complete infrastructure solution, with the tools, open integration framework, and platform to create a real-time enterprise.

A 13.2.2.14 Vignette Content Suite

Developer: Autonomy

URL: http://www.autonomy.com/

License: Commercial

Vignette Content Suite is the most open, scalable, reliable, secure and easy-to-manage solution on the market with which organizations can deploy dynamic Web applications using content from any data source. With an end goal of building effective online relationships, you need to effectively execute throughout all the phases of the content management lifecycle. Vignette Content Suite manages content according to your preferred business processes, provides insight into your customers' behavior, delivers personalization, Integrates and automates business processes

A 13.2.2.15 BroadVision

Developer: Broadvision

URL: http://www.interleaf.com/OneToOne/SessionMgr/home_page.jsp

License: Commercial

Personalized enterprise business portals are quickly becoming the standard platform for e-business initiatives. The BroadVision family of portal-powered applications, composed of BroadVision One-To-One Portal BroadVision One-To-One Commerceand BroadVision One-To-One Content has set a new standard for enterprise business portals by integrating advanced personalization, enterprise content management and BroadVision's unmatched commerce capabilities.

A 13.2.2.16 ATG Enterprise Suite Portal

Developer: ATG

URL: http://www.atg.com/en/products/portalsuite/

License: Commercial

The ATG Enterprise Portal Suite provides a complete portal solution to build more productive relationships with every user in your business community. It's rich out-of-the-box features, powerful application framework, and robust personalization capabilities allow you to quickly assemble and deploy customized, high-performance Web portals that provide unique user experiences. Offer your customers fast, individually tailored service and greater self-administration. Keep employees and partners informed and productive with secure access to collaborative tools and up-to-the-minute information. For your organization, provide a unified, enterprise-wide strategy that will build loyal relationships and drive down costs.

A 13.2.2.17 Tridion Dialog Server

Developer: Tridion

URL: http://www.tridion.com/com/Product/overview.asp

License: Commercial

Tridion DialogServer is a web content management solution. It efficiently structures enterprise-wide processes for the creation, management, distribution and delivery of business critical content across multiple websites, portals, e-applications and paper based publications.

Tridion DialogServer is built on a native XML and software component architecture (COM+ / J2EE), which guarantees easy integration with existing enterprise and e-business systems, including ERP and CRM applications and industry-standard application servers.

A 13.2.2.18 Autonomy Software

Developer: Autonomy

URL: http://www.autonomy.com/

License: Commercial

Autonomy Software automates the categorization, tagging, hyperlinking and personalization of large volumes of unstructured content for new media publishers and large corporate enterprises. As site visitors retrieve web pages, the system reads and understands the text and automatically suggests relevant content from other sources. It also automatically alerts users whenever a subject of interest comes up in a chat room conversation or appears in a breaking news story.

A 13.2.2.19 Netscape Compass Server

Developer: Netscape

URL: http://wp.netscape.com/compass/v3.0/index.html

License: Commercial

Netscape Compass Server provides a comprehensive set of tools that help administrators gather and organize enterprise resources scattered across intranets so that users can easily find and retrieve information whenever they need it. Besides offering flexible search and browse services, Netscape Compass Server allows users to subscribe to personally profiled topics of interest and receive a daily summary of relevant information from the intranet and the Internet. Setup wizards allow rapid installation, configuration, and deployment, while support for multiple platforms, servers, and document types allows administrators to easily manage and customize intranet information to suit the ongoing needs of users.

Netscape Compass Server enables users to specify individual interests and then receive customized daily summaries of relevant information. Users simply create a personal interest profile by subscribing to Compass Server categories that match their interests. Users receive a personalized My Compass newsletter, delivered daily to their email inbox, or personal web page.

An enterprise-scalable solution for gathering and indexing the contents of an entire corporate intranet, Netscape Compass Server enables users to search for information they need and browse areas of interest. Now users can easily access previously hidden intranet information resources with a click of a mouse.

A 13.2.2.20 AWD Business Intelligence

Developer: DST Systems, Inc

URL: http://www.dstsystems.com/fs/bo_awd/bobusi.html

License: Commercial

Reporting and trend analysis tools improve processing efficiency and decision making, helping to tame the mounds of generated operational data within the workflow environment. AWD/Business Intelligence has an "event-based" process that captures AWD workflow data and prepares it for reporting purposes. AWD/Business Intelligence ensures that all report data is correct, unambiguous, and meaningful.

AWD/Business Intelligence includes a graphical user interface (GUI) that allows easy access to the reported data. Using the GUI's standard work management queries, supervisors and managers can monitor daily operations and make more informed decisions regarding their resource allocations and current workloads.

AWD/Business Intelligence also includes a powerful tool to analyze trends using a multi-dimensional data model. This data model is designed to resemble the real-world business environment, eliminating the need for complex report queries and strengthening strategic planning.

A 13.2.2.21 Automated Work Distributor

Developer: DST Systems, Inc

URL: http://www.dstsystems.com/products/opawd.html

License: Commercial

It is an automated business process management/workflow and customer management system designed to improve productivity, enhance customer loyalty and reduce operating costs.

AWD applies your pre-defined Business Processes and your associates' Skill Inventory to the execution of your Work Inventory. AWD also stores all Customer Information, so you can frame future interactions in the context of prior contacts.

With AWD, all the knowledge you need to manage your business and service your customers is available to the right people at the right time.

A 13.2.2.22 Meta4 KnowNet

Developer: Meta4

URL: http://www.meta4.com

License: Commercial

KnowNet is the complete people and knowledge management solution, comprising a software solution and its associated strategic consulting services. It is designed to capitalize on the knowledge generated around business processes, to facilitate its distribution and reuse, to increase speed of learning and to provide practical management of knowledge in the company. It can identify the knowledge required for a given task and where or from whom it can be obtained. It locates experts, creates user communities and helps plans future training or know-how acquisition on detected needs.

A 13.2.2.23 Microsoft NetMeeting

Developer: Microsoft

URL: http://www.microsoft.com/windows/NetMeeting/default.ASP

License: Free

NetMeeting 3 is included in Windows 2000. It delivers a complete Internet conferencing solution for all Windows users with multi-point data conferencing, text chat, whiteboard, and file transfer, as well as point-to-point audio and video.

A 13.2.2.24 Teamware Office

Developer: Teamware

URL: http://www.teamware.com/

License: Commercial

Teamware Office is easy to install and the users can immediately start taking advantage of Teamware Office services:

- Secure e-mail by Teamware Mail featuring the integrated Secure Mail functionality
- Team and resource scheduling with Teamware Calendar, now providing To-Do lists through WebService and built-in PDA Calendar synchronization via SMS Connector
- Discussion forums on Teamware Forum
- Global document libraries in Teamware Library

The built-in WebService provides the transition from the traditional Windows intranet to the future-proof, web browser based, open internet environment.

A 13.2.2.25 Verity K2 Enterprise

Developer: Verity

URL: http://www.verity.com/products/enterprise/index.html

License: Commercial

Verity K2 Enterprise three-tier infrastructure integrates comprehensive search tools to discover information, accurate content organization to put information in context, and personalization features to automatically recommend documents, locate experts and create user communities. These three tiers support your knowledge workers with the power to organize, discover and connect people with information and each other.

Verity K2 Enterprise leverages your current technology investments so you can realize the true value of your knowledge assets—your people, content, technology and the relationships between them. To cure the tunnel vision that individual applications such as content management, ERP and CRM systems create, K2 Enterprise's search and classification tiers put all of your enterprise content into context so it's searchable with a single query and browseable through a common taxonomy. And its patent-pending social network tier uses the relationships created when people look for information to recommend relevant documents, locate subject matter experts and bring together communities of interest from across the enterprise. Most importantly, K2 Enterprise does all this while enforcing your existing security models and adapting to the way you do business—not the other way around.

By delivering better knowledge management and better return on assets, Verity K2 Enterprise ensures that you and your employees can make better, faster and more profitable business decisions. That's why over 1500 businesses—including nine of the top 10, 80% of the top 50 and 66% of the top 100 Fortune 500 companies—rely on market-leading infrastructure software from Verity for better knowledge management and better return on assets. So they can make better decisions and stay ahead of their competition.

A 13.2.2.26 Livelink: Personal Workspace

Developer: Open Text Corporation

URL: http://www.opentext.net/livelink/details/workspace_personal.html

License: Commercial

The Personal Workspace provides a single point of access to your intranet, extranet, Livelink news feeds and real-time business intelligence.

This workspace provides individuals with the ability to create a highly personalized workspace, optimized for productivity and innovation. By allowing you to organize your information and your thoughts in a way that is intuitive to you, you can access information when you need it and arrive at conclusions faster. In the Personal Workspace you can subscribe to information sources, organize your schedule and task lists, monitor Livelink Change Agents, or have information pushed to your desktop with Livelink Channels.

A 13.2.2.27 Teamware Pl@za

Developer: Teamware

URL: http://www.teamware.com/

License: Commercial

Teamware Pl@za is used for building interactive intra/extra/Internet solutions. It consists of a technology platform that can be seamlessly integrated to existing systems, and ready-made services that can be customized according to customer requirements.

By utilizing Teamware Pl@za you can create an interactive portal solution including personalized workspaces, discussion groups, document management, professional e-mail and calendar, user updateable pages with well defined access rights and lots more.

A 13.2.2.28 ProSim

Developer: Knowledge Based Systems, Inc.

URL: http://www.kbsi.com/Software/Prosim.htm

License: Commercial

ProSim is a simulation tool which allows you to

- Create animated simulations and virtual reality visualizations of your workflow in WITNESS
- Publish your knowledge base on the World Wide Web
- Index into distributed corporate information sources with process knowledge map
- Package and distribute standardized process knowledge with the ProSim Viewer
- Show cross organization processes with swim lanes

A 13.2.2.29 ADONIS

Developer: BOC Information Technologies Consulting GmbH

URL: http://www.microtool.de/case40/en/sp_case_ado.htm

License: Commercial

ADONIS toolkit offers components for the acquisition, modeling, analysis, simulation, evaluation, transformation and documentation of business processes. Adonis's most outstanding features are its powerful simulation library and the extensive customizing possibilities it offers. The latter enable the user to configure ADONIS in such a way that it is ideally suited to the requirements of a given application without any additional programming effort. Among other things, the metamodel on which the modeling is based can be defined freely during customizing. ADONIS is used company-wide by several big European companies as a strategic control instrument for business processes.

A 13.3 Assessment

Some of the features of the Knowledge Management are considered in the initial proposal of ALFANET toolkit in the Technical Annex. Collaboration is included as one of the main features of the ALFANET e-learning platform. So ALFANET consortium should evaluate the possibility of using some KM collaboration tools within ALFANET system instead of building a new collaborative environment.

Other important issue to take into acount is that it will be very useful that e-learning contents will be stored in such a format, complaint with standards, that could be used by KM tools for content management. In this ways both e-learning platforms and KM tools could use the same contents.

The integration of ALFANET platform with KM tools could provide the users with information and contents available in distributed repositories outside the e-learning application. Other possible relationship of KM with authoring tools is helping on capturing tacit knowledge into explicit knowledge.

The integration of ALFANET platform with HR tools which cover different aspects of the human resuources management, such as administration, succession planning, leadership development, competency management and organizational charting, should be also considered as an ALFANET feature. This integration could be achieved by XML communitacion.

A 13.4 References

[1] http://www.km-forum.org/what_is.htm

[2] http://www.bus.utexas.edu/kman/answers.htm

[3] http://www.knowledgeboard.com/

[4] http://www.totalkm.com/corpkmtools.html

[5] http://www.kmtool.net/

[6] http://www.gestiondelconocimiento.com/software.htm

Appendix 14 Privacy and Security Concepts

Security and privacy are among the most central aspects and quality criteria for a system whose functionality is based on storage and processing of sensitive user data. Of course, legal issues are relevant here, but psychological (and ethical) aspects cannot be neglected.

This appendix has two parts, the first part reviews approaches to privacy issues and gives references to related work. The second reviews security concepts that privacy relies upon, including technologies and standards that can be used to develop and implement a security concept for ALFANET.

A 14.1 Privacy

ALFANET raises a number of privacy issues. First of all, legal requirements must be complied with. But further, the system can only expect user acceptance and confidence if it convinces every party to be treated securely. Privacy guidelines are observed, whether they are dictated by law, ethics or by usability.

A 14.1.1 Legal aspects

A legal background, the system should consider at least two significant regulatory regimes:

- the European Commission's Data Directive (1995), operational in March 2001, and
- the U.S. Federal Trade Commission's (FTC) Fair Practice Guidelines (2000).

Privacy laws regulate the kinds of protection that personal data must receive, and the rights that subjects enjoy with regard to personal data about them. Data may usually be collected for specific purposes only, and only those personal data may be collected that are necessary for the indicated purposes (principle of parsimony). They may not be stored longer than is necessary for these purposes, and not further processed or given to third parties in a way incompatible with those purposes. A system that processes personal data must usually implement appropriate technical and organizational measures to protect these data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access. Additional restrictions sometimes exist for very sensitive data (e.g., racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and data concerning health or sex life). Except for very sensitive data, most protection requirements can usually be waived with the consent of the user.

These legislations articulate four basic areas that need to be taken into account to ensure that providers of information some guarantee of privacy [1]:

- Notice – individuals should have clear notice of the type of information collected, its use, and an indication of third parties other than the original collector who will have access to the data.
- Choice – individuals should be able to choose not to have data collected.
- Access - the data subject should be able to see what personal information is held about him or her, to inspect data about themselves, and request blocking, rectification and erasure in case they are incorrect or obsolete, or processed in violation of a privacy law if desired.
- Security - reasonable measures should be taken to secure (both technically and operationally) the data from unauthorized access.

A 14.1.2 Feature considerations

A 14.1.2.1 User types

Some studies [2] surveyed a number of users to establish levels of privacy concern in the U.S. They identified three main groups: 'fundamentalists', 'pragmatists' and 'unconcerneds'. They concluded that there will be considerable variation in the rules that people wish to have to govern perceptual contexts and privacy as 'one man's privacy is another person's spy…one person's contextual awareness is another person's lack of privacy'.

A 14.1.2.2 User tracking

'Activity awareness' is a feature of systems that track users' interaction in ALFANET. It is clear that a balance must be struck between supporting such activity unobtrusively, and making the use aware of the privacy implications of the activity.

The tradeoffs that must be made by designers of interfaces for systems that must comply with privacy regimes have been discussed for a number of years [4]. Complying with the requirement for notice can be overwhelming for users who may care about privacy, but have a low tolerance from being diverted from the primary task at hand to consider the privacy content of a given transaction. It suggest that basic user interface mechanisms are needed for 'unobtrusive notice'. [5]

Interface designers and development teams must consider who will use interfaces, and what they will be trusted (or not) to do. So it is important to minimize assumptions about trust between components in a multiparty system by specifying explicitly what or who is trustworthy. [6]

Tracking a user's Internet activities is inherently invisible act, and not all extensions make user aware of the privacy issues involved. An study by the University of Denver Privacy Center that considered the privacy implications of Internet Explorer extensions [7] attempted to reconcile observed behavior with the stated privacy policy and licensing agreement of software vendors. This study identified a number of areas where privacy might be compromised: Problems, such as lack of candor, loopholes, poor placement (information on privacy policy was tucked away in a corner), technical jargon and legalese, and perspective mismatch – short-term storage also requires notice, were observed with notice/awareness.

This study also observed what Web site privacy seals protect web site practices only –at the time of writing, there was no specification for communicating the privacy practices of downloaded software to P3P user agents [11]. It concludes that: "It is time to elevate privacy practices 'to first-class criteria that discerning consumers will count along with speed, memory consumption, and ease of use in the search for the perfect tool for the job".

A 14.1.2.3 Information sharing

The delicate balance between protecting user privacy, and the facilitating sharing of information is another important issue consired in some studies [8]. Designers must provide users with the means of specifying their own individual privacy policies – at the time of writing, the design of an interface for such mechanisms had received scant attention. The burden was on users to specify a policy for each object when a more generic application was required in the form of specifications for a set of privacy interfaces that facilitate the creation, inspection, modification and monitoring of privacy policies. CollabClio is a prototype interface that stores a person's browsing history and makes it searchable by content, keyword and other attributes. An ideal privacy interface "must make it easy to create, inspect, modify and monitor privacy policies"; privacy policies themselves should be proactive – that is they should apply to objects are they are encountered.

Two different ways of representing a privacy policy are considered by some authors [8]. The first is extensional (they offer the privacy policies of Firefly and Passport as examples, and suggest that these have scaling problems) – the interface creates a list that enumerates all the items in a set (this was done by means of a record light in a prototype). The second is intensional – this describes a set by characterizing all the objects in the set in a declarative statement. In a modified version of the prototype, a privacy policy editor window (to support the creation, inspection and modification of privacy policies), and monitor and query-log windows to allow users to see the effects of a policy have been created.

A 14.1.2.4 User profile sharing

A comprehensive review of privacy issues where user profiles must be supported across different communities is presented in some studies [9]. Knowing the identity of those with whom you communicate is essential for understanding and evaluating an interaction [10]. The main issue is the 'cold-start' problem [9], and this may be solved by the design of a platform for using user profiles in more than one application. They define 'identity management' as everyday decisions about what to tell one another about ourselves, a pressing requirement where different sets of information are released to different interaction partners – aliases, pseudonyms etc. An ideal identity management system should thus:

- Allow people to define different identities, roles etc
- Associate personal data to these
- Decide when to give data and when to act anonymously
- Maintain privacy and control (see other paper on 'control' as privacy)

- Make it easy for user to use different communities and thereby lower the entry barrier to online communities.

A 14.1.2.5 User modelling

User modelling has mainly been concerned with the adaptation of software systems to the user. With the raise of the WWW and the growing dissemination of user modeling techniques and systems into Internet and WWW applications, new problems appear. Acquisition, processing, and storage of personal data in an open and shared system like the Internet requires an intensified consideration of user demands to security, privacy, and anonymity. Since personalized websites collect personal data, they have to abide to relevant privacy laws. The stipulations of privacy laws already have far-reaching impacts on personalized systems. In most cases they imply that users must be notified about and consent to personalization, and that their user model must be made accessible to them. [3] [14]

Most implemented systems that employ a user model work on a local computing base. Information gathered about the user of the system resides locally on this base and is often encoded in the application maintaining the model. By that means, unintended but effective restrictions to the dissemination of items of the user model have been established. The trend in the design of new user modeling systems to (network) connectivity and standardized content abolishes these immanent borders.

Data about the usage of specific systems that are related to an identifiable person have to be treated in a special manner, and legal constraints must be taken into account. The inclusion of sensitive data usually restricts the treatment to a rigid range of utilization. Accepting this challenge entails various extensions that have to be added to current user modeling systems.

A 14.1.3 Related Projects

A 14.1.3.1 P3P: the Platform for Privacy Preferences Project

The most recent work on privacy policies for systems that will be used in multiple social contexts and supported by multiple platforms has focused on P3P[11], the Preference Exchange Language supported by W3C [1] [9] [12]. In its earliest drafts, P3P required every user to store requirements for the privacy practice of an Internet service in the form of formalized preferences given to the user browser. Every Internet service would send its privacy preferences to the user in the form of formalized proposals. The Internet service would offer one or more policies, and the user would select one or offer a counterproposal, negotiation would continue until agreement was reached, or the session terminated. The earliest version of P3P describes three different areas [12]:

- privacy protection policies as a set of formalized statements
- the decision-making process of the user
- a protocol for the transfer of personal data in the form of data fields.

This version was held to be too demanding and the next version retained only the formalization of the description of a privacy policy in terms of a ‘notice’ by the provider and a ‘choice’ by the user. A specific language (APPEL) has been developed. The P3P specification defines a vocabulary for describing the data practices of a service and the user agent can check the conformity of the privacy policy of a community with the user’s privacy preferences. APPEL (the Preference Exchange Language) describes four standard actions: accept, reject, inform, and warn [13].

A 14.1.3.2 Cobricks

Cobricks project [9], an identity management system whose basic functions are to create, store and access digital identities (user profiles). The owner of these is the user who provides the information not the service using the information. Identities can be created by the user himself or by a certification authority. The system provides specifications for the use of digital identities in three contexts:

- Authentication (or the provision of data for different functionalities – transactions, personalization, configuration);
- Control (that allows users to create different identities for different occasions and define and negotiate special access rights to one identity);
- Privacy (the system has to specify which services can access and write which data).

The following requirements for a privacy infrastructure are suggested:

- Flexible access right control system (rules and negotiation)
- Possibility to monitor access rights and accesses
- Possibility to use a pseudonym instead of real identity
- Purpose binding of a data accesses
- Possibility to allow access for temporary use
- Possibility to revoke granted access rights
- Control whether user data can be distribute to other services
- Integration of cryptographic techniques for anonymous data transfers
- Possibility of support from privacy authorities

A number of specific issues must be addressed in the design of identity management systems. These include the structure of profiles (P3P, or example uses 'hierarchically structured sets of attribute value pairs'. These might be "basic demographic attributes, but might also draw on information and interests and correlations with predefined clusters or stereotype".

Ratings may be given by a user to information (implicit by visit or explicit) and information provided about relationships /networks – colleagues, buddies. In addition, the system must contain meta information about who has stored the data (the source might sign the data or an identity management system guarantees the source of the data; in Cobricks: XML based schemas are provided for defining the ontologies; schemas are used in the web interface for dynamically creating the user interface for entering information, and for entering meta information). Users do not have to formulate privacy preferences themselves; they can adjust them however, and require a 'suggestive' user interface for this.

A 14.2 Security

Privacy is, however not possible without an underlying layer of security functionalities as a "technological" basis, which must protect the data to be searched and navigated from privacy infringements.

For this purpose, ALFANET will design a concept and implement an information access system that allows efficient access to retrieve relevant data and combines advanced querying and navigational features, based on user profile and security data.

A 14.2.1 Internet security issues

All communication over the Internet uses the Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP allows information to be sent from one computer to another through a variety of intermediate computers and separate networks before it reaches its destination.

The great flexibility of TCP/IP has led to its worldwide acceptance as the basic Internet and intranet communications protocol. At the same time, the fact that TCP/IP allows information to pass through intermediate computers makes it possible for a third party to interfere with communications in the following ways:

- **Eavesdropping**. Information remains intact, but its privacy is compromised. For example, someone could learn your credit card number, record a sensitive conversation, or intercept classified information.
- **Tampering**. Information in transit is changed or replaced and then sent on to the recipient. For example, someone could alter an order for goods or change a person's resume.
- **Impersonation**. Information passes to a person who poses as the intended recipient. Impersonation can take two forms:
    - **Spoofing**. A person can pretend to be someone else. For example, a person can pretend to have the email address abcdef@mozilla.com, or a computer can identify itself as a site called www.mozilla.com when it is not. This type of impersonation is known as spoofing.

- **Misrepresentation**. A person or organisation can misrepresent itself. For example, suppose the site www.mozilla.com pretends to be a furniture store when it is really just a site that takes credit-card payments but never sends any goods.

Normally, users of the many cooperating computers that make up the Internet or other networks don't monitor or interfere with the network traffic that continuously passes through their machines. However, many sensitive personal and business communications over the Internet require precautions that address the threats listed above. Fortunately, a set of well-established techniques and standards known as **public-key cryptography** make it relatively easy to take such precautions [17].

Public-key cryptography facilitates the following tasks:

- **Encryption and decryption** allow two communicating parties to disguise information they send to each other. The sender encrypts, or scrambles, information before sending it. The receiver decrypts, or unscrambles, the information after receiving it. While in transit, the encrypted information is unintelligible to an intruder.
- **Tamper detection** allows the recipient of information to verify that it has not been modified in transit. Any attempt to modify data or substitute a false message for a legitimate one will be detected.
- **Authentication** allows the recipient of information to determine its origin--that is, to confirm the sender's identity.
- **Nonrepudiation** prevents the sender of information from claiming at a later date that the information was never sent.

The sections that follow introduce the concepts of public-key cryptography that underlie these capabilities.

A 14.2.2 Solutions

A 14.2.2.1 Encryption and Decryption

Encryption is the process of transforming information so it is unintelligible to anyone but the intended recipient. Decryption is the process of transforming encrypted information so that it is intelligible again. A **cryptographic algorithm,** also called a **cipher,** is a mathematical function used for encryption or decryption. In most cases, two related functions are employed, one for encryption and the other for decryption.

With most modern cryptography, the ability to keep encrypted information secret is based not on the cryptographic algorithm, which is widely known, but on a number called a **key** that must be used with the algorithm to produce an encrypted result or to decrypt previously encrypted information. Decryption with the correct key is simple. Decryption without the correct key is very difficult, and in some cases impossible for all practical purposes [17].

The use of encryption/decryption is as old as the art of communication. In wartime, a cipher, often incorrectly called a "code," can be employed to keep the enemy from obtaining the contents of transmissions (technically, a *code* is a means of representing a signal without the intent of keeping it secret; examples are Morse code and ASCII). Simple ciphers include the substitution of letters for numbers, the rotation of letters in the alphabet, and the "scrambling" of voice signals by inverting the sideband frequencies. More complex ciphers work according to sophisticated computer algorithms that rearrange the data bits in digital signals.

In order to easily recover the contents of an encrypted signal, the correct *decryption key* is required. The key is an algorithm that "undoes" the work of the encryption algorithm. The more complex the encryption algorithm, the more difficult it becomes to eavesdrop on the communications without access to the key.

Encryption/decryption is especially important in wireless communications. This is because wireless circuits are easier to "tap" than their hard-wired counterparts. Nevertheless, encryption/decryption is a good idea when carrying out any kind of sensitive transaction, such as a credit-card purchase online, or the discussion of a company secret between different departments in the organisation. The stronger the cipher (that is, the harder it is for unauthorized people to break it) the better, in general. However, as the strength of encryption/decryption increases, so does the cost.

Following we introduce the use of keys for encryption and decryption: Symmetric-Key encryption and Public-Key encryption.

With symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a

degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

Symmetric-key encryption is effective only if the symmetric key is kept secret by the two parties involved. If anyone else discovers the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

Symmetric-key encryption plays an important role in the SSL protocol, which is widely used for authentication, tamper detection, and encryption over TCP/IP networks. SSL also uses techniques of public-key encryption, which is described in the next section.

The most commonly used implementations of public-key encryption are based on algorithms patented by RSA Data Security. Therefore, this section describes the RSA approach to public-key encryption.

Public-key encryption (also called asymmetric encryption) involves a pair of keys--a public key and a private key--associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published, and the corresponding private key is kept secret. Data encrypted with your public key can be decrypted only with your private key.

In this case, you can freely distribute a public key, and only you will be able to read data encrypted using this key. In general, to send encrypted data to someone, you encrypt the data with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding private key.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, it's possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL protocol.

As it happens, the reverse of the scheme also works: data encrypted with your private key can be decrypted only with your public key. This would not be a desirable way to encrypt sensitive data, however, because it means that anyone with your public key, which is by definition published, could decrypt the data. Nevertheless, private-key encryption is useful, because it means you can use your private key to sign data with your digital signature--an important requirement for electronic commerce and other commercial applications of cryptography. Client software such as Communicator can then use your public key to confirm that the message was signed with your private key and that it hasn't been tampered with since being signed. Ditigal Signatures and subsequent sections describe how this confirmation process works.

In general, the strength of encryption is related to the difficulty of discovering the key, which in turn depends on both the cipher used and the length of the key. For example, the difficulty of discovering the key for the RSA cipher most commonly used for public-key encryption depends on the difficulty of factoring large numbers, a well-known mathematical problem.

Encryption strength is often described in terms of the size of the keys used to perform the encryption: in general, longer keys provide stronger encryption. Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. Roughly speaking, 128-bit RC4 encryption is 3 x 1026 times stronger than 40-bit RC4 encryption.

Different ciphers may require different key lengths to achieve the same level of encryption strength. The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based. Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values. Thus a 128-bit key for use with a symmetric-key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

This difference explains why the RSA public-key encryption cipher must use a 512-bit key (or longer) to be considered cryptographically strong, whereas symmetric key ciphers can achieve approximately the same level of strength with a 64-bit key. Even this level of strength may be vulnerable to attacks in the near future.

Because the ability to surreptitiously intercept and decrypt encrypted information has historically been a significant military asset, the U.S. Government restricts export of cryptographic software, including most software that permits use of symmetric encryption keys longer than 40 bits.

In recent years, a controversy has arisen over so-called *strong encryption*. This refers to ciphers that are essentially unbreakable without the decryption keys. While most companies and their customers view it as a

means of keeping secrets and minimizing fraud, some governments view strong encryption as a potential vehicle by which terrorists might evade authorities. These governments, including that of the United States, want to set up a *key-escrow* arrangement. This means everyone who uses a cipher would be required to provide the government with a copy of the key. Decryption keys would be stored in a supposedly secure place, used only by authorities, and used only if backed up by a court order. Opponents of this scheme argue that criminals could hack into the key-escrow database and illegally obtain, steal, or alter the keys. Supporters claim that while this is a possibility, implementing the key escrow scheme would be better than doing nothing to prevent criminals from freely using encryption/decryption.

A 14.2.2.2 Digital Signatures

Encryption and decryption address the problem of eavesdropping, one of the three Internet security issues mentioned at the beginning of this document. But encryption and decryption, by themselves, do not address the other two problems mentioned in Internte Security Issues: tampering and impersonation.

This section describes how public-key cryptography addresses the problem of tampering. The sections that follow describe how it addresses the problem of impersonation.

Tamper detection and related authentication techniques rely on a mathematical function called a **one-way hash** (also called a **message digest**). A one-way hash is a number of fixed length with the following characteristics:

- The value of the hash is unique for the hashed data. Any change in the data, even deleting or altering a single character, results in a different value.
- The content of the hashed data cannot, for all practical purposes, be deduced from the hash--which is why it is called "one-way."

As mentioned in Public-Key Encryption, it's possible to use your private key for encryption and your public key for decryption. Although this is not desirable when you are encrypting sensitive information, it is a crucial part of digitally signing any data. Instead of encrypting the data itself, the signing software creates a one-way hash of the data, then uses your private key to encrypt the hash. The encrypted hash, along with other information, such as the hashing algorithm, is known as a **digital signature.**

In this case, two items are transferred to the recipient of some signed data: the original data and the digital signature, which is basically a one-way hash (of the original data) that has been encrypted with the signer's private key. To validate the integrity of the data, the receiving software first uses the signer's public key to decrypt the hash. It then uses the same hashing algorithm that generated the original hash to generate a new one-way hash of the same data. (Information about the hashing algorithm used is sent with the digital signature, although this isn't shown in the figure.) Finally, the receiving software compares the new hash against the original hash. If the two hashes match, the data has not changed since it was signed. If they don't match, the data may have been tampered with since it was signed, or the signature may have been created with a private key that doesn't correspond to the public key presented by the signer.

If the two hashes match, the recipient can be certain that the public key used to decrypt the digital signature corresponds to the private key used to create the digital signature. Confirming the identity of the signer, however, also requires some way of confirming that the public key really belongs to a particular person or other entity.

The significance of a digital signature is comparable to the significance of a handwritten signature. Once you have signed some data, it is difficult to deny doing so later--assuming that the private key has not been compromised or out of the owner's control. This quality of digital signatures provides a high degree of nonrepudiation--that is, digital signatures make it difficult for the signer to deny having signed the data. In some situations, a digital signature may be as legally binding as a handwriten signature.

A 14.2.2.3 Certificates and Authentication

A **certificate** is an electronic document used to identify an individual, a server, a company, or some other entity and to associate that identity with a public key. Like a driver's license, a passport, or other commonly used personal IDs, a certificate provides generally recognized proof of a person's identity. Public-key cryptography uses certificates to address the problem of impersonation.

To get a driver's license, you typically apply to a government agency, such as the Department of Motor Vehicles, which verifies your identity, your ability to drive, your address, and other information before issuing the license. To get a student ID, you apply to a school or college, which performs different checks (such as

whether you have paid your tuition) before issuing the ID. To get a library card, you may need to provide only your name and a utility bill with your address on it.

Certificates work much the same way as any of these familiar forms of identification. **Certificate authorities (CAs)** are entities that validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software (such as Netscape Certificate Server). The methods used to validate an identity vary depending on the policies of a given CA--just as the methods to validate other forms of identification vary depending on who is issuing the ID and the purpose for which it will be used. In general, before issuing a certificate, the CA must use its published verification procedures for that type of certificate to ensure that an entity requesting a certificate is in fact who it claims to be.

The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies (such as the name of an employee or a server). Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate.

In addition to a public key, a certificate always includes the name of the entity it identifies, an expiration date, the name of the CA that issued the certificate, a serial number, and other information. Most importantly, a certificate always includes the digital signature of the issuing CA. The CA's digital signature allows the certificate to function as a "letter of introduction" for users who know and trust the CA but don't know the entity identified by the certificate.

Authentication is the process of confirming an identity. In the context of network interactions, authentication involves the confident identification of one party by another party. Authentication over networks can take many forms. Certificates are one way of supporting authentication.

Network interactions typically take place between a client, such as browser software running on a personal computer, and a server**,** such as the software and hardware used to host a Web site. **Client authentication** refers to the confident identification of a client by a server (that is, identification of the person assumed to be using the client software). **Server authentication** refers to the confident identification of a server by a client (that is, identification of the organization assumed to be responsible for the server at a particular network address).

Client and server authentication are not the only forms of authentication that certificates support. For example, the digital signature on an email message, combined with the certificate that identifies the sender, provide strong evidence that the person identified by that certificate did indeed send that message. Similarly, a digital signature on an HTML form, combined with a certificate that identifies the signer, can provide evidence, after the fact, that the person identified by that certificate did agree to the contents of the form. In addition to authentication, the digital signature in both cases ensures a degree of nonrepudiation--that is, a digital signature makes it difficult for the signer to claim later not to have sent the email or the form.

Client authentication is an essential element of network security within most intranets or extranets. There are two forms of client authentication:

- Password-Based Authentication. Almost all server software permits client authentication by means of a name and password. For example, a server might require a user to type a name and password before granting access to the server. The server maintains a list of names and passwords; if a particular name is on the list, and if the user types the correct password, the server grants access.
- Certificate-Based Authentication. Client authentication based on certificates is part of the SSL protocol. The client digitally signs a randomly generated piece of data and sends both the certificate and the signed data across the network. The server uses techniques of public-key cryptography to validate the signature and confirm the validity of the certificate.

The contents of certificates supported by Netscape and many other software companies are organized according to the X.509 v3 certificate specification, which has been recommended by the International Telecommunications Union (ITU), an international standards body, since 1988.

Users don't usually need to be concerned about the exact contents of a certificate. However, system administrators working with certificates may need some familiarity with this information.

Certificate authorities (CAs) are entities that validate identities and issue certificates. They can be either independent third parties or organizations running their own certificate-issuing server software (such as the Netscape Certificate Server). Some third-party certificate authorities are:

- VeriSign. VeriSign is the leading provider of digital authentication products and services. The first commercial CA, VeriSign has issued Digital IDs for almost every secure Internet server

worldwide. Strict verification and security practices, enforced through automated background checks and state-of-the-art security systems, ensure the integrity of every VeriSign Digital ID.

- Thawte Consulting. Thawte has representatives in 20 countries, providing first-class local support and service. We offer personal certificates for SSL, S/MIME, and Netscape code signing.
- Società per i Servizi Bancari – SSB S.p.A. The Trusted Certification Authority. SSB provides highly secured X.509 v1 v3 certificates on behalf of banks for Internet clients and servers, financial services, and electronic commerce.
- Internet Publishing Services. IPS provides server, client, and object-signing certificates based on SSL standards.
- Certisign Certification Digital Ltda. The Brazilian Certificate Authority. With strong identity-checking procedures, Certisign issues only high-assurance X.509 digital IDs for SSL-compliant servers and clients.
- BelSign. BelSign International, with local registration offices across Europe, provides a range of digital certificates to Internet clients and servers based on strict verification practices.

Any client or server software that supports certificates maintains a collection of **trusted CA certificates.** These CA certificates determine which other certificates the software can validate--in other words, which issuers of certificates the software can trust. In the simplest case, the software can validate only certificates issued by one of the CAs for which it has a certificate. It's also possible for a trusted CA certificate to be part of a chain of CA certificates, each issued by the CA above it in a certificate hierarchy.

A 14.2.2.4 Public Key Infrastructure

The set of standards and services that facilitate the use of public-key cryptography and X.509 v3 certificates in a networked environment is called the **public key infrastructure (PKI)** [17]. PKI management is complex topic beyond the scope of this document. The sections that follow introduce some of the specific certificate management issues addressed by Netscape products:

- Issuing certificates
- Cerfificates and the LDAP Directory
- Key Management
- Renewing and Revoking Certificates
- Registration Authorities

14.2.2.4.1 Issuing Certificates

The process for issuing a certificate depends on the certificate authority that issues it and the purpose for which it will be used. The process for issuing nondigital forms of identification varies in similar ways. For example, if you want to get a generic ID card (not a driver's license) from the Department of Motor Vehicles in California, the requirements are straightforward: you need to present some evidence of your identity, such as a utility bill with your address on it and a student identity card. If you want to get a regular driving license, you also need to take a test--a driving test when you first get the license, and a written test when you renew it. If you want to get a commercial license for an eighteen-wheeler, the requirements are much more stringent. If you live in some other state or country, the requirements for various kinds of licenses will differ.

Similarly, different CAs have different procedures for issuing different kinds of certificates. In some cases the only requirement may be your email address. In other cases, your Unix or NT login and password may be sufficient. At the other end of the scale, for certificates that identify people who can authorize large expenditures or make other sensitive decisions, the issuing process may require notarized documents, a background check, and a personal interview.

Depending on an organization's policies, the process of issuing certificates can range from being completely transparent for the user to requiring significant user participation and complex procedures. In general, processes for issuing certificates should be highly flexible, so organizations can tailor them to their changing needs.

Issuing certificates is one of several managements tasks that can be handled by separate Registration Authorities.

14.2.2.4.2 Certificates and the LDAP Directory

The Lightweight Directory Access Protocol (LDAP) for accessing directory services supports great flexibility in the management of certificates within an organization. System administrators can store much of the

information required to manage certificates in an LDAP-compliant directory. For example, a CA can use information in a directory to prepopulate a certificate with a new employee's legal name and other information. The CA can leverage directory information in other ways to issue certificates one at a time or in bulk, using a range of different identification techniques depending on the security policies of a given organization. Other routine management tasks, such as Key Management and Renewing and Revoking Certificates, can be partially or fully automated with the aid of the directory.

Information stored in the directory can also be used with certificates to control access to various network resources by different users or groups. Issuing certificates and other certificate management tasks can thus be an integral part of user and group management.

In general, high-performance directory services are an essential ingredient of any certificate management strategy.

14.2.2.4.3 Key Management

Before a certificate can be issued, the public key it contains and the corresponding private key must be generated. Sometimes it may be useful to issue a single person one certificate and key pair for signing operations, and another certificate and key pair for encryption operations. Separate signing and encryption certificates make it possible to keep the private signing key on the local machine only, thus providing maximum nonrepudiation, and to back up the private encryption key in some central location where it can be retrieved in case the user loses the original key or leaves the company.

Keys can be generated by client software or generated centrally by the CA and distributed to users via an LDAP directory. There are trade-offs involved in choosing between local and centralized key generation. For example, local key generation provides maximum nonrepudiation, but may involve more participation by the user in the issuing process. Flexible key management capabilities are essential for most organizations.

Key recovery, or the ability to retrieve backups of encryption keys under carefully defined conditions, can be a crucial part of certificate management (depending on how an organization uses certificates). Key recovery schemes usually involve an **m of n** mechanism: for example, *m* of *n* managers within an organization might have to agree, and each contribute a special code or key of their own, before a particular person's encryption key can be recovered. This kind of mechanism ensures that several authorized personnel must agree before an encryption key can be recovered.

14.2.2.4.4 Renewing and Revoking Certificates

Like a driver's license, a certificate specifies a period of time during which it is valid. Attempts to use a certificate for authentication before or after its validity period will fail. Therefore, mechanisms for managing certificate renewal are essential for any certificate management strategy. For example, an administrator may wish to be notified automatically when a certificate is about to expire, so that an appropriate renewal process can be completed in plenty of time without causing the certificate's subject any inconvenience. The renewal process may involve reusing the same public-private key pair or issuing a new one.

A driver's license can be suspended even if it has not expired--for example, as punishment for a serious driving offense. Similarly, it's sometimes necessary to revoke a certificate before it has expired--for example, if an employee leaves a company or moves to a new job within the company.

Certificate revocation can be handled in several different ways. For some organizations, it may be sufficient to set up servers so that the authentication process includes checking the directory for the presence of the certificate being presented. When an administrator revokes a certificate, the certificate can be automatically removed from the directory, and subsequent authentication attempts with that certificate will fail even though the certificate remains valid in every other respect. Another approach involves publishing a **certificate revocation list (CRL)**--that is, a list of revoked certificates--to the directory at regular intervals and checking the list as part of the authentication process. For some organizations, it may be preferable to check directly with the issuing CA each time a certificate is presented for authentication. This procedure is sometimes called **real-time status checking.**

14.2.2.4.5 Registration Authorities

Interactions between entities identified by certificates (sometimes called **end entities**) and CAs are an essential part of certificate management. These interactions include operations such as registration for certification, certificate retrieval, certificate renewal, certificate revocation, and key backup and recovery. In general, a CA must be able to authenticate the identities of end entities before responding to the requests. In addition, some requests need to be approved by authorized administrators or managers before being services.

As previously discussed, the means used by different CAs to verify an identity before issuing a certificate can vary widely, depending on the organization and the purpose for which the certificate will be used. To provide maximum operational flexibility, interactions with end entities can be separated from the other functions of a CA and handled by a separate service called a **Registration Authority (RA).**

An RA acts as a front end to a CA by receiving end entity requests, authenticating them, and forwarding them to the CA. After receiving a response from the CA, the RA notifies the end entity of the results. RAs can be helpful in scaling an PKI across different departments, geographical areas, or other operational units with varying policies and authentication requirements.

A 14.2.2.5 Virtual Private Networking

A virtual private network (VPN) is a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of a tunnelling protocol and security procedures. A virtual private network can be contrasted with a system of owned or leased lines that can only be used by one company. The idea of the VPN is to give the company the same capabilities at much lower cost by using the shared public infrastructure rather than a private one. Phone companies have provided secure shared resources for voice messages. A virtual private network makes it possible to have the same secure sharing of public resources for data. Companies today are looking at using a private virtual network for both extranetsand wide-area intranets.

Using a virtual private network involves encrypting data before sending it through the public network and decrypting it at the receiving end. An additional level of security involves encrypting not only the data but also the originating and receiving network addresses.

The protocol or set of communication rules called Point-to-Point Tunnelling Protocol (PPTP) has been proposed that would make it possible to create a virtual private network through "tunnels" over the Internet. This would mean that companies would no longer need their own leased lines for wide-area communication but could securely use the public networks.

PPTP, sponsored by Microsoft and other companies, and Layer 2 Forwarding (L2F), proposed by Cisco Systems, are among the main proposals for a new Internet Engineering Task Force (IETF) standard. With PPTP, which is an extension of the Internet's Point-to-Point Protocol (PPP), any user of a PC with PPP client support will be able to use an independent service provider (ISP) to connect securely to a server elsewhere in the user's company. Tunnelling is the transmission of data intended for use only within a private, usually corporate network through a public network in such a way that the routing nodes in the public network are unaware that the transmission is part of a private network. Tunnelling is generally done by encapsulating the private network data and protocol information within the public network transmission units so that the private network protocol information appears to the public network as data. Tunnelling allows the use of the Internet, which is a public network, to convey data on behalf of a private network.

Tunnelling, and the use of a VPN, is not intended as a substitute for encryption/decryption. In cases where a high level of security is necessary, the strongest possible encryption should be used within the VPN itself, and tunnelling should serve only as a convenience.

A 14.2.2.6 Secure WWW connections

It is highly desirable that Internet carriers protect the privacy and authenticity of all traffic, but this is not a requirement of the architecture. Confidentiality and authentication are the responsibility of end users and must be implemented in the protocols used by the end users. Endpoints should not depend on the confidentiality or integrity of the carriers. Carriers may choose to provide some level of protection, but this is secondary to the primary responsibility of the end users to protect themselves.

WWW security has become important as increasing amounts of sensitive information, such as credit card numbers, are being transmitted over the Internet.

SSL and SHTTP are protocols for secure WWW connections:

Secure Socket Layer (SSL)

The Secure Sockets Layer (SSL) protocol, which was originally developed by Netscape [15], is a set of rules governing server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information such as credit card numbers.

SSL requires a server SSL certificate, at a minimum. As part of the initial "handshake" process, the server presents its certificate to the client to authenticate the server's identity. The authentication process uses Public-Key Encryption and Digital Signatures to confirm that the server is in fact the server it claims to be.

Once the server has been authenticated, the client and server use techniques of Symmetric-Key Encryption, which is very fast, to encrypt all the information they exchange for the remainder of the session and to detect any tampering that may have occurred.

If a secure SSL connection is established between the web browser and the web server, the “http” in the web address will normally change to “https”, for example: “http://www.abcde.com” becomes “https://www.abcde.com”.

Servers may optionally be configured to require client authentication as well as server authentication. In this case, after server authentication is successfully completed, the client must also present its certificate to the server to authenticate the client's identity before the encrypted SSL session can be established [17].

Although SSL is the scheme proposed by Netscape Communications Corporation, SSL also gained the support of Microsoft and other Internet client/server developers as well and became the de facto standard until evolving into Transport Layer Security.

SSL has recently been succeeded by Transport Layer Security (TSL), which is based on SSL. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (http) and Transport Control Protocol (TCP) layers. TLS and SSL are an integral part of most Web browsers (clients) and Web servers. If a Web site is on a server that supports SSL, SSL can be enabled and specific Web pages can be identified as requiring SSL access.

Transport Layer Security (TSL)

Transport Layer Security (TLS) is a protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third party may eavesdrop or tamper with any message. TLS is the successor to the Secure Sockets Layer.

TLS is composed of two layers: the TLS Record Protocol and the TLS Handshake Protocol. The TLS Record Protocol provides connection security with some encryption method such as the Data Encryption Standard (DES). The TLS Record Protocol can also be used without encryption. The TLS Handshake Protocol allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before data is exchanged.

The TLS protocol is based on Netscape's SSL 3.0 protocol; however, TLS and SSL are not interoperable. The TLS protocol does contain a mechanism that allows TLS implementation to back down to SSL 3.0. The most recent browser versions support TLS. The TLS Working Group, established in 1996, continues to work on the TLS protocol and related applications. [16]

Secure Hypertext Transfer Protocol (SHTTP)

SHTTP is the scheme proposed by CommerceNet, a coalition of businesses interested in developing the Internet for commercial uses. It is a higher-level protocol that only works with the HTTP protocol, but is potentially more extensible than SSL. Currently, SHTTP is implemented for the Open Marketplace Server marketed by Open Market, Inc on the server side, and Secure HTTP Mosaic by Enterprise Integration Technologies on the client side.

Secure Hypertext Transfer Protocol defines a new URL protocol designator, 'shttp'. Use of this designator as part of an anchor URL implies that the target server is SHTTP capable, and that a dereference of this URL should undergo SHTTP processing.

A 14.2.2.7 IPSec

IPsec (Internet Protocol Security) is a developing standard for security at the network or packet processing layer of network communication. Earlier security approaches have inserted security at the application layer of the communications model. IPsec will be especially useful for implementing virtual private networks and for remote user access through dial-up connection to private networks. A big advantage of IPsec is that security arrangements can be handled without requiring changes to individual user computers. Cisco has been a leader in proposing IPsec as a standard (or combination of standards and technologies) and has included support for it in its network routes.

IPsec provides two choices of security service: Authentication Header (AH), which essentially allows authentication of the sender of data, and Encapsulating Security Payload (ESP), which supports both authentication of the sender and encryption of data as well. The specific information associated with each of these services is inserted into the packet in a header that follows the IP packet header. Separate key protocols can be selected, such as the ISAKMP/Oakley protocol.

Officially spelled IPsec by the IETF, the term often appears as IPSec and IPSEC.

Internet Key Exchange (IKE)

The IKE protocol is a key management protocol standard which is used in conjunction with the IPSec standard. IPSec is an IP security feature that provides robust authentication and encryption of IP packets.

IPSec can be configured without IKE, but IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard.

IKE is a hybrid protocol which implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

IKE automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration [18].

Specifically, IKE provides these benefits:

- Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPSec security association.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits Certification Authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

A 14.3 Assessment

The findings about privacy reviewed here can contribute to the ALFANET project in a number of ways:

- The tradeoffs on interface design will be noted, and implementations of 'unobtrusive' interfaces will be accessed and absorbed, where appropriate, into the prototype.
- User modelling privacy in ALFANET will benefit from compliance with P3P, and the design team may also wish to develop a more detailed specification along the lines described within Cobricks section.

The technical requirements of ALFANET system security are anticipated high, since trust is a central aspect between users as well as in the system. The state of the art in security technologies is sufficient for the implementation of a security scheme for the ALFANET.

As technologies applicable to ALFANET, the following are considered:

- Data Encryption (SSL – Secure Socket Layer, XML ciphered).
- User Authentication (Single sign-on for ALFANET users across all interfaces during runtime; Username/Password protection for content at URL level)
- Authorisation of exchanges of information (ACL – Access Control List).

In an initial approach, these technologies are used to form different levels of security on profiles or parts thereof for defined users or groups are envisioned. This means that special customisable rules could be applicable for each profile to ensure privacy.

The consortium should take into account security and privacy aspects for achieving an open design that will assure the future exploitation of ALFANET toolkit.

A 14.4 References

[1] Ackerman, M. Durrell, T. and Weitzner, D.J. (2001) Privacy in context. Human-Computer Interaction, 16, 167– 176.

[2] Ackerman, M. Cranor, L. and Reagle, J. (1999) Privacy in e-commerce: examining user scenarios and privacy preferences, Proceedings of the ACM Conference in Electronic Commerce. New York: ACM.

[3] Fink, J., Kobsa, A., and Schreck, J. (1997). Personalized Hypermedia Information Provision through Adaptive and Adaptable System Features: User Modeling, Privacy and Security Issues. Proceedings of the Fourth International Conference on Intelligence in Services and Networks, Como, Italy, Springer

[4] Hudson, S.E. and Smith, J. (1996) Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems, Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 96) New York: ACM.

[5] Ackerman, M.S. (2000) the intellectual challenge of CSCW: the gap between social requirements and technical feasibility. Human-Computer Interaction, 15, 179-203.

[6] Viega, J., Kohno, Tadayoshi and Potter, B. Trust (and mistrust) in secure applications: exploring and considering trust assumptions during every stage of software development. Communications of the ACM, 44 (2), 31 – 36.

[7] Martin, D., Smith, R., Brittain, M., Fetch, I. and Wu, H. The privacy practices of web browser extensions. Communications of the ACM, 44 (2), 45 – 50.
(available at http://www.privacyfoundation.org/advisories/advbrowserext.htm)

[8] Lau, T., Etzioni, O. and Weld, D. (1999) Privacy interfaces for information management. Communications of the ACM, 42 (10), 89-94.

[9] Koch and Worndl (2001) Community Support and Identity Management Proc. Europ. Conference on Computer-Supported Cooperative Work (ECSCW2001), Bonn, Germany, Sep. 2001

[10] Donath, J.S. (1998) Identity and deception in the virtual community, In P. Kollock and M. Smith (eds.). Communities in Cyberspace. London: Routledge.

[11] Platform for Privacy Preferences (P3P) Available at http://www.w3c.org/P3P

[12] Grimm and Rossnagel (2000) P3P and the privacy legislation in Germany: can P3P help to protect privacy worldwide?

[13] APPEL. (2000) A P3P Preference Exchange Language. W3C working draft, Feb. 26, 2001.

[14] Kobsa, A. (2001). Tailoring Privacy to Users Needs. Lecture Notes in Computer Science

[15] http://home.netscape.com/products/security/ssl/index.html

[16] http://searchsecurity.techtarget.com/sDefinition

[17] http://developer.netscape.com/docs/manuals/security/pkin/contents.htm

[18] http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113t/113t_3/isakmp.htm