JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche Fakultät

Activity Sequence Modeling and Multi-Targeted Clustering for Personalization in E-Learning

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:
Mirjam Augstein

Angefertigt am:
Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM)

Beurteilung:
Univ. Prof. Dr. Jörg R. Mühlbacher (Betreuung)
Univ. Prof. DI Dr. Gerhard Widmer

Mitwirkung:
Dr. Alexandros Paramythis, MSc

Linz, November 2011

Abstract

The impact of e-learning on the global learning process has been steadily growing over the past decades. Not only has e-learning, in different manifestations, made its way into the majority of educational settings, it has in many aspects caused a revolution of learning environments and a paradigm shift regarding the general understanding of learning itself.

While the traditional focus of e-learning lay on the individual learner, the theories and practices of learning collaboration of the physical world were progressively transferred into electronic counterparts. At the same time, the role of the learner changed from the one of a passive consumer of content to the one of an active participant, and learning in general shifted from a teacher-centered process to a learner-centered one.

In parallel to the change of the focus, also the technology used to support learning evolved. The general area of information systems underwent a process of steady growth in several directions, resulting in the information pool becoming insurmountably large for any single person to perceive, analyze and comprehend. This development gave rise to the need for information filtering and selection facilities separating relevant from irrelevant pieces of information.

As learners are different in various ways, including their interests, background, knowledge, long-term and short-term goals, their motives for interacting with the system, etc., there is no way of universally assessing the relevance of information, leading to the need for personalization.

In the area of e-learning, adaptive software systems autonomously tailor the appearance and amount of learning content, learning paths, and learning support to the individual learner's or groups of learners' requirements and characteristics.

This thesis focuses on the analysis and interpretation of learner activity data in order to provide a basis for reliable conclusions regarding learners' individual needs and requirements. It proposes a general approach to the utilization of activity data in the learner modeling process, emphasizing the fact that learner activities cannot necessarily be treated as independent from each other, but might be interrelated.

In more detail, an activity sequence modeling method is presented and discussed in combination with an unsupervised learning approach applicable at different levels in order to: detect predefined, well-established problem-solving styles in students' problem-solving sequences; discover new problem-solving styles along predefined learning dimensions; and discover potentially interesting learning dimensions and associated problem-solving styles.

Deliberations on how the gained pieces of information about learners' problem-solving behaviour can be fed back into the process by offering individual adaptations based on them, complete the cycle of adaptation. Furthermore, adaptivity in the area of e-learning in general, and in relation to the proposed approach, is discussed from the perspective of security and privacy, issues that are of high relevance in systems that rely on the collection and interpretation of users' (personal) data.

Kurzfassung

Über die letzten Jahrzehnte ist der Einfluss von E-Learning auf den globalen Lernprozess stetig gestiegen. E-Learning hat in verschiedensten Formen und Ausprägungen Einzug in einen Großteil der Bildungsszenarien gehalten, ein Umstand der nicht nur eine Revolution der Lernumgebungen angestoßen, sondern auch einen Paradigmenwechsel im allgemeinen Verständnis des Lernvorgangs verursacht hat.

Während im E-Learning-Bereich, anders als in der nicht-digitalen Welt, traditionell der individuelle Lerner im Vordergrund stand, rückte in den vergangenen Jahren kollaboratives Lernen immer mehr in das Zentrum der Aufmerksamkeit. Gleichzeitig änderte sich die Rolle des Lernenden von der eines passiven Informationskonsumenten zu der des aktiv den Lernprozess mitgestaltenden Teilnehmers. Ein ehemals Lehrer-zentrierter Prozess wandelte sich zu einem Lerner-zentrierten.

Parallel zu dieser Fokusverschiebung entwickelten sich auch die den Lernprozess unterstützenden Technologien rasant weiter. Nicht nur im Bereich des Lernens erfuhr die Gesellschaft einen Wandel zur Informationsgesellschaft, die stetig wachsende, immer unübersichtlicher werdende Mengen an Informationen zu verarbeiten hatte. Methoden zur Informationsfilterung und -selektion wurden unabkömmlich um die, für das Individuum relevanten Stecknadeln im Heuhaufen zu identifizieren; ein Prozess, der aufgrund der Vielfalt und Unterschiedlichkeit einzelner Personen in verschiedensten Aspekten keinem universellen Schema folgen konnte.

Die neuen Anforderungen ließen den Wunsch nach Personalisierung im Informationschaos erwachen – die Idee adaptiver Systeme war geboren. Adaptive Systeme passen sich automatisch an die individuellen Bedürfnisse, Anforderungen und Charakteristika des einzelnen Benutzers an, um diesen im Arbeitsprozess bestmöglich zu unterstützen, ein Konzept das auch im Bereich des E-Learning mittlerweile zu großer Popularität gefunden hat.

Diese Arbeit befasst sich mit der Analyse und Interpretation von Lerneraktivitäten als Basis für zuverlässige Schlussfolgerungen über die individuellen Bedürfnisse der einzelnen Lerner. Ein in sich abgeschlossenes, allgemeines Verfahren zur Verwendung von Lernerdaten in der Benutzermodellierung, wird beschrieben, das besonders die Bedeutung von Relationen zwischen Aktivitäten hervorhebt. Lerneraktivitäten werden nicht weiter als in sich abgeschlossene Interaktionen mit dem System interpretiert sondern vielmehr als Bestandteile von Aktivitätssequenzen, die sich mehr oder weniger stark in Zuständen gegenseitiger Beeinflussung oder Abhängigkeit befinden können.

Diese Arbeit stellt einen Ansatz zur Modellierung von Aktivitätssequenzen vor, der, in Verbindung mit einem unüberwachten maschinellen Lernprozess zur Mustererkennung, auf mehreren Ebenen angewandt werden kann: um vordefinierte, konkrete Problemlösestile in Aktivitätsdaten zu identifizieren, um neue, bislang undefinierte Problemlösestile innerhalb vordefinierter, so genannter "Dimensionen" zu erkennen, und um automatisiert potenziell bedeutsame Dimensionen und sich darin manifestierende konkrete Problemlösestile ausfindig zu machen.

Eine anschließende Diskussion behandelt sowohl die Rückführung der, durch den beschriebenen Prozess gewonnenen Informationen über individuelle Problemlösestile, in den Adaptionskreislauf, als auch die durch die Gewinnung, Verarbeitung, Interpretation und eventuelle Weitergabe personenbezogener Interaktionsdaten auftretenden Fragestellungen zu Sicherheit und Privatsphäre der Benutzer.

Acknowledgements

The process of writing this thesis was an interesting, educating, exciting, multicoloured and encouraging one, not least because of the support of many people who accompanied me through the ups and downs I faced during that time. I want to take this opportunity to thank these people who that have encouraged and motivated me, who have patiently beard my lack of time, who have encouraged me to think in different directions, and who have generally helped me to improve this thesis and the research efforts behind it.

First of all I want to thank my supervisor Univ. Prof. Dr. Jörg R. Mühlbacher who not only gave me the opportunity to participate in two outstandingly interesting research projects that have inspired me and essentially shaped my further journey through the scientific world, but also patiently provided a lot of valuable input that helped me to significantly improve this thesis and preceding work.

I also want to thank Univ. Prof. Dr. Gerhard Widmer for repeatedly taking the time to discuss my ideas and pointing out different perspectives on them to me which especially encouraged creative and critical thinking, and for his helpful reviews on previous versions of this thesis.

I want to thank Dr. Alexandros Paramythis who has guided my work for years. You did not only awaken my huge interest in scientific work in general, and in adaptivity and related fields in specific, but you were and still are a perfect example of a researcher to me because you continuously manage to pass on your fascination and excitement for new ideas or concepts to others. Thank you for your never-ending patience and trust and the numerous discussions of my ideas – I think I would not be who I am and where I am now without you.

I want to thank the rest of the FIM[1] team for four years of good cooperation and support, especially the adaptive wing including the “Skiing Dutchman” that has enriched these years with adaptive BBQs, adaptive (and awful, regarding our performance) tennis matches, adaptive wine tasting and much more.

I want to thank my mother, Hanna Köck for always being there for me whenever I needed her. Thank you for supporting me in everything that was important to me, and for giving me the opportunity to take up studies at a university in the first place.

I want to thank David Schwingenschlögl for again pointing out a lot of little, mean spelling mistakes, after he had already done a perfect job as a proof-reader on my diploma thesis some years ago.

[1] Institut für Informationsverarbeitung und Mikroprozessortechnik, Johannes Kepler University, Linz

Last but not least I want to thank my husband, Mag. iur. Jürgen Augstein for patiently supporting me throughout the writing process. Thank you for never complaining because of my lack of time, for enduring the full spectrum of my moods, for helping me wherever you could to keep additional work away from me, and for the thousands of other little and not so little things I do not have enough space here to mention.

Contents

List of Figures

List of Tables

Acronyms

AAHS – Asynchronous Adaptive Hypermedia Systems

ADL – Advanced Distributed Learning

AH – Adaptive Hypermedia

AHS – Adaptive Hypermedia Systems

AI – Artificial Intelligence

AJAX – Asynchronous JavaScript and XML

ANN – Artificial Neural Network

ARFF – Attribute-Relation File Format

BN – Bayesian Network

CBT – Computer-Based Training

CF – Collaborative Filtering

CI – Computational Intelligence

CMS – Course Management System

CRF – Conditional Random Field

DMM – Discrete Markov Model

EDM – Educational Data Mining

FCM – Fuzzy cMeans

(G)MM – (Gaussian) Mixture Model

GTN – Graph Transformer Network

HMM – Hidden Markov Model

IMS CP – IMS Content Packaging

IMS LD – IMS Learning Design

IMS LIP – IMS Learner Information Package

IMS QTI – IMS Question & Test Interoperability

IMS SS – IMS Simple Sequencing

IOHMM – Input-Output Hidden Markov Model

ITS – Intelligent Tutoring System

KC – Knowledge Component

LDA – Linear Discriminant Analysis

LMS – Learning Management System

LOM – Learning Object Metadata

LTSC – Learning Technology Standardization Committee

MEMM – Maximum Entropy Markov Model

NB – Naïve Bayes

NN – Nearest Neighbour

P3P – Platform for Privacy Preferences

PAPI – Public and Private Information

PCA – Principal Component Analysis

RCC – Relational Collaboration Coefficient

RSS – Really Simple Syndication

SCORM – Sharable Content Object Reference Model

SMO – Sequential Minimal Optimization

SOM – Self-Organizing Map

TA – Teachable Agent

TRFS – Top Ranked Feature Set(s)

ULSM – Unified Learning Style Model

UML – Unified Modeling Language

Chapter 1

Introduction

This thesis discusses, in general terms, the application of machine learning techniques in adaptive e-learning systems, and, more specifically, the modeling of sequential usage data as a basis for clustering in order to identify different kinds of learner behaviour.

The analysis and interpretation of learner behaviour is of crucial importance in the scope of personalization in e-learning, as the information gained there founds the basis for individual and group learner models. These models again facilitate the application of different kinds of adaptive support, based on different pedagogical and didactic approaches within the learning process. Especially when considering collaboration in an adaptive e-learning setting, several challenges are faced.

The work reported in this thesis was carried out within the scope of the ASCOLLA[1] project aimed at facing these challenges. The project's objectives can be summarized as follows [Paramythis and Mühlbacher, 2008]:

- individual and group learner modeling,
- modeling and employment of didactic approaches,
- adaptive awareness,
- adaptive support for collaboration establishment,
- adaptive support for creating personal learning histories,
- adaptive support for ongoing collaboration / cooperation

The work described herein concentrates on the learner modeling part but also contributes to adaptively supporting collaboration and collaboration establishment by the information it compiles about learners' different approaches to problem-solving.

As already shortly introduced, the ASCOLLA project comprised several fields of research and therefore also scientifically contributed to several areas as listed in the following section, emphasizing e-learning and adaptivity.

[1] ASCOLLA – Adaptive Support for Collaborative E-Learning, supported by the Austrian Science Fund (FWF; project number P20260-N15), carried out between 2008 and 2010.

1.1 Fundamentals Behind Personalized Learning

This section provides a brief overview of the basics of three different research areas and their respective historical outlines that are connected within the scope of personalized learning: e-learning in general, adaptivity, and machine learning. The first provides a highly interesting application environment for the concepts of adaptivity, while the latter equips the interdisciplinary setting with analysis methods and techniques.

1.1.1 E-Learning

During the past decades, e-learning has become a more and more important part of the global learning process and step by step revolutionized educational environments. There are many synonyms for the term "e-learning" that are used in literature but there does not seem to be consensus on their respective definitions and whether they are to be used as synonyms or if they all describe similar yet different concepts. For example, we can find the terms "computer-based learning" [Association of American Colleges, 1981], "technology-enhanced learning" [Heeter, 1999], or "networked learning" [Steeples and Jones, 2002] used in relevant literature. In the context of this thesis, the term "e-learning" will be used and can be understood as combining the other ones mentioned.

Historical Outline

Early roots of e-learning can be found almost thirty years ago in the early 1980s when researchers identified the value of decentralization and temporal flexibility for the learning process utilizing the computer as a medium to present text-based learning material [Lackinger and Mühlbacher, 1984]. The new kind of learning material was seen as an extension or complement to the traditional script and could be accessed in educational institutions' specially equipped computer labs. The approach did not aim at replacing either the paper-based script or the educational setting in general but built upon the advantages involved with asynchronous communication and the discontinued necessity for students and lecturers to be at the same place at the same time.

About ten years later, in the 1990s, when e-learning had further developed to Computer-Based Training (CBT), presented, for instance, as learning packages on CD-ROM or made available as web-based seminars [Aiken et al., 1998], fortified doubts arose among the involved parties (e.g., university professors, teachers and their respective institutions) [Mühlbacher, 1998]: Would the new e-learning concepts be able to replace traditional classroom settings, and thus also the role of the trainer? Would the differences between the "brick and mortar" university and the distance university be suspended? How could the quality of the learning material be evaluated and assured?

Despite the dissents and doubts, e-learning became more and more important in a variety of educational settings during the following years. End of the 1990ies, the aspect of collaboration was integrated in the idea of e-learning which originally had focused on the individual learner. For instance, [Aiken et al., 1998] describe a scenario where a seminar on interactive learning held in Zurich (Switzerland) was attended by some participants from Linz (Austria). Active group discussion as well as communication with experts via the Internet was explicitly encouraged. In addition, students had to collect material and present results on web pages. Traditional e-mail, mailing lists, and a forum were used as communication channels. One of the main challenges involved in the experiment was to encourage students to actively participate.

In general, the endeavour was successful as the involved universities could identify several benefits for different parties. First, students could practice using the Internet for purposes of information retrieval and communication, they could experience inter-cultural cooperation and were given the opportunity to participate in team organization and division of responsibilities.

Second, the universities could divide the workload among several teachers, the teachers could cooperate in team teaching and reuse their material in the following years. Furthermore, as opposed to traditional classroom settings, the number of participants was not limited by the number of available seats and students could organize their work being independent from teaching hours.

However, also problems and potential disadvantages were reported: some of the students, enthusiastic in the beginning, lost motivation during the duration of the course, some participants even dropped out. In addition, the different semester start and end dates of the different universities complicated the process, as participants were at different levels of knowledge during the course. In general, group work and group-based assessment are often not entirely transparent to the teachers, i.e., it can be hard to assess the individual group members' contributions.

In the beginning of the 2000s, the trend towards collaborative work became more and more popular in e-learning. In parallel, a general paradigm shift and transformation of roles could be observed – instructors became coaches and learners became more and more active parts of the process instead of just receiving and consuming content [Mühlbacher et al., 2002].

While traditionally the focus lay on the teacher ("teacher-centered teaching", see, for instance, [Paris and Gespass, 2001]), it has been moving more and more towards the learner ("learner-centered teaching", see, for example, [Weimer, 2002] or [Weld, 2002]).

Learner-centering, according to [Anderson, Terry, 2004], means that the "whims and peculiarities of each individual learner are uniquely catered to". Anderson further argues that it is thus necessary to ensure that also the needs of teachers, institutions, and of the society supporting the student and the institution, are met.

Over the decades, the technology used to support e-learning, evolved. The simple "making-available" of learning content via so-called learning environments was not sufficient for very long. Aspects like reusability of content and greater coverage of didactical models and pedagogical strategies became more and more important. Learning platforms should better support and motivate the learner, the learner should furthermore gain a better understanding of specific topics, achieve better results and also acquire profound social skills.

During the past years, as smartphones and tablets flooded the market for mobile devices, another form of learning, so-called "mobile learning" or "m-learning" attained more and more importance in supporting and complementing e-learning processes. M-learning does not aim at replacing classical e-learning but at enhancing the process. M-learning is characterized by extraordinary flexibility regarding time and location, as the learner can basically learn anywhere at any time without the necessity of sitting in front of a desktop machine.

[Loidl-Reisinger, 2006] describes the m-learning setting as "anywhere, anytime, any data and any device", i.e., learning content can be retrieved, viewed or repeated via an arbitrary device from an arbitrary place which is now also supported by the numerous free wireless access points providing Internet access in public places.

Thus, m-learning is not necessarily a different kind of learning, but rather characterized by different learning conditions and environments, as stated in [Seipold and Pachler, 2011], and users are encouraged to get to know their everyday life-worlds as learning spaces.

These developments continued revolutionizing the concepts of learning and teaching during the following decade and brought along different developments in the conception and implementation of learning tools and platforms as further described in the following section.

Especially during the past few years, the importance of e-learning and therefore also the need for higher quality of learning content and support has been steadily growing. Not only have schools, universities, and other educational institutions realized and exploited the benefit of e-learning as a supplement to traditional learning scenarios in "blended learning" settings, but a new branch of education, based on pure e-learning in the context of online courses, trainings and whole courses of studies, has been risen.

Standardization

Going hand in hand with learning environments becoming more complex and more frequently used in different contexts, more simple transferability between the platforms was demanded. Learning content should become reusable, interoperable, more easily accessible and more durable [Svensson, 2001], [Loidl-Reisinger and Paramythis, 2003] – the need for standards had arisen. [Varlamis and Apostolakis, 2006] summarize the motivations and aims behind standardization in e-learning as follows:

- Standards would enable users to switch between e-learning platforms and programs after having become familiar with standardized e-learning technology.

- Producers of e-learning content could focus on the development of content in a standardized format which would also prevent them from having to put effort into organizing the same content in different ways to make it suitable for different platforms.
- E-learning facility vendors could lower their development costs which would in turn also make the respective tools cheaper.
- A large selection of reusable e-learning content is available to application and platform designers which enables them to assemble the content and tools they consider most efficient and suitable for the specific context.

E-learning standards do not only consider the content itself but the whole e-learning process including initial learning design as well as production, deployment and assessment. Different standards take effect on and provide settings and structures for different phases and elements of this process. [Varlamis and Apostolakis, 2006] group interoperability standards into the following categories: content description (metadata), content packaging, learner management and communication of results.

Content description (metadata) requires learning components, in order to be well categorizable and searchable, to be supplemented with consistent meta information, which is, for instance, defined in the IEEE Learning Object Metadata (LOM) standard [IEEE 1484.12.1-2002, 2002]. Different formats of learning objects and difficulties in integration are, according to [Sonntag, 2006], the main reasons for the need for the integration of metadata in e-learning. With "different formats", Sonntag refers to the formats of bundles, i.e., learning packages, rather than to the formats of the actual learning content.

Content packaging defines how learning content can be bundled in a standardized way so that it can be easily exported from and imported to e-learning environments. The resulting packages can include not only the content itself but also information on assembly, delivery and presentation. Content packaging is considered by different standards or specifications, for example in IMS Content Packaging (IMS CP) [IMS Global Learning Consortium, 2003a], IMS Simple Sequencing (IMS SS) [IMS Global Learning Consortium, 2003c], Advanced Distributed Learning (ADL) initiative Sharable Content Object Reference Model (SCORM) [Advanced Distributed Learning Initiative, 2002] or IMS Question and Test Interoperability (IMS QTI) [IMS Global Learning Consortium, 2003b]. The latter refers to assessments as a special kind of learning content. Questions and tests are part of most e-learning courses and should thus be particularly well transferable, regarding not only the assessment content itself but even more the structure and concepts behind the test (for example, single or multiple choice tests, open text questions, etc.).

Learner management defines a format for describing learner profiles. A learner profile contains data about a learner's registration information and privileges but can also store additional information about a learner's learning characteristics, knowledge and aims, which is essential within adaptive learning systems (see Section 1.1.2). In order to enable sharing of learner profiles between different platforms, standardization is relevant also in this area. Learner information is modeled, for instance, by IMS Learner Information Package (IMS LIP) [IMS Global Learning Consortium, 2001] or IEEE Learning Technology Standardization Committee (LTSC) Public and Private Information (PAPI) [IEEE Learning Technology Standardization Committee, 2001].

Communication of results defines a format for so called “performance reports” [Varlamis et al., 2005]. A performance report contains any kind of feedback to a learner’s interaction with the system, especially regarding assessments. Standardized communication protocols and data models should enable communication-flow between the system and the learning components, and are integrated, for instance, into the initiative of ADL SCORM.

Concepts and Technologies

As can be read from the previous section, not only the technologies used to facilitate or support teaching changed but also pedagogical/didactical strategies and teaching approaches underwent a drastic change. However, as pointed out by [Ally, 2004] there is an ongoing discussion about whether successful (online) teaching relies rather on the technologies used or on specific instructional design [Clark, 2001], [Kozma, 2001]. Ally claims that special delivery technologies can provide efficient access to learning material. However, it is suggested by [Schramm, 1977] that it is the content and instructional strategy that influence learning more than the delivery technology, which is also confirmed by [Bonk and Reynolds, 1997]. Summing up, we can state that what leads to success regarding efficiency of learning usually is a combination of both technology and strategical content design.

Various technologies for online learning along with their use in education are described by [McGreal and Michael, 2004], who list the following technologies that can be utilized in e-learning:

- *streaming audio* as a supplement to classroom-based course delivery, for example, in the form of pre-recorded lectures,
- *streaming video*, delivering, for example, a prepared lecture,
- *push technologies and data channels* delivering, for example, news or information from relevant course sites,
- *audio chat and voice-over-IP* as a supplement to the traditional text chat that can be utilized for synchronous teacher-to-students or student-to-students communication enhancement,
- *web whiteboarding*, often combined with voice-over-IP, and allowing for the emulation of classroom lessons because students can actively participate,
- *instant messaging*, used for direct communication between the participants of a learning scenario, however, not found useful as a means of content delivery yet,

- *hand-held and wireless technologies* that already started to replace, for example, paper-based teaching and learning, and become more and more important as affordable access to high bandwidth increases and the cost of wireless devices decreases,

- *peer-to-peer file sharing*, used for offering research and other materials to other participants, and

- *learning objects*, i.e., reusable learning material consisting of discrete lessons, learning units or courses that can be incorporated into various learning scenarios.

[Fahy, 2004] analyzes media characteristics in the context of learning technology and lists the following media because in his opinion, these constitute the most popular tools in online learning: print and text, still graphics and illustrations, sound and music, video and moving graphics, and multimedia. Fahy's analysis is based on a six-element topology of teaching tasks and objectives [Fleming, 1987] that includes attention, perception and recall, organization and sequencing, instruction and feedback, learner participation, and higher-order thinking and concept formation.

According to Fahy, *attention* is a fundamental aspect in order to achieve learning success and thus training must be able to attract and to hold the learner's attention, which usually tends to be individual, selective, fluid, and especially attracted to novelty.

Perception requires the learner to selectively focus on and make sense of stimulation in the environment, and *recall* includes a student's ability to memorize and reuse relevant learning.

Organization and sequencing is considered to be strongly related to the detection of diversity in learners' needs because this can lead to the need to reorganize material and activities. In general, organization and sequencing is regarded a highly important task in instructional planning [Fleming, 1987]. In specific, media design must apply the following principles: (a) the first and the last items in a sequence are of particular importance, (b) modeling and demonstrations may result in learning, and (c) learning can be increased by repetition and review [Fleming, 1987].

Feedback, according to Fahy, is, in addition to skillful *instruction*, needed to allow learners monitor their progress, but also to enable them to recognize potential for improvement. However, it has to be considered that not every kind of feedback is helpful for every type of learner and should therefore be selected carefully (for instance, a general principle suggests a more mature learner to require more informative feedback (see also [Fleming, 1987]).

Learner participation is essential because "learning requires engagement with the subject matter, and engagement often implies some kind of performance".

Further, *concept formation* is relevant because learning concepts are said to be part of a process leading to engagement with related concepts. For example, a complex problem-solving task (as will become relevant in the context of this thesis), requires the learner to be able to recall previously learned concepts but also to combine them autonomously.

Finally, *higher-order thinking* skills are considered challenging in the e-learning context because the respective systems must "move beyond the mere identification and use of facts, to creative and synergistic linking of concepts".

For [Fahy, 2004] and [Fleming, 1987], the facts described above lead to the following implications for the design of technology in teaching. *Attention* must be attracted and held by, for example, change and variety but, in parallel, also by similarity, predictability and routine in the teaching/learning process.

In *organization and sequencing* individual learner differences should be considered and repetitions should contain variety (e.g., para- or rephrasing).

Feedback should also be tailored to the individual student. For instance, for mature learners a system's "incorrect" responses should be accompanied by additional explanatory feedback.

According to [Cannell, 1999], the degree of *learner participation* depends on creativity of the participants, the resources and technologies available. Learners can be integrated by, for example, questions, seminars, learning teams, peer groups or presentations of written reports.

Higher-order thinking and *concept formation* can be encouraged by activities like analysis, synthesis and evaluation.

According to [Ally, 2004], it is not only challenging for institutions delivering online courses to appropriately prepare their learning material and use proper technologies, but also to create a perception among the students that online learning provides benefits for them.

Ally mentions the following advantages distance students gain through e-learning:

- time zones, location and distance are not an issue any more,
- materials can be accessed at any time,
- real-time interaction between students and instructors is possible,
- the Internet can be used in parallel to communicate with experts in the field and to access supplementary learning material,
- learners can complete online courses while working in a job, and
- learning can be contextualized.

A similar list of advantages is provided from the instructor's viewpoint [Ally, 2004]:

- tutoring can be done any time and from anywhere,
- learning material can be updated in a way that enables learners to view the changes at once,
- it is easier to direct learners to specific information based on their needs, and

- online learning systems can help to determine learners' individual needs and requirements.

The last point ultimately leads to the need for personalized learning systems.

Compared to a traditional educational setting, planning an online course involves several additional considerations, as pointed out by [Harmon and Jones, 1999] who introduce several different "levels of web use" in education, based on how the web could be efficiently used for online education.

Their levels "represent a continuum from basic occasional use to advanced continual use". Harmon and Jones state that not for all settings the web must be necessarily useful and present factors that could be helpful to determine what level is appropriate for the respective scenario. The following list introduces the six levels of web use:

- *Level 0 – no web use*
- *Level 1 – information web use:* this level means that only stable information is provided to the students. Mostly this kind of information is of organizational nature and may not even contain course content.
- *Level 2 – supplemental web use:* this level suggests the web as an additional medium for information delivery, for example, to offer presentations or documents for download. It is, however, not used for provision of the core course content which still takes place in the traditional classroom setting.
- *Level 3 – essential web use:* this level suggests regular web access to the student, in order to be a productive class member. Classes however are still expected to meet face-to-face.
- *Level 4 – communal web use:* at this level of web use, classes meet both face-to-face and online. Also, course content may be provided in a traditional paper-based way or online. At this level, pure hypertext-based material is not sufficient any more, it requires use of additional online tools and media like forums or videoconferencing. Thus it is a prerequisite for both instructors and learners to possess fairly good knowledge about web technologies.
- *Level 5 – immersive web use:* this level substitutes the traditional face-to-face classroom setting with online meetings, i.e., instructors and learners do not meet offline any more at all. Thus, both must have a high level of technical knowledge.

In addition to the levels just explained, Harmon and Jones discuss various factors they consider useful for the decision for a specific level: distance, stability of material, need for multimedia, need for student tracking, number of students, amount of interaction, social pressure to use the web, need for offline reference, infrastructure, comfort levels and access.

Distance, for example, is of relevance because if instructors and/or students are spread over greater geographical distances, it is difficult or even impossible to organize a face-to-face meeting. Thus, the web is of higher value in such a scenario than it is in a setting where instructors and students could theoretically meet face-to-face at any time. To give another example, access is concerned with the question whether course material is accessible and how. If a course participant or instructor travels a lot, material should be accessible from everywhere, i.e., via the web. However, access does not solely mean Internet access, but also access to equipment.

As already shortly introduced, e-learning does not have to indicate that everything (for instance, course and course material) has to move to the web, it could also be applied in integrated settings where both traditional face-to-face meetings and online ones are combined. This kind of setting is usually labeled *blended learning*.

According to [Graham, 2005], “blended learning” has become a buzzword in educational and corporate settings that still lacks a uniform definition. Therefore, Graham lists the definitions he considers the most common ones.

Blended learning can be described as “combining instructional modalities” (see, for example, [Singh and Reed, 2001]), as “combining instructional methods” (see, for example, [Rossett, 2002]), and as “combining online and face-to-face instruction” (see, for example, [Reay, 2001] or [Rooney, 2003]).

The last definition is maybe the most popular one at the moment and also better reflects the historical emergence of blended learning environments [Graham, 2005]. The first two definitions are thus maybe less relevant but interesting though, as they “reflect the debate on the influences of media versus method on learning” [Graham, 2005] (see also, for instance, [Clark, 1983], [Clark, 1994], or [Kozma, 1991]).

In general, multiple reasons suggesting the introduction of a blended learning setting can be identified. For example [Osguthorpe and Graham, 2003] list the following motivations:

- pedagogical richness,
- access to knowledge,
- social interaction,
- personal agency,
- cost effectiveness, and
- ease of revision.

While in traditional learning settings transmissive strategies are still more popular than interactive ones, in blended learning settings the number of active learning strategies becomes higher, for example, through the application of learner centered-, collaborative or peer-to-peer learning strategies, which leads to enhanced *pedagogical richness* which in this case goes hand in hand with the factor *social interaction*.

One might argue that social interaction tends to decrease in blended learning settings because the direct face-to-face contact between instructors and learners is lost, but on the other hand, blended learning opens additional channels of communication and cooperation and in parallel widens the range of opportunities to get in touch with other people that do not have to be in the same geographical location.

Furthermore, *access to learning and learning material* often becomes simpler and more flexible in blended learning settings as it is not necessary any more to meet at a specific, predefined location at a specific, predefined time. Also, blended learning if applied properly, can be very *cost efficient*, among other factors, because the contents are *reusable*.

[Graham, 2005] additionally discusses *levels of blending* and lists the *activity level*, where blending occurs when both face-to-face and virtual elements are included in a learning activity, the *course level*, where blending occurs when a course makes use of both face-to-face and online activities, the *program level*, where blending occurs when participants can choose a mix between face-to-face courses and online ones, and the *institutional level*, where blending occurs if institutions make organizational commitments to both face-to-face and virtual learning.

Blended learning does not only bring along advantages and potential, it also involves additional challenges. For example, the role of live interaction must be redefined, new models for support and training must be developed, a balance between innovation and production must be found and there must be strategies to deal with digital divide [Graham, 2005]. Blended learning can by now be rather considered a standard than a trend – it is likely that the question of *whether* to blend will vanish and be replaced by the question of *how* to blend.

Both pure online and the various blended learning settings have in common that they need to consider the fact that the learning process is different, compared to the traditional classroom setting, and that thus also learner support differs. Direct general and individual instructions as given in a classroom can in some cases not be given as easily in an online / blended learning context because first, the instructors receive less direct feedback from the learners, and second, the learning path through the material might differ for every student. Information and input that are relevant for some learners might thus be completely inappropriate for others.

Adaptivity as will be introduced in Section 1.1.2 can therefore enhance not only learning content by personalization, but also learner support. Learner support involves, for example, individual feedback, additional explanations where needed, or hints and assistance in problem-solving.

Another important (although less relevant for the purpose of the work presented in this thesis) aspect of learner support is introduced by [Hughes, 2004], who discusses non-academic support elements and list, for example, administrative and technological support or study skills assistance.

E-Learning Platforms

During the past years, a variety of online learning management environments have been made available. Thus it is practically impossible to provide a complete list, which is why some of the most frequently used ones were selected (see a more detailed discussion by [Hauger and Köck, 2007][2]) and will be shortly described in the following paragraphs.

Blackboard [Blackboard Inc., 2010] was founded in 1997 and provides course and content management features, collaboration tools and a number of other services combined in an "Academic Suite" and a "Business Suite". It is one of the most popular commercial e-learning systems and, for example, also offers a version for mobile learning.

OLAT [OLAT, 2011] is a free Learning Management System (LMS) developed since 1999 in Zurich. It also contains learning as well as communication and collaboration facilities and aims at offering didactical freedom so that learning content can be organized and presented in different ways. Like most LMS it was mainly developed to support learning and thus to be applied by educational institutions but also offers a special service for companies.

Moodle [Moodle, 2010] is a free learning environment that has its origins in 1999. The system's general design tries to consider pedagogical principles and learning theories. The lesson module of Moodle also provides different learning paths. As the user's possible answers on a question can be used as starting points for different learning paths, some kind of "very weak adaptivity" is supported.

WeLearn [Divotkey et al., 2002], [Mühlbacher and Putzinger, 2006], [Putzinger and Szedmina, 2006] is a free, web-based learning environment the development of which started in 2002 by the Institute of Information Processing and Microprocessor Technology (FIM), Johannes Kepler University in Linz. WeLearn is organized as a framework consisting of the platform itself, the settings and the course materials [Mühlbacher et al., 2002]. It does not only provide learning and organizational facilities but offers communication and interaction support in addition. These components can be arbitrarily arranged by the course administrator which makes possible the application of different teaching strategies. WeLearn aims at encouraging the learner to become an active part in the learning process, acting in a self-organized way and constructing problem-oriented knowledge. WeLearn is still used in a variety of different 2nd and 3rd level educational institutions in Austria and was one of the officially endorsed e-learning platforms of the Austrian Federal Ministry of Education, Science and Research.

ATutor [ATutor, 2010] is a free system that was designed to support learning and content management and to specifically consider accessibility and adaptability issues. It was first released in 2002 after two studies were conducted that evaluated the accessibility of learning platforms to people with disabilities. The system in its latest release became a collection of tools for creating online classrooms.

.LRN [.LRN, 2010] is a free e-learning and community building software originally developed at the Massachusetts Institute of Technology (MIT). Today it is supported by a worldwide consortium of educational institutions, non-profit organizations, some industry partners and

[2] Please note that Mirjam Köck, who is cited several times at different places, and Mirjam Augstein, author of this thesis, are the same person. The name change was caused by getting married during the process of completing this thesis.

open source developers. .LRN is built on the top of OpenACS (Open Architecture Community System) [OpenACS, 2010] which is a toolkit for developing scalable, community-oriented web applications.

Sakai [Sakai, 2010] is a service-oriented Java-based free LMS developed in 2004 by the universities of Michigan, Indiana, Stanford and the MIT, who contributed their existing LMSs to the new e-learning platform. Later other projects and partner institutions joined the Sakai community and developed Sakai tools based on their products. Today Sakai is developed by 116 cooperating organizations and funded via a partners program.

CLIX [imc Advanced Learning Solutions, 2010] is another commercial LMS developed by the imc (information multimedia communication) AG. It is available in different releases, each especially suitable for different application scenarios. Additionally, there are a couple of auxiliary features that can be added to the basic application in order to fit the individual needs of a scenario or project.

These and other LMSs made their way into a variety of different educational settings. However, only few made attempts to individually support their users or groups of users (for example, through the integration of an intelligent agent in the WeLearn system [Sonntag, 2003]), many of them do not or only weakly support personalization in learning processes, i.e., the learner, for example, receives learning content and navigation through this content that is tailored to individual needs, as introduced in the following section.

1.1.2 Adaptive Systems - One Size Does Not Fit All

During the past decades, the world of information systems underwent a process of steady growth in several different directions. Not only did computers make their way into almost every field of people's business and private lives and also constantly varied their appearances, but also information itself and our way to deal with it changed.

The information pool became larger and larger, one might even say inscrutably huge, which is accompanied by the need for new and better techniques to pick the relevant parts. It is not sufficient any more to receive and process information that is brought to us by various kinds of media.

Given the enormous amounts of data that is produced and broadcast, it is not possible any more to deal with all – we have to pre-select information which is potentially relevant and in a second phase filter it according to our personal interests and current requirements. In the modern information society, efficient information management strategies are an important foundation for success in education, business, research and development, etc.

Adaptive software systems approach the challenge of proper information provision from different perspectives. First, they take into account that all available information may be too much to be processed and understood by every user, and second, they correctly consider that users are different in various ways. Users may differ in their interests, backgrounds, knowledge, their long-term and short-term goals, their motives for interacting with the system, etc.

These differences reveal more and more clearly that it is unrealistic to find a "bespoke solution" that matches all different needs – *one size cannot fit all* in a contemporary information system.

Personalized systems autonomously adapt their appearance, behaviour and data representation to the individual user's requirements and characteristics. This concept is made use of in, for instance, e-learning systems which is the most relevant application area within the scope of this thesis and also within the ASCOLLA project.

Other popular contexts adaptivity is successfully implemented in, are *e-commerce* with Amazon [Amazon.com, 2010] being a widely known adaptive (recommender) system, *web browsing* (see, for example, the AVANTI web browser [Paramythis, 2009]), *(web) search*, where search results can be arranged according to a user's current requirements and interests (see, for example Prospector [Schwendtner et al., 2006], [Paramythis et al., 2008], [van Velsen et al., 2009], [Paramythis and van Velsen, 2009], [König et al., 2009]), or location-based guidance (see, for example, the PALIO tourist information services [Paramythis, 2009]).

Ideas, Concepts and Techniques

The basic concept of adaptivity thus overcomes two problems emerging through the flood of information as it can be found everywhere in the modern "information society". First, adaptive systems can in general *limit the amount of information*, based on different strategies that will be introduced and partly discussed in detail later in this thesis, and second, they can *tailor information* to a user's very personal requirements and interests. Therefore, the system evolves from a pure information provider into a more advanced assistant for finding the individual's personal needle in the haystack.

Adaptive systems need to know specific characteristics about their users in order to draw reliable conclusions they can base their decisions on. This information is usually stored in and retrieved from a *user model* representing the user's knowledge, interests, goals and other attributes (see, e.g., [Brusilovsky and Millán, 2007]). Data that goes into the user model is collected from various sources that do not only include explicit statements a user provides in order to help the system create a realistic model, but also data from implicit user interaction observation [Brusilovsky and Maybury, 2002].

[Kobsa and Schreck, 2003] divide information about a user relevant for personalized systems into *user data*, *usage data*, and *environment data*. *User data* contains, for example, demographic data, user knowledge, goals, or preferences, *usage data* comprises all kinds of observable system usage, and *environment data* includes, for instance, information about the hardware and software environment.

[Teltzrow and Kobsa, 2004] use a similar classification, splitting, however, *usage data* into *usage data* and *usage regularities*, where *usage data* comprises viewing behaviour, ratings or selective actions, and *usage regularities* include usage frequency, action sequences or situation-action correlations.

The data stored in a user model can differ and is dependent on the application area. While in an adaptive shopping portal the user's interests, profession or hobbies may play a very important role, in an e-learning system it is more interesting for the system how much the user already knows about specific subjects, what kind of learning behaviour is preferred, etc.

Adaptivity is most popular in the area of the web where the research field of *Adaptive Hypermedia (AH)* and *adaptive web-based systems* has been growing rapidly. The first "milestone" was laid with a publication in 1996 [Brusilovsky, 1996] where an initial overview of adaptation methods and techniques was given, followed by an updated survey in 2001 [Brusilovsky, 2001]. [Knutov et al., 2009] additionally mention the first reference model for AH applications [De Bra et al., 1999] and an implementation following this model [De Bra and Calvi, 1998], [De Bra et al., 2006] as further milestones in the evolution of adaptive systems.

The following paragraphs line out the most important adaptation methods and techniques as they are applied in adaptive web-based systems.

In general, we can consider and answer six major questions [Knutov et al., 2009]:

- What can we adapt?
- What can we adapt to?
- Why do we need adaptation?
- Where can we apply adaptation?
- When can we apply adaptation?
- How do we adapt?

Further, we can distinguish between different ways and levels of adaptation. The most common one is the level of *content*, i.e. the content itself is tailored to a user's requirements. Content-based adaptation includes, for instance, personalized access to material (e.g. documents) and personalized navigation paths. [Brusilovsky, 1996] introduces a categorization of adaptation technologies in AH, distinguishing between adaptive navigation support (cf. [Brusilovsky, 2007] for a detailed explanation) and adaptive presentation support. Adaptive navigation support includes ([Brusilovsky, 2007] and [Henze, 2000], based on [Brusilovsky, 1996]):

- Direct guidance, where the system provides a sequential path through learning material.
- Adaptive sorting, where links of a document are sorted according to their assumed relevance (based, for instance, on previous knowledge).

- Adaptive hiding, where links are hidden or disabled if the system assumes that they are not relevant or distracting.
- Link annotation, where links are annotated by text, colouring, an icon, or dimming in order to provide additional information to the learner.
- Map annotation, using annotation methods for adapting graphical overviews.

Adaptive presentation support includes ([Henze, 2000], based on [Brusilovsky, 1996]):

- Additional explanations, where the granularity of the content is fit to a user's current knowledge, goals, etc., as used by, for instance, MetaDoc [Boyle and Encarnacion, 1994], KN-AHS [Kobsa et al., 1994] or EPIAIM [De Carolis et al., 1993].
- Prerequisite explanations, where concepts that are prerequisites for the current content, but are not sufficiently well known, are explained (used in, for instance, C-book [Kay and Kummerfeld, 1994]).
- Comparative explanations, focusing on the comparison of current concepts with own ones (used in, for instance, ITEM/IP [Brusilovsky, 1992]).
- Explanation variants, needed where displaying or hiding parts of information is not sufficient (used in, for instance, HYPADAPTER [Encarnaçao, 1995]).
- Sorting, where fragments of information are sorted according to their relevance for the user (used in, for instance, HYPADAPTER [Encarnaçao, 1995] or EPIAIM [De Carolis et al., 1993]).

Note that although most of the example systems listed above are in the field of adaptive educational environments, there are several different application areas and systems, for example, online shopping (e.g., [Amazon.com, 2010]), social networks (e.g., [Facebook, 2010]), or other communities of interest (e.g., [movielens, 2010], [Pandora, 2010] or [Last.fm, 2010]).

[Knutov et al., 2009] introduce an extended taxonomy of adaptation techniques based on "old" categorization of [Brusilovsky, 1996], that further distinguishes between content adaptation techniques and adaptive presentation techniques. The first includes inserting/removing fragments and altering. The second mainly includes dimming fragments, sorting fragments, stretchtext, zoom/scale, layout, link sorting/ordering, link annotation, and combinational techniques. Adaptive navigation techniques in the new scheme, concentrate on link generation, guidance and link hiding.

Data Processing and Communication Approaches

This section introduces different ways of processing usage data in order to provide adaptive system behaviour and different kinds of adaptive support.

Server-Side Monitoring Traditionally, usage data was and still is, in most cases, collected by server-side monitoring and analysis of user behaviour, i.e. by interpreting HTTP requests of different resources [Hauger, 2008]. A resource in this case can be, for instance, a document or a document pool, an entry in a forum, the forum itself, a private message, a chat room, etc.

User interaction is in this case mostly restricted to click activities, i.e., a user activity is registered as soon as a click occurs, which then triggers the transfer of an HTTP request to the server, which again answers with an HTTP response and at the same time keeps the information about the requested resource in order to update the user's respective model.

As described in the previous section, this procedure includes both explicitly and implicitly provided information. This kind of information processing founds the basis of most adaptive (web-based) systems and is thus indispensable in the process of user modeling.

Examples of adaptive systems exclusively using server-side technologies are AHA! [De Bra and Calvi, 1998], [De Bra et al., 2006], KnowledgeTree [Brusilovsky, 2004], KnowledgeSea [Brusilovsky and Rizzo, 2002] and KnowledgeSea II [Farzan and Brusilovsky, 2005], or SIETTE [Conejo et al., 2004]. Some of these systems additionally consider computed values like the time a user spent viewing a specific resource.

Asynchronous Web Technologies For some applications where particularly fine-grained user models are necessary or at least advantageous, server-side monitoring proved to be insufficient. There, user interaction considered for the model building process would include client-side actions that are ignored in traditional modeling techniques.

For instance, it may be considered necessary to determine if a resource, for example, a page containing information, has really been read (and understood) or was loaded but then skipped. In such a case, relevant interaction data within this resource can be listed as follows [Putzinger, 2007a], [Putzinger, 2008], [Hauger, 2008]:

- mouse activity,
- keyboard activity,
- focus,
- time spent,
- scrolling, and
- partial or repeated reading.

As explained by Hauger, the *focus* can be indicative of whether a resource is really processed by a user. In his opinion, exceptions where a user, for example, reads in an inactive browser window while taking notes in an editor, can be neglected.

The *time spent* on a resource is undoubtedly an important factor that is however, not only considered in adaptive systems based on asynchronous technologies but is mostly added as a criterion in those based on server-side monitoring also.

Scrolling must, according to Hauger, be slow enough to allow reading to be an indicator. Furthermore, scrolling up a few lines could be indicative of a user not having understood parts of the text and rereading it.

Mouse and keyboard activity can give additional evidence of a page not being skipped. Hauger states, for example, that users may tend to move the mouse unintentionally and that the mouse cursor can also be a clue for determining the locus of attention. He further mentions activities like text copying, printing or pasting via keyboard shortcuts as indicators.

Systems in the area of adaptivity making use of the concept of asynchronicity in the web are labeled Asynchronous Adaptive Hypermedia Systems (AAHS). Putzinger introduces the concept of instant adaptation [Putzinger, 2007a], [Putzinger, 2007b] in such systems, that allows "page fragments to be dynamically exchanged according to the results of the underlying adaptation system".

Instant adaptation is different to traditional adaptation techniques because the latter is restricted to adapting content or other hypermedia elements when a page is created. Instant adaptation suggests adaptation of page fragments after the page has been sent to the client, by making use of asynchronous technologies like Asynchronous JavaScript and XML (AJAX).

[Hauger and van Velsen, 2009] pick up on page fragmentation (see also [Hauger, 2009a], [Hauger, 2009b] and [Hauger et al., 2010]) and instant adaptation. They further state that based on client-side interactions, a system could assume specific kinds of reading behaviour which could then become the basis for (instant) adaptation.

Conclusions on Data Acquisition and Processing Approaches We can conclude that asynchronous web technologies which have become more and more popular with the emergence of the "Web 2.0" [O'Reilly, 2005], will make their way into the most popular and widely-used adaptive systems, can contribute to more fine-grained user models and establish new perspectives in adaptation research in general.

However, it must be stated also that there is still a way to get there because assumptions must be proven, adaptation concepts must be tested before being ready for the market. For example, referring to the statement of [Hauger, 2008], who names a situation where a user is reading in an inactive browser window while taking notes somewhere else, a neglectable exception, we must consider such exceptions also concerning reading behaviour. It is possible that people exhibit automated activities like clicking, scrolling or selecting in a way that is not indicative for specific types of reading behaviour.

For what is being described in this thesis, the distinction between synchronous and asynchronous / server-side and client-side technologies is only marginally relevant because the focus here lies on the analysis and interpretation of (sequential) usage data and the concepts discussed here can be theoretically applied to arbitrary kinds of data.

Adaptive Support for Learning and Collaboration

Although until now possibly more popular in the area of e-commerce (see, for example, Amazon [Amazon.com, 2010]), research on adaptivity has been expedited in various fields as described in the following sections. This thesis focuses on the area of (collaborative) e-learning. Learning in general and collaborative learning in particular, are processes that can be very well supported through different means of assistance. Assistance can be provided via direct tutoring as found in traditional classroom settings, or, for example, via peer-group cooperation.

Many kinds of assistance can however be very well automated. Generally, we can distinguish between two kinds of support: *(a) learning support* and *(b) collaboration support*.

Learning support is mostly content-based, meaning that a learner should be assisted when problems are encountered, when a task is not understood well, when a solution is classified as wrong and an explanation should be given, etc. *Learning support* also means that a student is provided content or navigation elements that fit personal needs, which is usually not, or only in a restricted way possible in traditional instructor / learner settings because one instructor has to take care of a considerable number of learners which makes it difficult to offer personalized training for all.

As described in [Köck and Paramythis, 2010], adaptive learning support can be provided on the basis of different learning types. Students may differ, for example, in the way they seek for assistance which makes necessary different ways of providing assistance also. Adaptive learning support will be a main focus of this thesis and will be treated in detail in Chapter 7.

As already mentioned, adaptive support in learning settings is not restricted to content, it can also be provided on the basis of collaboration, meaning that users should be assisted in (a) the way they form collaboration groups and (b) the way they work together within a group.

Regarding (a), learners should be supported in "establishing ad-hoc collaboration amongst themselves", and in "forming groups with varying life spans and learning goals" [Paramythis and Mühlbacher, 2008]. These kinds of support include the facilitation of spontaneous interactions between learners that did not get in touch with each other before, or setting up structured group work.

Regarding (b), different didactic approaches can be employed to offer different ways of support for ongoing collaboration, like strategies for improved context or situation awareness [Endsley, 1995].

The actual appearance of adaptive support also depends on the learning content that is being imparted. [Sonntag and Paramythis, 2008], for instance, describe an e-learning scenario in the area of legal studies where a case generator provides adaptively composed content to the students.

Another example is given by [Weber et al., 2001], who describe NetCoach, an authoring tool used to develop adaptive learning courses without any programming knowledge. NetCoach implements adaptive link annotation and curriculum sequencing. The latter aims at individually planning a student's path through the learning content in order to fit personal learning characteristics, knowledge and preferences.

During the past years, when e-learning became popular and made its way into traditional learning settings, the need for standards and specifications arose also in the area of adaptive e-learning systems. Learning platforms started to integrate standardized learning packages and learning processes (see, for more details, Section 1.1.1). But not only the learning process itself should be prepared to fit a specific standardized format but also adaptive support for learning should be integrated into the respective standards / specifications [Paramythis and Loidl-Reisinger, 2003].

One of the most popular specifications in the area of learning is IMS Learning Design (IMS LD) [IMS Global Learning Consortium, 2011]. IMS LD is a specification that tries to model different pedagogies in the area of online learning. IMS LD is based on XML and models distinct learning activities as well as pedagogical approaches and takes into account collaborative learning settings. For instance, an IMS LD model contains persons, roles, learning activities and learning paths.

Collaborative learning can be in general described as any "*situation* in which *two or more* people *learn* or attempt to learn something *together*" [Dillenbourg, 1999]. Dillenbourg however points out that this definition is ambiguous as it can be interpreted in several different ways, as "situation in which two or more people.." could refer to a setting with 2-3 people as well as to a whole community or society, "learn" could refer to following a course as well well as to the lifelong learning process, and "together" could refer to direct face-to-face-contact as well as to a computer-supported setting, or to synchronous or asynchronous cooperation. Therefore, "collaboration support" can be provided in manifold ways and reaches from structuring the collaboration process to the regulation of interactions.

IMS LD integrates and computationally represents so-called "collaboration scripts" that set up, for example, "differences among learners in order to trigger contentious interaction" [König and Paramythis, 2010b]. A definition of a collaboration script can contain interaction and collaboration instructions related to the respective scenario. However, these scripts were traditionally non-dynamic and thus not flexible enough to fit the requirements of scripting in the context of adaptive collaboration support.

Recently (see, for example, [Paramythis and Cristea, 2008], [König and Paramythis, 2010a], [König and Paramythis, 2010b] or [König, 2010]) efforts have been made to develop extensions to IMS LD which make possible adaptive support for collaborative learning. [König and Paramythis, 2010b] add, for instance, the possibility for groups to be modeled dynamically at run-time, and the possibility for roles to be assigned to group-members at run-time. Furthermore, they suggest an extension of the specification of actions which originally provides only limited possibilities: showing/hiding objects, changing the value of properties and giving feedback.

König and Paramythis list the actions necessary in order for adaptation to be supported as follows:

- Object adaptations deal with adapting an object's attributes and its life cycle. An object in the context of IMS LD does not necessarily have to be a "learning object", but can, for instance, also be a person, group, environment, activity or event.
- Relation adaptations are adaptations affecting the relations between objects, i.e., the objects' attributes that are needed to establish links between them. A relation can, for instance, be "membership", linking a person with a role, "ownership", linking a person to a property, or "visibility", linking an environment to a learning object.
- Control flow adaptations refer to starting, stopping, or modifying (e.g. new branches) the script process.
- Environment adaptations refer to adaptations within IMS LD services. An IMS LD service can, for instance, be a communication, coordination or awareness service and is described by parameters like its type of interaction (synchronous or asynchronous, implicit or explicit), action coordination, modality of interaction (i.e., text-based, graphical or audio/video), or privacy mode. König and Paramythis describe IMS LD services as "black-boxes" that cannot be monitored or even manipulated, which would, however, be necessary in order to actively support collaboration.
- Adaptation control deals with managing the adaptations themselves, referring to, for instance, conditions, event handlers or actions.

The extensions to IMS LD are, at the moment, in the implementation phase, i.e., the definition phase has been completed. An executable model linked to the Sakai e-learning platform [Sakai, 2010] is being developed, which should be applied on real-world learning settings, which can then be followed by an evaluation of the approach. The work described in this thesis ties in with such efforts, in that it provides the observational evidence required to determine whether adaptation is necessary, and what it's nature should be.

1.1.3 Machine Learning and Data Mining

Adaptive systems particularly rely on the information they gain from the interaction with the user. Such data must, however, be processed and interpreted in a way that allows for drawing reliable conclusions regarding the user's individual requirements and needs. This leads to a need for *(a) a way of retrieving expressive data* and *(b) means to interpret the data.*

Thus, classic data mining and machine learning techniques can be utilized for the collection, interpretation, and processing of user activity and interaction data. A technique used for the interpretation of the data must be at least semi-intelligent as the information hidden in user activities is potentially too complex to be detected by simple, for example, statistical means. More "intelligent" approaches as well-known in the field of machine learning can help to retrieve both fine-grained and multi-faceted information about individual users and the relations and dependencies between them.

Machine learning is a field strongly related to the one of Artificial Intelligence (AI) or Computational Intelligence (CI) that has become popular already in the early 1980ies when researchers started to define learning in the context of non-human "intellect". An early definition of learning in this sense was provided by [Simon, 1983] who described it as "changes in [a] system that ... enable [it] to do the same task or tasks drawn from the same population more efficiently and more effectively the next time."

[Shavlik and Dietterich, 1990] build upon this definition and define two ways in which a system can "change". First, they mention a system's capability to acquire new knowledge from external sources, and second, they argue that a system can modify itself to "exploit its current knowledge more effectively".

Further, Shavlik and Dietterich derive two different kinds of learning from the two ways of system change: inductive learning and similarity-based learning. The first kind of learning can be further subdivided into *supervised learning* and *unsupervised learning*, the second of which will be a substantial part of the work described in this thesis.

There are also hybrid types positioned between supervised and unsupervised learning known as semi-supervised learning (see, for example, [Zhu, 2008] or [Goldman and Zhou, 2000]).

Supervised Learning

Supervised Learning can also be described as guarded learning or controlled learning because the learning process is largely driven by pre-labeled data providing reference information to the system, i.e., training data is used to deduce the functions used to label new data. Thus, in a supervised learning setting, the system receives as information not only input data but also the desired output for a specific number of training instances. Therefore, this kind of learning is well applicable for all problems where the desired output can be clearly defined beforehand. Supervised learning is mainly applied for classification problems where data instances should be assigned to a particular "class", based on example classifications. The example instances are classified based on the definitions of all possible classes, usually provided by a human operator.

Example tasks for supervised learning are, for instance, text or e-mail categorization, face detection, speech recognition and synthesis or sound classification. In e-mail categorization, the system's task could be to determine whether an e-mail should be regarded spam or not, or to estimate an e-mails level of importance for the recipient (see, for example, [Ayodele et al., 2010]). Similarly [Goren-Bar et al., 2000] aim at the classification of text documents.

Face detection or, more generally, visual object recognition is a task relevant not only for social networks where a system automatically detects faces on pictures uploaded by the users, but can also be used to detect faces on videos, for example, by a surveillance camera, or by next-generation cars that can automatically avoid collisions with objects appearing on the street.

[Heisele, 2003] introduces an approach to detecting objects on videos that is robust against changes in the object's pose or illumination. Speech synthesis, i.e., the conversion of text to human vocal output, or the other way round, with the goal to convert voice input to text, has been in the focus of research interest for a long period of time.

We can find early ideas hundreds of years ago when people started to try to build machines that create human speech. These ideas became more concrete at the beginning of the 20th century and of course, became popular with the emerging fields of electronic signal or natural language processing.

An example suggesting supervised learning as a basis for speech imitation is provided by [Howard and Huckvale, 2005]. Sound classification is another possible supervised machine learning task that has always been attractive for researchers in computer science and mathematics [Widmer, 2006].

Related work is, for example, provided by [McKay and Fujinaga, 2009], who introduce a free software for automatic music classification, [Ezzaidi and Rouat, 2006], who describe an approach to musical genre classification, or [Eck, 2007], aiming at automatically tagging music. These examples show that the application field for supervised learning techniques is quite broad. It can furthermore be also applied in the scope of (adaptive) e-learning, for example to classify student activities (see Section 1.2 and later ones in this thesis).

Supervised learning comprises a wide range of different algorithms. Mostly, the decision for a specific algorithm is based on the concrete application scenario because often the algorithms perform differently on different kinds of data. Therefore an overall comparison of algorithms like provided by [Baeza-Yates and Ribeiro-Neto, 1999] can be difficult as the results strongly depend on the characteristics of the data.

Examples for supervised learning algorithms are Bayesian Networks (BNs) and Naïve Bayes (NB), Artificial Neural Networks (ANNs), tree- or rule-based approaches, Support Vector Machines (SVMs), or Nearest Neighbour (NN) approaches. All these algorithmic approaches have in common that they are used to label unclassified test data based on models they retrieved from already classified training data. They are however, significantly different in the way the respective models are built and organized.

Tree-based classifiers, for example, build up decision trees as models that can be described as hierarchical graphs with exactly one root and several branches representing the different possibilities for decisions, each pointing towards a consequence the preceding decision would lead to.

Rule-based classifiers derive from the already classified input data a set of rules that, when applied to unlabeled data, become the basis for the decision of the best-fitting class.

Bayesian Networks use (directed, acyclic) graph-based models that represent variables and conditional dependencies between them based on Bayes' theorem (see, for example, [Pawlak, 2000]) to determine these values. *NB* can be described as the simplest form of a BN, assuming all variables as independent from each other.

SVMs are based on mapping data into a higher-dimensional feature space and representing data instances as feature vectors. The approach aims at finding the best linear separator (here, a multi-dimensional hyperplane) between the classes.

Nearest Neighbour classifiers make their decisions based on the majority vote of an instance's nearest neighbours, mapping the data instances into an n-dimensional space and determining the distance between data points via classical distance metrics like the Euclidean distance [Black, 2004].

Artificial Neural Networks are simplistic models of biological neural networks as can be found in the human brain. Basically, an ANN is made up of several layers of partly interconnected nodes. The network's output is the result of a so-called network function which is composed by other functions, for instance, whether a data instance passes through a specific node is determined via the node's activation function.

Unsupervised Learning

In contrast to supervised learning, where it is the goal to detect predefined classes in given input data, unsupervised learning aims at the identification of classes without any previous information on their nature and without any rewards from its environment [Ghahramani, 2004]. Thus, unsupervised learning can be well applied for pattern discovery, including the distinction between meaningful patterns and random noise, and to find out in what way data is organized.

Unsupervised learning often aims at finding clusters in the input data provided to the system with a cluster being the consolidation of, in some way "similar" data. The cluster description can then become a "class" as it is used in supervised learning. Unsupervised learning generates *descriptive models*, in contrast to the *predictive* ones in supervised learning. The need for human participation is postponed until the end of the phase of clustering here, whereas in supervised learning, it is needed beforehand to classify training data.

To name an example, [Zancanaro et al., 2007] describe an unsupervised learning approach used to analyze patterns in museum visitors' behaviour, ultimately aiming at the personalization of information presentation for the respective visitor. Therefore, the system logged the visitors' movements in the museum space and later clustered the logs in order to detect patterns.

A different approach is described by [Morris and Trivedi, 2009] who use unsupervised learning of motion patterns in order to increase awareness to a vehicle's surrounding, ultimately aiming at increasing driving safety. Morris and Trivedi describe how natural observation during driving can become the basis for automatically learning behaviours of surrounding vehicles.

[Corral et al., 2007] apply unsupervised learning to analyze data security by identifying and quantifying potential vulnerabilities in a system. Unsupervised learning is part of a hybrid approach there and was found useful for the extraction of implicit, previously unknown information.

Also in e-learning, unsupervised learning approaches are well applicable; it may, for instance, be desirable to gain information about learners, their behaviour in moving through learning content or choosing tasks. For instance, a learning system may want to cluster learners according to their way of solving specifically structured tasks, or, to find similarities and differences in general learning behaviour. This kind of information can later become the basis for individual adaptation or recommendation of group constellations, as described in more details later in this thesis.

Like in supervised learning, there is a range of different algorithms in unsupervised learning. As this thesis concentrates on clustering in the context of e-learning, clustering approaches are most relevant here. Examples for clustering algorithms are kMeans, Fuzzy cMeans (FCM), (Gaussian) Mixture Models ((G)MMs), or Self-Organizing Maps (SOMs).

The kMeans algorithm (see, for example, [Forgy, 1965] for an early description) is based on the observation that the "optimal placement of a center is at the centroid of the associated cluster" [Kanungo et al., 2002], [Faber, 1994], [Du et al., 1999]. Around these centroids there is a number of data points for which this centroid is the nearest neighbour.

Thus, kMeans is a distance-based approach that aims at maximizing the distance between the cluster centroids and minimizing the distance within the clusters. It therefore requires a metric to determine the distances between data points. The clustering process starts, e.g. with a randomly chosen set of centroids and assigns the data points to the center nearest to them. Next, the process computes new centroids in the clusters by averaging over the values of the data points in the clusters. Further, it is evaluated whether the new constellation is an improvement over the one before. As long as there is improvement, the process repeats instance assignment to the nearest centroids.

The FCM approach (see, for example, [Dunn, 1973], [Hathaway and Bezdek, 1988]) is different in that it allows an element (i.e., data point) to appear in more than one clusters. Furthermore the relationship between a data point and a cluster is not simply a binary "is in" and "is not in" one - data points can be in clusters to a certain degree. Thus, data points situated at the edge of a cluster may be in the cluster to a smaller degree than the points close to the center of the cluster. Regarding the general process, FCM is rather similar to kMeans.

Mixture models (see, for example, [Lindsay,]) are based on each cluster being mathematically described by a parametric distribution and on the data set being modeled as a mixture of the distributions. This approach aims at not only being able to describe association of given data points with clusters and output a *descriptive* but also *generative* model of the data in order to be used to compose new data points following the given distribution. One popular variant of MMs is the GMM assuming a Gaussian (normal) distribution. GMM unsupervised learning is based on a hypothesis about the number of Gaussian components behind the produced data and an estimation of the likelihood for models to have produced the output. The process then aims at finding the model that is most likely to have produced the given data.

SOMs [Kohonen, 2001] are variants of ANNs that produce a typically two-dimensional map (i.e., model) preserving the topological properties of the input space, meaning that similar clusters should be positioned next to each other on the map. Thus, SOMs are "self-organizing" in that the concrete structure of a map is determined by the process and the topological characteristics of the data, rather than being predefined.

A SOM basically consists of nodes and connections between them, where the nodes represent cluster centroids and data items are mapped to the node they have most similarities with (i.e., the smallest distance to). The clustering approach used by SOMs is a two-level one, i.e. first a number of prototypes ("proto-clusters") is formed that is larger than the expected number of clusters. In the next step these prototypes are combined to constitute the actual clusters [Vesanto and Alhoniemi, 2000]. Every data instance belongs to the same cluster as the nearest prototype. SOMs are well applicable and frequently used for data and topology visualization.

1.2 Goals and Outline of Work

Not only has research attention in the general idea of adaptivity been rapidly growing during the past decades, also has the domain of e-learning, as an application domain for personalization, moved into the focus of adaptive systems research.

Although adaptive e-learning received high attention during the past years, it has, for a long while, not been treating learning as a social process (see, for instance, [Brusilovsky et al., 2004]). Furthermore, learning activities have long been treated as unrelated attempts to (actively or passively) consume learning content (either as an individual or a group), rather than as interconnected with each other. Only recently, these connections have been taken into account (see, for example, [Soller and Lesgold, 2007]), which led to an altered picture of the learning process, consisting of interrelated sequences of actions.

Generally, the diversity of learning settings leads to great variety in learner behaviour, which makes analysis, needed in order to gain information to feed into the learner models, a complex task that cannot be approached in a standardized manner any more.

The work reported in this thesis

- builds upon the premise that patterns in user behaviour are indicative of more general behavioural "styles" that users commonly exhibit,

- assumes that these patterns can be captured by analyzing user interaction data with machine learning and data mining methods,

- emphasizes the significance of sequential information in user interaction data for its expressiveness regarding the recognition of behavioural patterns and their interpretability, and

- suggests the information about behavioural styles in the domain of learning, and, more specifically, problem-solving as a basis for more fine-grained individual user support in adaptive learning environments.

This thesis, in the context of the ASCOLLA project, describes different ways to gain and interpret interaction information by means of machine learning and data mining methods. As already introduced, the ASCOLLA project focused on adaptively supporting the collaboration process in learning contexts, treating, for example, learner modeling, modeling of didactic approaches, or adaptive support for collaboration establishment.

The thesis comprises a general description of how learner activity data can be utilized in the learner modeling process – online learning activities are monitored and interpreted in order to enrich the user models and, ultimately, make possible new and expanded forms of adaptive intervention in the context of e-learning. Different methods (with the focus lying on clustering) are applied for information extraction, however, sharing one overall aim: a profound basis for supporting individual and group learning processes should be provided.

The main focus here lies on the fact that learner activities cannot necessarily be treated as independent from each other, but might be interrelated. Thus, a sequence modeling method based on Discrete Markov Models (DMMs) that specifically captures problem-solving behaviour, is introduced. Problem-solving is an essential part of the learning process, in traditional learning settings as well as in the context of e-learning. Problem-solving behaviour can, similarly to learning in a more general sense, be described by different concrete styles. Learning styles (see, for instance, [Lefrancois, 2006]), are probably better known and more often considered than problem-solving styles and describe behaviour at a different level, however, the definitions of both are rather similar.

Learners' problem-solving styles can only hardly be analyzed without considering detailed *sequential* information. Thus, their analysis renders an appropriate case study for the sequence modeling and interpretation approach proposed in this thesis. Experiments were run using real-world interaction data monitored by an Intelligent Tutoring System (ITS) that were converted into problem-solving sequence DMMs. These resulting sequence models were then, partly automatedly, analyzed in order to discover semantically meaningful information about learners, particularly focusing on their problem-solving behaviour, that could in turn be reintegrated in the adaptation cycle. The clustering process used for the purposes of sequence analysis, is applicable at three levels, depending on the respective discovery goal:

- *Level I (**pattern-driven**)* aims at the detection of predefined behavioural patterns related to problem-solving in learners' activity sequences. These patterns (i.e., styles) may be derived from well-established, well researched educational theories and problem-solving "prototypes", i.e. approaches frequently observed in the behaviour of learners.
- *Level II (**dimension-driven**)* aims at semi-automatically detecting concrete, but still unknown, styles related to behaviour associated with a specific learning or problem-solving dimension. A dimension can, in this context, be understood as a set of attributes relevant to learning behaviour, that, however, still leaves open the concrete values for these attributes.
- *Level III (**open discovery**)* aims at the open-ended detection of potential learning / problem-solving dimensions and concrete styles within these dimensions. Human intervention in the process should be reduced and should concentrate on the assessment of the validity of the system's findings. Thus, clustering at this level is largely driven by the system that takes over the task of evaluating intermediate clustering results. Furthermore, the system predicts the potential of these results to be sufficiently descriptive in order to derive concrete styles and more general dimensions in the latest phases of the process.

The rest of this thesis is structured as follows. Chapter 2 provides an overview of related work, i.e. a state of the art analysis. Chapter 3 provides an overall view on adaptive support based on different machine learning approaches and, on one hand, identifies the relations between them, and on the other hand, explains how they all contribute to the same big picture. This chapter furthermore describes the data used for the process later.

Chapter 4 describes a classification-based approach to the prediction of potential user interests. This chapter provides a view on the application and the potential of supervised learning in adaptive systems in order to draw conclusions utilizable for later, e.g., recommendations.

Chapters 5 and 6 present the unsupervised learning approach to drawing conclusions about learners, which lies in the main focus of this thesis. Chapter 5 introduces a sequence-based activity modeling procedure, Chapter 6 presents a novel multi-level clustering approach as applied to the models previously derived. Furthermore, the results of the different clustering stages are discussed and interpreted, and later used as a basis for adaptive behaviour in Chapter 7 that describes system interventions aiming at closing the adaptation cycle.

Chapter 8 discusses privacy and security issues in personalized systems, and Chapter 9 summarizes findings, discusses issues related to the application of the proposed approach to different domains, and gives an overview of potential future work.

This thesis also contains the work published in [Köck and Paramythis, 2011], including most of the tables and figures in Section 5.3 and Chapter 6. Small parts of the text in [Köck and Paramythis, 2011] were written by the second author, Alexandros Paramythis and remained in the text of this thesis for reasons of completeness.

Chapter 2

State of the Art

This chapter provides an overview of the state of the art in the related fields of research. Parts of the findings presented in this chapter have been previously published also in [Köck and Paramythis, 2011]. As already shortly outlined, the approach presented here utilizes partly novel machine learning and data mining techniques in the scope of adaptive systems in order to provide personalized learner support on one hand and to reduce the amount of human effort required throughout the whole process to a minimum on the other.

2.1 Machine Learning and Data Mining for Personalization

As introduced in Chapter 1, machine learning and data mining techniques are commonly used to collect, process, and interpret large amounts of data, for example, as in the concrete case discussed in this thesis, data implicitly produced by users when interacting with a system.

As personalized systems not only rely on the information explicitly provided by their users but also, and most importantly, on information that is implicitly gained from user-system and user-user interaction, this field is a broad application area for machine learning and data mining. However, not all adaptive systems make immersive use of, in a way "intelligent", techniques but rather tend to prefer simpler statistical analyses. This section presents and discusses some examples in the general area of personalization that use machine learning and / or data mining techniques.

[Anderson, 2002] introduces the MONTAGE system building personalized visitor web portals that combine content and links from multiple web sites into one individual view for every user. This kind of personalization is based on the prediction of users' goal and does not modify existing content but creates views on the content. Anderson presents three hypotheses that were later used in the design and implementation phase of the system:

- users want quick access to their routine browsing destinations,
- routine web browsing tools benefit from tailoring links and views to the browsing context, and

- past web behaviour patterns can be used as a basis for the prediction of future browsing.

The MONTAGE system groups contents by topics and links every topic to a topic-specific personalized "montage", i.e. view. This topic-specific montage then aggregates content from various locations. In this system, activity data needed for personalization is mainly collected by monitoring users' page request sequences, including the respective page's URL and the topic of the page's content. The assignment of topics to content is done by a classification approach (i.e., supervised learning) using an SVM.

Basically, the system then computes predicted values for the following five aspects of the user model: *candidate pages* (a subset of the previously visited pages is selected), *interest in a page* (based on how much time the user spent on the page), *interest in a topic*, *probability of revisit* (given the current context), and *savings possible* (meaning that the system favours pages where a manual revisit would be difficult or cost more effort than for others).

The MONTAGE system directs all web browsing through a proxy in order to retrieve the required data. Another discourse on web personalization is provided by [Eirinaki and Vazirgiannis, 2003] who do not only summarize different aspects of web mining in general (for instance, user profiling and data collection), but also list a number of (popular) web sites offering personalization, including Amazon [Amazon.com, 2010], CDNOW [CDNOW, 2010], and Food.com [Food.com, 2010].

A different view on machine learning applied for personalization purposes, although not in the area of the web, is presented by [Ypma et al., 2007] who introduce real-time hearing aid personalization based on Bayesian feature selection. The aim is to personalize a hearing aid during usage in order to fit the individual user's needs. Therefore, the algorithms applied must be able to learn preferred parameters from user interaction.

Another approach is described by [McBurney et al., 2008], who present adaptation of pervasive environments based on machine learning and dynamic personalization. They discuss a general approach to dynamically model user preferences based on monitoring activity data.

Personalization is integrated into several tasks like, for instance, *service selection*, where preferences are used to select the best fitting service, *service parameterization*, where services are configured using the preference outcome for the current context, *session adaptation*, meaning that sessions are dynamically manipulated (i.e., paused or transferred), and *network selection*, aiming at the selection of the most appropriate network for an individual user [McBurney et al., 2008]. The preference modeling process utilizes a decision-tree-based algorithm. The authors present muting a VoIP service as an example scenario for their dynamic preferences approach.

[Zimmerman et al., 2004] present an approach to personalized television based on individual recommendations of TV shows. Their approach involves a recommender engine that, based on tracking users' preferences, individually recommends television content.

Additionally, it includes automatically created explanations for these recommendations, aiming at gaining and keeping users' trust in the system. The recommendations are based on two different information sources, explicit and implicit user input that make up two different recommenders. Neural networks are then applied for fusing both recommenders' outputs.

Another example for the use of machine learning / data mining techniques for personalization is given by [Das et al., 2007], who describe an approach to generating personalized recommendations for users of Google News [Google News, 2010]. Their recommender uses three different approaches, for instance, collaborative filtering based on a clustering process. An evaluation demonstrated the system's scalability and also showed that the process can handle live traffic of Google News. Furthermore, Das et al. claim that their approach is easily extendable to other domains.

[Mandel et al., 2006] discuss a system for "performing flexible music similarity queries using SVM active learning" that aims at facilitating searching and organizing digital music collections. During an evaluation period, the system was used for the classification of different pop songs according to the factors mood, style and artist. Although Mandel et al. do not explicitly discuss their approach's applicability for personalization, it could be well extended and applied for the organization and personalization of individual users' music collections, based on a user's ratings.

While this section presented some examples for system personalization based on the application of machine learning / data mining techniques, the next section will explicitly discuss related work in the area of "intelligent" adaptive educational systems.

2.2 Machine Learning and Data Mining in (Adaptive) Educational Systems

Educational Data Mining (EDM) or data mining in e-learning [Romero and Ventura, 2006] is a broad field of research that combines aspects and issues of different areas (e.g., e-learning / distance education, machine learning, adaptive systems, etc.).

[Romero and Ventura, 2007] categorize work in EDM into *(a) statistics and visualization*, and *(b) web mining* [Srivastava et al., 2000], [Mobasher, 2005] that can be further split into *clustering, classification and outlier detection*, *association rule mining and sequential pattern mining*, and *text mining*.

Web (usage) mining can additionally be further categorized into *offline web mining* aiming at the discovery of patterns or other information to help educators to validate learning models, and *online or integrated web mining* where the patterns that are discovered are fed into an "intelligent" system that could assist learners in their online learning endeavors [Li and Zaïane, 2004].

A different viewpoint on EDM is provided by [Baker and Yacef, 2009] and [Baker, 2010], who identify the following categories: *prediction*, including classification, regression and density estimation, *clustering*, *relationship mining*, including association rule mining, correlation mining, sequential pattern mining and causal data mining, *distillation of data for human judgment*, and *discovery with models*.

The focus of the work described in this thesis lies on *web usage mining*, especially emphasizing *clustering* and *sequential pattern mining* in the context of *student modeling*, based on the analysis of *logged user activity data*. Besides the different ways of categorization, the process of data mining in educational settings can be split into the following phases [Romero et al., 2007]: *data collection, data pre-processing, application of data mining,* and *interpretation, evaluation and deployment of the results.*

2.2.1 Educational Data Mining in General

Data acquisition and pre-processing are fundamental steps in the process of EDM and also constitute the first phases of the adaptation cycle. The nature of the data monitored is a decisive factor for the later stages in the cycle and further analysis. Most adaptive educational systems have in common that they strongly rely on this early phase of the process. They might differ, however, in the way data is actually monitored, and the granularity of the data itself. For instance, systems may treat user activities as individual items (either in an aggregated or event-based way) [Amershi and Conati, 2009], [Romero et al., 2008] or consider activity sequences [Soller, 2007], [Soller and Lesgold, 2007].

A further distinction can be made by the way data is analyzed later; during the past years a trend towards the combined use of data mining and machine learning techniques for the analysis of activity data can be observed [Romero and Ventura, 2007], [Romero et al., 2007], [Hämäläinen et al., 2004], [Amershi and Conati, 2009]. Systems based on individually treated user activities often aim at either the prediction of students' success (or, even more concrete, grades) [Romero et al., 2008], or future behaviour or interest [Köck, 2009], or at the extraction of individual users' and groups' characteristics [Choi and Kang, 2008].

[Romero et al., 2007] and [Romero et al., 2008] describe a data mining process driven by an extension to the Course Management System (CMS) Moodle [Moodle, 2010]. The original pool of logged data contains very fine-grained activities, e.g., every single click a user makes for navigational purposes.

However, data is not analyzed in its original granularity but summarized and thus converted to a more aggregated format (e.g., the number of assignments done by a student, the number of quizzes failed, the number of quizzes passed, the total time spent on assignments, etc.). The ultimate goal is the evaluation of the usefulness and performance of different classification algorithms for the prediction of students' final grades.

Another perspective is provided by [Choi and Kang, 2008] who monitor and analyze learner activity data in order to identify conflicting and facilitating factors on online collaborative learning. Conflicting factors are described as factors ultimately obstructing the achievement of learning objectives. Facilitating factors are described as elements that learners recognize as positive or supportive in attaining the learning objective.

Choi and Kang introduce an approach that, compared to the previously described one, relies more on semantic information behind user activities. In general, all activities are monitored; analysis, however, extracts the relevant parts and predefines common learner behaviours as, for instance, “summarize learning material”, “outline tasks”, “modify material”, or “write meeting minutes”.

[Vialardi et al., 2009] describe another data mining approach in the context of educational systems that aims at predicting how suitable a specific course is for a specific student (based on the system’s prediction of success for the respective course) via classification, in order to provide personalized recommendations. Unfortunately, the authors don’t provide a detailed description of the base data records they use. From the rules generated by their classification system (including, for instance, the number of courses a student is enrolled at), however, it can be concluded that they operate with accumulated data that is better comparable to what is described in [Romero et al., 2008] than to what is proposed in this thesis.

A conceptually related approach is presented by [Su et al., 2011] who describe a clustering- and decision-tree-based approach to eliciting appropriate learning content (objects) to provide learners with specific requirements and learning/interaction contexts. This approach is specifically intended to match so-called “user requests” for content (which encapsulate hardware capabilities, learner’s preference, and network conditions), to content elements in a learning object repository.

[Anaya and Boticario, 2009a] explore data mining in educational systems with particular focus on collaborative learning processes. They worked with students of the National Distance Learning University in Spain (UNED), using the learning environment dotLRN [.LRN, 2010]. The authors define their goals as revealing learners’ collaboration, being domain-independent and offering the information immediately after the process has finished. Their approach operates with statistics of interactions in forums, thus not considering semantic information. The students were given access to more tools than the forum only, such as FAQs, news, calendar, etc. The statistical indicators were used as a basis for clustering that should reveal information about learners’ collaborative behaviour. A paper by the same authors [Anaya and Boticario, 2011] presents an updated and more comprehensive view of their approach, introducing metrics based on the statistical indicators, which are shown to have superior performance to clustering in characterizing the collaboration behaviour of learners.

[Beal et al., 2006] discuss an approach to classification of learner engagement based on multiple data sources. As opposed to the approach described later in this thesis, which uses learner activity data only and does not rely on human efforts during the monitoring process, the one of [Beal et al., 2006] explores an integrated way of information acquisition, comprising also students' self-reported motivation profiles and teachers' ratings.

2.2.2 Student Modeling Based on Clustering

This section explores a more concrete part of related work that describes clustering in the context of student modeling (see an even more specialized selection in Section 2.2.3). [Amershi and Conati, 2009] provide a detailed description of their classification and clustering approach to user modeling. Their base data originates in a learning environment more exploratory than traditional tutoring systems, with students being required to have a deeper, more structured understanding of concepts in the domain, following the principles of constructivism (see, for example, [Piaget, 1954] or [Ben-Ari, 1998]). The data they use is converted to feature vectors that are later fed into the clustering phase; a feature vector represents an aggregated version of a student's activities. Thus, there is only one feature vector for each student, which results in a low overall number of vectors. Amershi and Conati describe another similar approach [Amershi and Conati, 2006] where clustering is used to automatically recognize learner groups in exploratory learning environments.

A clustering approach based on collaboration behaviour is provided by [Anaya and Boticario, 2009b] who describe how statistical indicators in learner activity data are used to determine cluster membership. Data was monitored for UNED students via the platform dotLRN [.LRN, 2010]. The described monitoring process started with an initial questionnaire and a mandatory individual task that has to be completed by every learner. The respective results were then used to manually group the learners into teams of 3 members each. In a later phase the teams were given additional tasks, where, for instance, every member had to solve one part of a specific problem, or the team had to merge individual solutions. An expert observed these processes and used the findings on learner collaboration to label statistical data (i.e., an aggregated version of logged learner activities) that was then fed into a clustering algorithm with the objective of revealing relations between the statistical indicators and collaboration behaviour.

2.2.3 Sequence-Based Approaches

This section summarizes clustering approaches in the context of e-learning that consider sequential information in activity log data and are therefore best comparable to the work described in this thesis. In machine learning, the use of Markov models is prominent if the domain requires sequences to be represented or analyzed as they provide a convenient way of modeling interrelated data. In the area of EDM, this advantage has recently been exploited in different pieces of research.

For example, we can find collaboration analysis based on Hidden Markov Models (HMMs) by [Soller and Lesgold, 2007] and [Soller, 2007]. [Soller and Lesgold, 2007] describe the modeling process for the example case of knowledge sharing, defining a knowledge sharing episode as "a series of conversational contributions and actions that begins when a group member introduces new knowledge into the group conversation, and ends when discussion of the new knowledge ceases".

The subsequent analysis aims at determining role distribution (knowledge sharer vs. receiver), analyzing how well the knowledge sharer explained the new knowledge and evaluating how the receiver assimilated new knowledge. The communication interface via which the activity sequences are logged, includes tagging functionality that helps to categorize individual activities. The tagging process is a manual one, i.e., it requires human effort. In the experiments Soller and Lesgold describe, trained HMMs provide a very good accuracy at identifying the role of the knowledge sharer. It is, however, not entirely clear why hidden models were used, as the number of states is known in advance. For the work described in this thesis, DMMs with a predefined number of states (indicated by the learner actions possible on the platform) are used.

Another pattern detection approach is described by [Beal et al., 2007], who use HMMs to model students' performance on problem-solving. The models are fit to students' activity sequences with three hypothesized hidden states that correspond to students' "engagement levels". The resulting HMMs are later used to cluster students into groups showing specific kinds of behaviour. Furthermore, the models become a basis for prediction at a later stage in the process. In this case, HMMs are obviously a well-fitting analytic approach because they are used for explicitly modeling unobservable influences, as also indicated by the better prediction accuracy, compared to simple Markov chains.

A different sequential pattern mining approach is described by [Perera et al., 2009] who exploit activity data monitored by the system to support mirroring, i.e., to extract and present patterns that characterize the behaviour of successful groups. They do not restrict the available tools or provide specific rules about how to use them, but aim at monitoring collaboration processes that are as authentic as possible, including the selection of tools and frequency of use. The main goal of this work is to "extract patterns and other information from the group logs and present it together with desired patterns to the people involved, so that they can interpret it, making use of their own knowledge of the group tasks and activities". The underlying concepts are based on the "Big Five" theory of group work [Salas et al., 2005] that defines five key factors: leadership, mutual performance monitoring, backup behaviour (e.g., reallocating work between members), adaptability, and team orientation.

After the monitoring process, the final data pool contained both the traces of user actions and the groups' progressive and final marks. Based on their performance (i.e., grades), the groups were ranked. The ranking was then used to determine what kind of behaviour distinguished the stronger from the weaker groups. This was done by a simple statistical analysis in the first phase, and by application of data mining techniques (clustering on groups and on students) in subsequent ones.

Group-level clustering base data contained aggregated group activities like, for instance, the average number of events in a specific tool, student-level clustering base data contained similar information for individual students. The clusters detected during the student-level clustering phase define different types of users, e.g., “managers” or “loafers”. The analysis of the activities by a pattern extraction process revealed the most important activities that were indicative of “strong” and “weak” groups. The authors point out, however, that their approach is not fully matured yet and still bears some limitations based on the data (due to limited types of events) and on the way in which output was interpreted.

[Jeong and Biswas, 2008] present another approach to behaviour modeling. They describe a study with middle school students operating with a Teachable Agent (TA). They again use HMMs to represent sequences of activities in order to reveal patterns that lead to learning success. The concrete goal in this case was to find out if “Learning by Teaching” provides better opportunities for learning, compared to other settings (a self-regulated and a coaching system). Thus, the sequence models were used mainly as an aid to evaluate learning concepts in this approach.

[Li and Yoo, 2006] describe the modeling of student learning behaviour with Bayesian Markov Chains that was used in the adaptive tutoring system AtoL [Yoo et al., 2005]. Their approach presumes a specific format of tasks including specific levels of difficulty and only considers two base observations, i.e., correct and incorrect answers (partly comparable to our states). Additionally, Li and Yoo assume that there are exactly three basic student models based on the three learning types they define (i.e., “reinforcement type”, “challenge type” and “regular type”).

Their goal is to use clustering for modeling student behaviour and to use the resulting models to predict the learning styles of new users. The results are interesting in the context of the work described in this thesis, because they show that basic sequential information can be successfully used in clustering processes and improve static models derived, for instance, through an initial survey. What is being proposed in this thesis, however, goes one step further and does not presume a specific number of models but rather aims to allow these to be dynamically determined.

[Zhou and Xu, 2010] discuss a general idea of applying different kinds of constraints that can be used to extract useful patterns. They state that commonly, sequential mining returns a huge number of patterns but usually only a part of them is educationally meaningful. They therefore suggest the injection of pedagogical constraints into pattern mining and list the following types:

- *Time Constraints* can be used if data is time-stamped. This kind of constraint could, for example, define a sequential pattern to be longer or shorter than a specific predefined period.

- *Length Constraints* are used to specify the length of sequential patterns, i.e., the number of activities involved.

- *Action Constraints* limit the types of actions that can appear in a pattern.
- *Context Constraints* define a certain context to which a pattern is linked. A context could be any form of location within a system, e.g. a document, a paragraph or a forum topic.
- *Distance Constraints* specify positional or temporal relationships between actions.
- *Super-Pattern Constraints* can be applied to extract patterns that contain specific "sub-patterns", which can be useful for the detection of relationships between patterns.

Although these constraints could be injected during or after the mining process, Zhou and Xu only apply the second option; in first experiments the approach was successfully applied to real usage data.

[Popescu, 2009] describes another approach to the analysis of behaviour considering activity sequences and discusses the role of learning styles. She further conducts a study based on a Unified Learning Style Model (ULSM) that analyzes user behaviour in order to detect meaningful patterns in learning. The following 6 dimensions based on 12 learning preferences are considered:

- visual preference / verbal preference,
- abstract concepts and generalizations / concrete, practical examples,
- serial / holistic,
- active experimentation / reflective observation,
- careful with details / not careful with details, and
- individual work / team work.

Data acquired for the analysis includes basic statistical information like the time spent on a task or the number of actions performed, and additionally comprises simple sequential information in the form of instructional role sequences.

The approach was tested in a setting with 75 undergraduate computer science students. The experiments showed that the behaviour of students with different ULSM preferences is different also in a web-based educational system. To show that this kind of learning preferences does not only apply for face-to-face but also online settings was one of the main purposes of the reported line of work. Thus, Popescu's purpose for this line of work differs from the one followed in this thesis. Yet, the results are relevant because they show that the analysis of learning activities considering sequences is possible.

Chapter 3

Facets of Data Analysis and Adaptive Support – Pieces of a Puzzle

The previous chapters introduced the motivations behind the idea of adaptivity in general, and in the domain of e-learning in specific, and provided an overview on methods, concepts and technologies that have been traditionally and recently implemented in this area towards the goal of ultimately providing personalized support for learners.

The following chapters describe an approach of how learner activity data can be modeled, analyzed and interpreted in order to reach the goals defined in Section 1.2. Ultimately, a fine-grained analysis of learner behaviour should contribute to better tailored adaptive support in the area of e-learning.

This chapter provides an overview on different methods to implement adaptive support and explains how they can fit together. The methods described here are all based on machine learning techniques but pursue different subgoals that all contribute to one overall aim. The chapter further introduces the data sets which will be used for experiments and theoretical examples described later in Chapters 4, 5 and 6.

3.1 Adaptation Techniques and Purposes

As already introduced in Section 1.1.2, adaptation can target different system components, as, for example, content, navigation, general interaction or, on a higher level, collaboration. This thesis focuses on providing a basis for *individual user support* and *adaptive collaboration support* (see Chapters 5 to 7) and also contains a description of an approach to the prediction of user interests (see Chapter 4) with navigation and content adaptation as the ultimate goal. In general, in order to provide different kinds of adaptations, the system must be able to infer information about a user, groups of users and possibly also the relations between them. This can be achieved by

- predicting a user's interests, characteristics, preferences, and in general, behaviour,

- predicting a user's belonging to a specific group like a group of users exhibiting a particular learning or problem-solving style,
- predicting a user's relations to other users, or
- predicting a user's interest in specific tools or types of content.

In addition to different goals and different prediction aims, two more criteria must be considered in the design phase of an adaptive component. First, in some cases, very detailed information about the users of the system is available – for instance, in cases where the same course has been conducted several times and analyses of usage data have shown that users' behaviour changes only marginally across various semesters, or in cases where users have already taken several different courses on the platform, whereas in other cases the system does not know anything about its users. This can crucially influence the decision on how usage data is analyzed and evaluated. If a lot of information is available or the prediction task is trivial (i.e., for example, binary), it might be possible and efficient to preliminarily define the possible prediction outcomes.

In the case of the prediction task being trivial, like finding out whether a specific item could be interesting for the user, the predefined outcomes could be *interesting / not interesting* or interest values on a scale of, for instance, 1 *to* 5. In the case of the prediction task being non-trivial but a huge information body being available, the predefined outcomes could be different learner types. In this concrete case, it would be necessary to know what types of learners can be expected, which could be derived from experience with the system and its users in combination with the definitions of well-known learner types.

However, if this kind of information is not available, the system's task becomes more difficult – it must be able to automatically determine patterns in usage data to retrieve definitions of the differences between users' behaviours in order to derive the classes that were predefined in the other scenario described before.

Thus, the decision whether to use a supervised or an unsupervised learning approach (see Section 1.1.3) depends on the amount of information available preliminarily, i.e., if what is looked for in the data, is known, *classification* can be applied (see Chapter 4), whereas *clustering* is applied if what can be expected in the data is not obvious beforehand. The latter can be the case if the task is, for instance, to find out what problem-solving styles exist in a specific domain like mathematics, physics or computer science (see Chapter 6).

The prediction process itself is simpler in the case of a predefined output because no additional human input is required for the interpretation of the results, however, in the case of no predefined results, novel knowledge about the users who produced the data can be gained, which is possible only in a limited way with classification.

Second, in some cases, temporal information in usage data is highly relevant, whereas in other cases it might be neglectable. Whether temporal and other relational information in a usage history are essential or not, depends on the environment the data has been produced in, and on the adaptation strategy and thus also prediction aims.

For instance, if prediction should be done based on independent activities (as shown in Chapter 4), it is not necessary to have the respective activity linked to the preceding or succeeding one. If the prediction task is, for example, to find out if a specific item is interesting for a specific user, this can well be done based on individual activities.

If, however, the prediction task is to find out what kind of learning style a specific user exhibits, it might be necessary to know in what order the user performed different steps. In case sequential information is needed to achieve reliable predictions, an additional modeling step must be performed before data can be actually sent through the analysis / prediction phases (see Chapter 5).

The modeling step aggregates usage data by bundling data instances based on their relations to each other. The outcome is a new data instance which can then be passed over to the analysis unit responsible for prediction. The modeling approach again depends on the nature of the data and the prediction aims. If usage data comprises information of different tools and users, it might be interesting to find out how users are related to each other, or how tools are used in combination. If usage data comprises data of only one tool that can be used in different ways, it might be interesting to find out in what order the different activities happened.

E-learning environments like Sakai [Sakai, 2010] usually offer a lot of tools for different purposes like communication, cooperation or learning support. Tutoring system like Andes [VanLehn et al., 2005] usually focus on learning content. The resulting different data sets are described in the following section.

3.2 Data Sets

In the following chapters, two different data sets are used for the demonstration of different approaches to the analysis of usage data in order to gain information about users and their behaviour that could later be used as a basis for adaptation. This section introduces and explains the two data sets.

3.2.1 Data Set I: E-Learning Platform Sakai

Data Set I contains usage data monitored by the e-learning platform Sakai [Sakai, 2010]. The data set involves usage data of various tools: announcements, forums, content (uploaded documents), calendar, assignments and wiki. Usage data comprises so-called “active” and “passive activities”, with “passive” being all read-activities where a user has viewed any content

Table 3.1: This table lists the attributes and possible values in Data Set I.

Attribute	Tool	Activity	Resource	User	Site
Possible Values	Announcements Calendar Assignments Content Forums Wiki	View New (Item) New (Topic) Revise Submit (Assignment) Grade (Assignment)	All Resource IDs e.g., the ID of a document	All user IDs	All site IDs

or communication item like an entry in a forum. "Active" comprises all other events like creating or updating an item.

Usage data was monitored by an "activity monitor" extension that was integrated into the Sakai environment for the installation on which the experiments were conducted. A data preprocessor later filtered out activities that were regarded not relevant for the analysis and converted the remaining data to a format better processable by the analysis unit used later.

Table 3.1 provides a list of the elements in a data instance. The overall data set contains 4967 instances produced by 31 users.

Data Set I is used later in Chapter 4 and Sections 5.1, 5.2 and 5.3.

3.2.2 Data Set II: Tutoring System Andes

Data Set II contains usage data monitored by the ITS Andes [VanLehn et al., 2005] and made available via the PSLC DataShop [Koedinger et al., 2008], [Koedinger et al., 2010]. This data set was selected for two reasons – first, because it is available via a public data repository, which makes the results easily comparable and replicable for subsequent work, and second, because it forms a contrast to the Sakai data set.

Data Set II, in contrast to Data Set I, does not contain data logged by different tools, but comprises usage data of a learning-only environment. Sakai integrates different tools for communication, cooperation and learning, whereas the ITS concentrates on learning support, which makes the information collected more fine-grained but also more narrow regarding the overall picture.

Furthermore, on the ITS, users interact with the system only without influencing each other or even noticing each other's presence on the platform. Thus, the ITS data set can be very well used to determine detailed information about a single learner's behaviour while the Sakai data set can be used to determine more superficial information about a user's interests and

Table 3.2: This table lists the attributes and possible values in Data Set II.

Attribute	Student	Student Response Type	Student Response Subtype	Topic	Problem (Knowledge Component)	Unique Step
Possible Values	All Student IDs	Attempt (Correct or Incorrect) Hint Request Cancel	Answer (Correct or Incorrect) Next Step Help Explain Further What's Wrong	E.g., Electric Fields Electric Potential Magnetic Field Electro-magnetic Induction Inductance	All Problem IDs, e.g. Charge1A Coul1A Gauss1	All Unique Step IDs

preferences which can be very useful in order to get a good overall picture of the user. Table 3.2 provides a list of the elements in a data instance. A student basically has the opportunity to answer a question or to request help; help can be requested in different ways, as shown in the table.

In total, the raw data comprises student activities of three semesters (same course) and contains $\sim$ 280000 activity instances produced by 73 users for 2007, $\sim$ 265000 activity instances and 97 users for 2008, and $\sim$ 115000 activity instances and 45 users for 2009. As described later in this thesis, the data listed in table 3.2 was not used in the raw form but was preprocessed before being analyzed and interpreted.

The preprocessor extracted, for every user, the sequences of actions within one problem, i.e., all activities that were taken by a student to solve a specific task. In a next step, data was re-organized (as described in detail in Chapter 5), resulting in data instances modeling a student's solving sequences for a specific topic, i.e. a data instance then aggregates the problem-solving behaviour of different Knowledge Components (KCs) within a task.

Table 3.3 shows the attributes in the resulting data set. The table is to be read as follows: a (transition) probability value stands for the probability of a student behaving in a specific way, e.g., for requesting a hint directly after an incorrect answer. A "prior probability" stands for the probability of a student taking a specific action as a first activity of solving a task, for instance, a prior probability of 1 for correct attempt in a specific topic means, that a student, in 100% of the KCs in the topic, provided a correct answer as a first activity. A value of 0.5 for the transition probability from an incorrect answer to a hint request would mean that the student in 50% of the cases, requested a hint as a next step when having submitted an incorrect answer.

Table 3.3: This table lists and describes the attributes of instances in the preprocessed data sets.

Feature name	***Feature description***
PRIOR_PROB_C	prior probability of a correct attempt
PRIOR_PROB_I	prior probability of an incorrect attempt
PRIOR_PROB_H*	prior probability of a help request as the first action of the first task in a problem
TRANS_PROB_C_C	transition from a correct attempt to a correct attempt
TRANS_PROB_C_I	transition from a correct attempt to an incorrect attempt
TRANS_PROB_C_H[*]	transition from a correct attempt to a help request
TRANS_PROB_C_E	transition from a correct attempt to end of task (i.e., step finish)
TRANS_PROB_I_C	transition from an incorrect attempt a correct attempt
TRANS_PROB_I_I	transition from an incorrect attempt to an incorrect attempt
TRANS_PROB_I_H[*]	transition from an incorrect attempt to a help request
TRANS_PROB_I_E	transition from an incorrect attempt to end of task (i.e. step finish)
TRANS_PROB_H[*]_C	transition from a help request to a correct attempt
TRANS_PROB_H[*]_I	transition from a help request to an incorrect attempt
TRANS_PROB_H[*]_H[*]	transition from a help request to a help request
TRANS_PROB_H[*]_E	transition from a help request to end of task (i.e. step finish)
TRANS_PROB_E_C	transition from end of task to a correct attempt
TRANS_PROB_E_I	transition from end of task to an incorrect attempt
TRANS_PROB_E_H[*]	transition from end of task to a help request
PERC_HELP_STEP	percentage of help requests in a user's activities
PERC_INCORRECT	percentage of incorrect attempts in a user's activities

As the system can provide four different kinds of hints, different features containing H are listed in the table. $H[*]$ means that this feature might exist several times (with $H1$, $H2$, $H3$ and $H4$). As described later, the data sets exist twice - one of them considering the *different* system hints (i.e., the "extended help processing configuration"), the other one aggregating all kinds of hints as H – in these data sets, no distinction between the hints is made (i.e., the "aggregated help processing configuration").

Data Set II is used later in Sections 5.1 and 5.3 and Chapter 6.

Chapter 4

Prediction of Interest Based on Individual User Activities

As introduced previously, this chapter provides an example of supervised machine learning as the basis for later adaptation. It discusses a machine learning approach but also presents a simpler statistical one, and compares both. The greater part of an earlier version of this chapter was published by [Köck, 2009], including tables and figures.

In general, if a system should offer recommendations related to communication and learning content, it is necessary to infer a user's level of interest in specific topics. Therefore, a user's history on the system is examined and previous interests are used to predict future ones. Here, Sakai data (see Data Set I in Chapter 3.2) is used. In a first step, a user's passive ("consumption") activities are collected. This kind of activities is further also referred to as "read activities". Reading an element (e.g., an entry in a forum or a document) denotes a user's interest.

In a second step, the other activities become interesting. Given, a user was interested in one specific element, similar ones can be found and it can be assumed that those are also interesting for this user. Given this kind of "knowledge", the system can try to infer user interest for as many activity events as possible which can then become the basis for adaptation. This general idea can be put into practice by several different implementation approaches which provide different quality and granularity of results. All of them have in common that the primary objective is to classify data continuously produced by users' activities on a platform.

As explained in Chapter 3, data can be analyzed in different ways – different levels of classification are distinguishable: classification of individual user activities, and classification of user activity sequences. The first kind, as opposed to the second one, treats activities as if they were independent. The second kind is promising for fine-grained modeling of behaviour, but it requires a higher amount of reference information (see for more details the following chapters).

This chapter concentrates on the first kind, which does not consider the time context of, and relations between, activities but uses activity items as independent of each other. Nevertheless, in most cases (depending on the learning technique) the system must still be provided a certain amount of reference data before classification of fresh data can be performed. This means that, in this case, no long period of training is necessary as long as some representative data sets are available. Therefore, the only prerequisite for this kind of classification is a certain period of data collection (depending on users' level of activity). Classification of independent activity items can be useful at the level of both individual users and groups.

A classification approach has to be capable of computing realistic values for every user's interest in an event. Basically, we can distinguish between (trivial) statistical and more "intelligent" approaches. The main characteristic of intelligence in this context is that the respective approaches are capable of learning. The trivial statistical approach will work for some scenarios ([Jung et al., 2005], for example, introduce a statistical model for user preferences which performs well) but it can turn out to be too inflexible in others.

The following sections describe two different approaches operating on the same kind of data.

4.1 A Statistical Approach

This section describes a common statistical approach to the classification of activities based on data provided by the "Recent Activities" tool [1] which is an add-on to the e-learning platform Sakai [Sakai, 2010]. The tool logs the user activities and stores them in the format described in the previous chapter as Data Set I (see Section 3.2.1).

4.1.1 The Environment

The "Recent Activities" tool, as the name suggests, summarizes recent activities within a learning platform at different scopes: over all courses a learner attends, and for an individual course. It includes information about users' activities in different kinds of facilities:

- announcements,
- chat,
- forums,
- personal messages,
- resources (uploaded documents),
- calendar, and
- assignments.

[1] The „Recent Activities“ tool was developed in the context of the Adaptive Learning Spaces (ALS) project. For further information, please refer to http://www.als-project.org

The users can additionally select the time period they want to get information for and choose between “since last login”, “for the past day”, “for the past week”, “for the past month” and “all dates”. The tool then offers two different views – a user can either have the activities listed grouped by sites (i.e., a course) or by activity type.

The *site view* lists all sites the user is registered at and that include recent activities. Clicking a site will open another list showing all tools with recent activities. Clicking a tool opens another list showing the actual activity items (like e.g., chat messages). In order to prevent too much information, the number of items that are shown is limited. If not all items are shown, the user gets a short information text about the actual number of items. In every tool list, there is a link taking the user to the location where the activity occurred.

The *activity type view* lists all tools where activity occurred recently. Clicking a tool opens a list containing all sites where activity occurred in the respective tool. Clicking the site then opens the list of items. When using the “Recent Activities” tool from within a course site, this is the only view provided.

In addition to the general overview, the tool offers a personalized version on the basis of interest models created from implicit evidence derived through naturally occurring learner interactions. Personalization for both views can be enabled by a simple click. After that, the page will show light bulbs next to the items the system believes are interesting for the user. Furthermore, an additional view called “Personalized Overview” will appear. It contains a list of links to the site / tool combination the system believes are interesting for the user. These features are enabled as soon as the system has enough information to do recommendations.

As a supplement to the basic functionality of the tool, the information is also made available via two different Really Simple Syndication (RSS) feeds: a general and a personalized one. Again, the personalized feed contains only items about tools / site-tool-combinations where the system believes they are of specific interest for the current user. The algorithm is the same that is responsible for creating the personalized overview and the light bulb markers in the site and activity view. Again, if the system does not have enough information to perform personalization yet, this feed will contain the same information as the main feed. As soon as there is enough information the feed will change. The user is informed about the current state by the description of the RSS channel.

4.1.2 The Algorithm

In its first version, the tool utilizes a relatively trivial statistical algorithm to compute a user’s potential interests in different elements [Köck, 2009]. This algorithm uses past interest to infer future one based on the statistics produced. Past interest is indicated by users’ read activities, i.e., every time a student reads an element like an entry in a forum or a chat message, or downloads a document, etc. the respective statistics is updated.

The general process works as follows: first, the distribution of a user's read activities among tools in a site is computed, using standard statistical metrics like mean, standard deviation, and variance to determine probability / density distributions. Given only the mean, the risk of statistical outliers distorting the picture would be too high, therefore, the standard deviation is considered. Given the standard deviation, a tool's individual deviation σ_{T_x} from this value can be used in order to identify significant results. Significance in this context includes both, significantly high and significantly low values. The following simple example illustrates the algorithm's behaviour.

Consider an example using 5 hypothetical tools and 25 read activities produced by 1 user within a specific time frame, distributed among the tools as $c_1 = 10$, $c_2 = 2$, $c_3 = 4$, $c_4 = 3$, $c_5 = 6$. The system then wants to compute this user's interest value for every tool. This would result in the following:

$$\bar{x} = 5$$

$$\tilde{x} = 4$$

$$\sigma^2 = \frac{1}{5} * \sum_{i=1}^{5} (x_i - \bar{x})^2 = 8$$

$$\sigma \approx 2,83$$

In a next step, a tool-specific metric can be determined as

$$\sigma_{T_x} = |c_x - \bar{x}| - \sigma$$

which will mark all resulting $\sigma_{T_x} > 0$ as significant (in both directions). In the above example σ_{T_1} and σ_{T_2} will be positive values, marking T_1 as significantly high (as $c_1 > \bar{x}$) and T_2 as significantly low (as $c_2 < \bar{x}$) regarding interest.

This algorithm, however, is not as flexible as one based on techniques more flexible than pure statistics. For instance, a statistical formula, even if it contains variable elements cannot adapt to different scenarios as well as a self-learning algorithm does. Users may differ in their interests across sites, tools or resources. Additionally, courses themselves can also differ, for instance, regarding the role of communication within a course.

Furthermore, we might want to weight activities based on different factors like the time when they occurred. Therefore, the statistical formula would have to differ for different combinations of users, tools, courses and resources, which would again entail additional effort and might still not lead to the intended flexibility. Thus, the next version of the "Recent Activities" tool faced a major change in the underlying algorithms, replacing pure statistics by classification techniques to predict user interest.

4.2 A Machine Learning Approach

4.2.1 Motivation

In order to overcome issues and problems raised by naïve statistical approaches, classification techniques of the field of machine learning can be used. These techniques do not make as many assumptions as statistical approaches do, but learn from the user. Although the classifiers used for the experiments described here differ drastically in their way of model building, they have in common that their models consider all features provided as input.

Here, 8 attributes are available, 6 of which (the anonymized user id, event id, tool id, site id, related resource and the interest class) are taken into account by the classifiers. The remaining two, index and timestamp, were removed by a filter in pre-processing because this kind of classification does, as mentioned in the introductory part of this chapter, not yet consider temporal relations.

This means, all solutions we get by the classifiers adapt to all feature values of new input events. Thus, not only the site where the event occurred is considered, but also, e.g., its creator and the tool where it originated. To further extend flexibility and personalization, the classifier then computes an event's interest value for every user individually. This implies that the approach works separately for every user.

As the classifier is continuously fed with new information, it is able to learn and adapt its behaviour during the process. As each of the classifiers builds a model (e.g. a decision tree, a BN, or a rule base) which can be queried, it is also possible to extract information from it, which will offer additional knowledge about users, behaviours, and the whole construct of content, courses and tools. In addition, dependencies and correlations between attributes could be found which might become important for further event design.

4.2.2 Experiments

This section describes experiments designed to test the classification approach on real user activity data and compare the performance of different techniques for different aims.

Description

The experiments aim at producing a group-based interest model – the system is fed with all users' activities and tries to classify new events as interesting or non-interesting for every user individually, but uses this knowledge to build an overall group model. This model can be referred to in later stages of the adaptation cycle to offer group-based adaptations.

Working with group models can be beneficial in several ways. For example, to avoid the "cold start" problem [Höök, 1997], a group model can become the default for a new course participant. The system then does not have to create new models from scratch any more but can build upon one based on the interest and activities of a group working on the same content and tasks.

The experiments also outline the extension to the behaviour of the “Recent Activities” tool. As already described, CI techniques can improve the performance, flexibility, and accuracy of adaptive components because they learn from the user. The experiments described in this section use these techniques to replace the statistical model. Real usage data is provided by a monitoring extension to Sakai. The instances are independent and handled as random set elements here.

The overall data set contains 4967 instances with 6 features, as described before. String attributes were converted to nominal ones, meaning that before data went into the classifiers, a filter collected all possible values for a feature (for instance, all tools where activity was monitored). The resulting set of values then allows for better computation of probabilities. The experiments were run on data of one specific course about the Unified Modeling Language (UML), with 31 participants in total. Data was collected over a period of several months and went through some pre-processing during which irrelevant or pseudo-data (e.g. produced by test users) was removed. During these steps, anonymization was also performed by encoding user IDs with a one-way hashing algorithm.

Process and Technologies

The experiments were carried out iteratively with training, testing and evaluation steps repeated for different classification algorithms. Validation was performed in two different ways – by 10-fold cross-validation, and by specifically split training- and test sets. A comparison of the algorithms' results concluded the experiments and became the basis for classifier rating and final selection. The Weka machine learning software [Hall et al., 2009] was used to run the experiments. For a more detailed description of the algorithms please refer to Weka documentation and tutorials. The following paragraphs describe the configuration of the classification algorithms which were used.

Naïve Bayes: The NB approach builds a simple network with one parent node (the class label, in our case the interest value). There are no important additional configuration alternatives.

Bayesian Network: The network applied for the experiments used the SimpleEstimator approach for finding the conditional probability tables of the net. The TAN algorithm (determining the maximum weight spanning tree and returning a BN augmented with a tree) was applied for searching network structures.

Sequential Minimal Optimization(SMO): SMO is used to train an SVM. Standard settings with relatively low complexity (the higher, the fewer wrong classifications are accepted) and a polynomial kernel $K(x, y) =< x, y >^p$ with exponent $p = 2$ were used.

Multilayer Perceptron (Backpropagation Neural Network, later in this section referred to as *NN*): A network with $a = attributes + classes$ hidden layers of sigmoid nodes, a learning rate of 0.7, momentum of 0.2 and 500 learning cycles were used. Please note that run-time filters like nominal to binary slow down the process significantly.

IBk (Nearest Neighbour): $k = 10$ and the LinearNearestNeighbourSearch (brute force) algorithm for nearest neighbour search and cross-validation were used.

JRip (Rule-based): This algorithm implements a propositional rule learner and provides a set of rules which are then used as a basis for classification decisions. The experiments described here used 10 folds (for pruning and growing rules) and 6 optimization runs.

J48 (Tree-based): This algorithm, building a decision tree, used a confidence factor (small values mean more pruning) of 0.25 and reduced error pruning here.

RandomTree: This algorithm, building a decision tree, used a KValue (i.e. the number of randomly chosen attributes) of 1 and an unlimited tree depth.

Results

The results of the described base experiments are listed in Table 4.1 and Figure 4.1. The base experiments used 10-fold cross-validation to get a first impression of the classifiers' performance. Subsequently, more specific experiments were conducted in order to find out how their performance changes over time.

The experiments were conducted on a $2,98$ GHz dual-core machine with 4 GB RAM, running a 64-bit Windows XP. As a first experimental step, the performance of different classification techniques is compared to the performance of a statistical approach as described in Section 4.1. The percentage of correctly classified instances ranges from 96.63 (NB) to 98.41 (SMO) for the machine learning techniques. The statistical model obtains a result of 68.94%. In the following, a more detailed comparison of the classifiers listed in Section 4.2.2 is provided.

The percentage of correctly classified instances from now on refers to "positive" instances (i.e. the instances with an interest value of 1) only. The overall results, containing "negative" instances also are less expressive, as the number of these instances is higher and their classification much easier (for the CI techniques only). This leads to a very similar overall performance of the classifiers and subsequently to a misleading picture and potentially wrong conclusions.

The results show that the classification task itself can be handled relatively well by different classification techniques. As there is only little discrepancy regarding the number of correctly classified instances, process duration becomes an even more important criterion. After running experiments with cross-validation, which provided a first general impression on the performance of machine learning techniques on the data, a similar experimental setup

Table 4.1: This table lists classification results of various algorithms on the overall data set (10-fold cross-validation). The neural network classifier is listed twice, once with filters (f). The table further displays the percentage of correctly classified positive instances, the True Positive rate for class 1, the Root Mean Squared Error, the time taken to build the model T_m, and the time taken for the overall process T_o.

Class.	***Corr.***	***TP***	***RMSE***	T_m ***(s)***	T_o ***(h,m,s)***
NB	47.6%	76.1%	0.1550	$< 0.01s$	$< 1s$
BN	70.0%	72.3%	0.1112	$< 0.01s$	$< 1s$
SMO	70.4%	84.5%	0.1261	$105.39s$	$16m54s$
NN(f)	70.3%	12.3%	0.1668	$17.22s$	$21h16m8s$
NN	56.0%	60.6%	0.1366	$17.16s$	$2m51s$
IBk	70.4%	84.5%	0.1092	$< 0.01s$	$5s$
JRip	68.0%	85.2%	0.1143	$0.63s$	$7s$
J48	70.7%	74.8%	0.1137	$< 0.01s$	$< 1s$
RT	70.4%	84.5%	0.1092	$0.13s$	$< 1s$

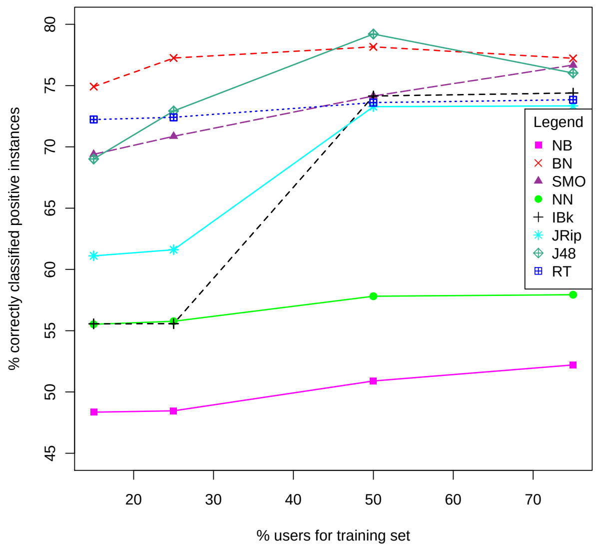


Figure 4.1: These plots show how the performance of classifiers increases with an increasing amount of training data.

was modeled in order to measure how fast the classifiers learn from given input data. The experiments were run on split data (training set and test set).

As a group model should be built, the percentage split for the data set is based on users, not resources. This means that the training set does not contain a certain percentage of the data but all data of a certain percentage of users. The experiments were run several times with the events for 15%, 25%, 50%, and 75% of the users as training and the remainder as test set. Depicted in the plots (Figure 4.1), the results show three different trends. BN, RandomTree and J48 (the last two both tree-based approaches) show good classification performance right from the beginning and relatively steady behaviour. SMO could also be added to this „cluster“ of algorithms, regarding its effectiveness.

Next, a second cluster can be seen containing IBk and JRip. These algorithms show good results but not right from the beginning. However, their plateau is at about the same place as the first cluster.

The third trend can be seen in MultilayerPerceptron (NN in the figure) and NB which are steady in their performance but don't provide promising results. This means that for subsequent work the classifiers of the third cluster were not considered any more.

The most promising candidates are those in the first cluster, where the favourites are BN and the tree-based classifiers. SMO, compared to the other classifiers, is relatively slow, with the time needed to build a model increasing at least linearly as the training set grows. In general, a linear algorithm is reasonable for run-time employment. Here, comparing SMO to the faster classifiers, the discrepancy in computation complexity (< 0.01 seconds as opposed to 1.15 seconds for building the model for the smallest training set) is significant enough to be an exclusion criterion.

Another important criterion for the selection of a classifier is here the possibility of information extraction, given a model. Descriptive classifiers like BNs, rule- or tree-based approaches enable very simple extraction of semantic information, whereas function-based ones like neural networks or SVMs behave like blackboxes. Generally we can conclude that learning classifiers perform well on the present kind of input data.

As the experiments described above have shown, classification approaches can perform well on user activity data as produced on a learning platform like Sakai. The results could potentially be improved by combining (complementary) classifiers using ensemble methods like bagging, boosting or stacking [Opitz and Maclin, 1999].

- *Bagging* [Breiman, 1996] is a method where one classifier is trained with several subsets of the data.

- *Boosting* [Freund and Schapire, 1996] is a method where one classifier is given differently weighted data.

- *Stacking* [Wolpert, 1992] is a method implementing a meta-learning process based on the predictions of several different classifiers.

The solution suggested here is not restricted to Sakai or even the "Recent Activities" tool, as data of any learning environment can easily be converted to a similar format.

4.3 Summary

This chapter described a classification-based approach for the analysis of individual user activities, aiming at providing a basis for adaptive learning support. Specifically, an analysis as described here is applicable if learning support should be content-based rather than process-based.

However, although the classification approach clearly outperformed the simpler statistical approach, and although it improved performance, a supervised learning process like this is restricted in several ways.

First, supervised learning in general requires intervention already in early phases of the overall process. More precisely, it is necessary to have a sufficient amount of training data correctly classified. Second, supervised learning is not capable of autonomously detecting patterns, which is, however, a requirement the approach presented here should be able to meet. Therefore, as described in the following chapters, a more dynamic clustering approach, i.e., unsupervised learning, is better suitable here.

Additionally, here individual user activities were analyzed and treated as if they were independent from each other. This is sufficient for many scenarios but might bear the danger of losing an important dimension of information in others. The following chapters discuss two aspects identified as limiting for the approach presented in this chapter. Different ways of modeling sequences in user activity data will be introduced, before one is selected and set up for usage in unsupervised learning.

Chapter 5

Data Analysis and Activity Sequence Modeling

This chapter describes ways of representing different kinds of dependencies in activity data. Unlike the classification approach explained in Chapter 4, activities are here not treated as independent from each other. The chapter discusses, on a theoretical basis, two different categories of approaches to retrieving sequential information in user activity data.

The first category depends on activity data produced by a *group of users* who are aware of each other's activities and are given opportunities to interact, e.g. via tools they share, like a forum. An approach in this category primarily tries to model the dependencies between the users given a particular cooperation situation. In a similar way, such an approach can also be applied in order to model dependencies between tools or contexts, however, users are probably the most interesting subjects of analysis because information about them is potentially more useful for later adaptations.

The second category of approaches does not depend on cooperation data being available, instead, it aims at analyzing the parts of the behaviour of an *individual user* that are relevant for cooperation behaviour.

The approach that seems most promising regarding its suitability for activity data analysis and extraction of information that can become the basis for adaptation, is then selected for implementation. The selection further depends on the nature of the available data.

The following sections describe examples for both categories (see Sections 5.1 and 5.2 for the first, and Section 5.3 for the second one). Section 5.4 explains which approach was finally selected for implementation and why it was the most promising one.

5.1 A Graph-Based Similarity Model

This section describes a theoretical approach to modeling dependencies between activities, users, tools, etc. with directed, potentially cyclic graphs.

5.1.1 Description

This concept is based on modeling different kinds of temporal dependencies that can be extracted from subsequent activities within e-learning platforms as graphs. It will be explained using example data sequences of the two data sets introduced in Section 3.2.

Example *a* is shown in Table 5.1 and Figure 5.1. Table 5.1 shows a sequence of different activities performed by different users within different contexts. This example sequence was taken of Data Set I described in Section 3.2.1. The first modeling step is to describe the sequence of activities as a graph (see Figure 5.1(a)). The graph enumerates the events with their indices and additionally shows the edges' weights which are based on occurrence frequency. As every unique event can only occur once, the weights for the transitions are all equally 1 here.

In a next step, additional graphs are constructed, based on the initial one that are carrying different kinds of semantic information each. In example *a*, there are 5 different aspects that can become the basis for a new graph model: user, tool, type of event, resource, and site. The new graphs are constructed as follows: the base graph is analyzed according to one specific aspect, e.g. the related user (see Figure 5.1(e)).

Instead of the activity index, the nodes in the new graph now represent the different users that were involved in the activity sequence. The edges now depict the (number of) transitions between the users. As opposed to the base graph, the user-based one might also contain cycles. The process of graph construction according to a specific aspect is repeated for all aspects that are potentially interesting for later stages of the adaptation cycle. Figure 5.1 shows the base graph and the aspect-based resulting graphs as produced by the conversion process.The resulting graphs become then the basis for later pattern recognition.

Example *b* is shown in Table 5.2 and Figure 5.2. This example is different to example *a* in several ways. A data set as used here is produced by an ITS which aims at supporting the individual learner rather than the group process. Thus, the students only see their own activities and are not influenced in any way by the activities of others. For that reason, the resulting graphs must be created for every learner individually, a graph depicting the relations between users is not informative because the relations that might be read out of it would be coincidental. Furthermore, the ITS in this case organizes learning material into different topics which are again split into units called "Knowledge Components". These units then each represent a task for the students who can provide different kinds of answers (see the explanation of Data Set II in Section 3.2.2). This means that using the modeling approach described here, would result in many different graphs, each of them representing a student's path for a specific KC. Figure 5.2 shows the graphs that would be created for the analysis of the activity data in Table 5.2.

Example *b* shows, that the approach is not equally well applicable for different kinds of data. The data set depicted in Table 5.2 and Figure 5.2 is apparently not suitable for identifying relations between users, because the users do not have any possibility to interact with each

Table 5.1: This table shows a small test data set containing different events that occurred within an e-learning platform in the order as listed here. A data instance is described by an index, the tool in which the event occurred, the type of event (i.e., activity), the related/resulting resource, the user responsible for the event, and the site in which the event occurred. Indices 4 and 5 do not provide a site because they occurred in an overview-page that is not handled as a course page.

Index	***Tool***	***Activity***	***Resource***	***User***	***Site***
1	calendar	new	resource4	user3	site3
2	announcement	new	resource5	user3	site3
3	content	new	resource6	user3	site4
4	roster	view	resource7	user4	null
5	forum	new topic	resource8	user4	null
6	announcement	new	resource9	user4	site5
7	announcement	new	resource10	user4	site6
8	calendar	new	resource11	user5	site6
9	announcement	new	resource12	user5	site6
10	content	new	resource13	user5	site7

other. As the subfigures show, we can, however, find out in which order students use different types of answers. For this task it would be better to find a way to combine the resulting models (i.e., the ones that represent the same user and problem but different KCs). This would result in fewer models carrying more information, an idea which is further discussed and applied later in Section 5.3.

The following Sections 5.1.2 and 5.1.3 refer to the general idea rather than to its application with a specific data set.

5.1.2 Process and Objectives

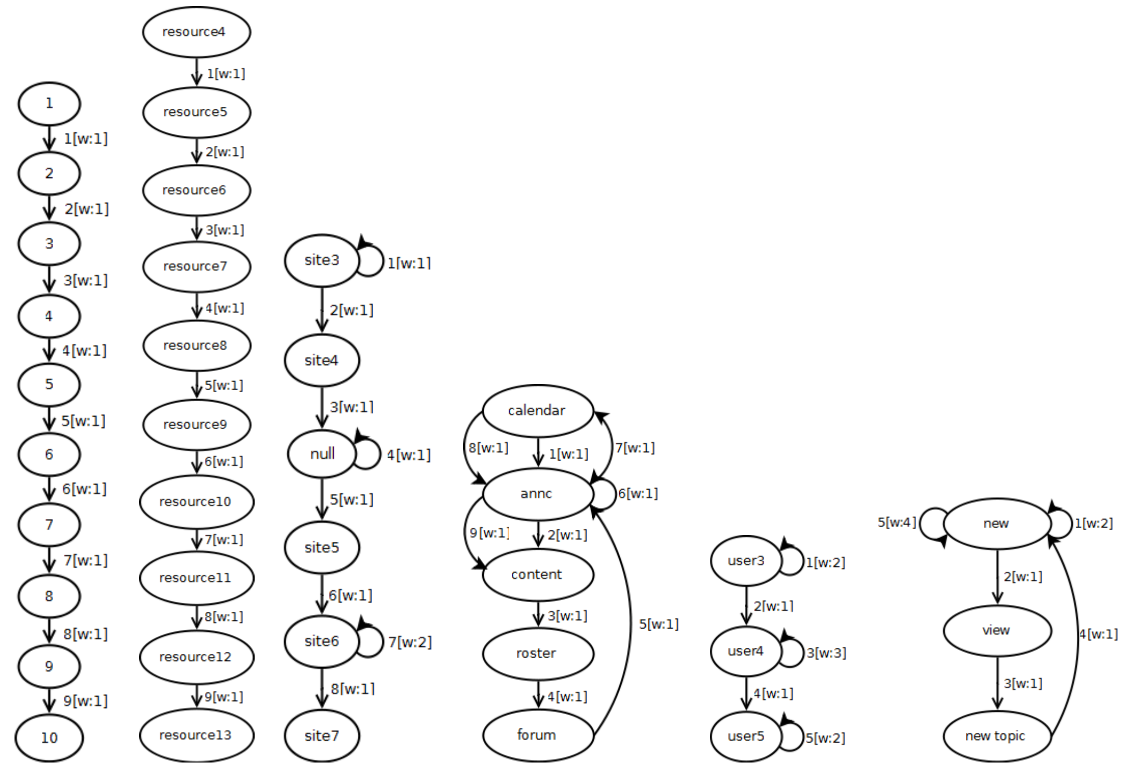
The overall process can be divided into the two phases *model construction* and *model application and adaptation*. The first phase is needed to generate adequate reference models and includes a certain period of activity monitoring, before expressive graph models can be built. The second phase can again be split into subphases, resulting in the following overall procedure:

1. live model construction,
2. model matching, and
3. model feedback.

Live model construction includes the monitoring of current user activities on the platform, and the live construction of new activity graphs, following the same process as described for

Table 5.2: This table shows a small test data set containing different events that occurred within an ITS in the order as listed here. A data instance is described by an index, the student who caused the event, the type of the student's response (e.g., Hint Request), the more detailed type of the student's response (e.g., the request to get information about what was wrong with the student's answer), the topic, the so-called "Knowledge Component" (i.e., the concrete problem), and the unique step (i.e., a unique activity within a specific problem).

Index	***Student***	***Student Response Type***	***Student Response Subtype***	***Topic***	***Knowledge Component***	***Unique Step***
1	Student1	Answer (Correct)	Answer (Correct)	Topic1	KC1	S1
2	Student1	Answer (Incorrect)	Answer (Incorrect)	Topic1	KC2	S1
3	Student1	Hint Request	What's Wrong	Topic1	KC2	S2
4	Student1	Answer (Incorrect)	Answer (Incorrect)	Topic1	KC2	S3
5	Student1	Hint Request	Next Step Help	Topic1	KC2	S4
6	Student1	Answer (Correct)	Answer (Correct)	Topic1	KC2	S5
7	Student1	Hint Request	Explain Further	Topic2	KC1	S1
8	Student1	Answer (Correct)	Answer (Correct)	Topic2	KC1	S2
9	Student2	Answer (Incorrect)	Answer (Incorrect)	Topic1	KC1	S1
10	Student2	Answer (Correct)	Answer (Correct)	Topic1	KC1	S2



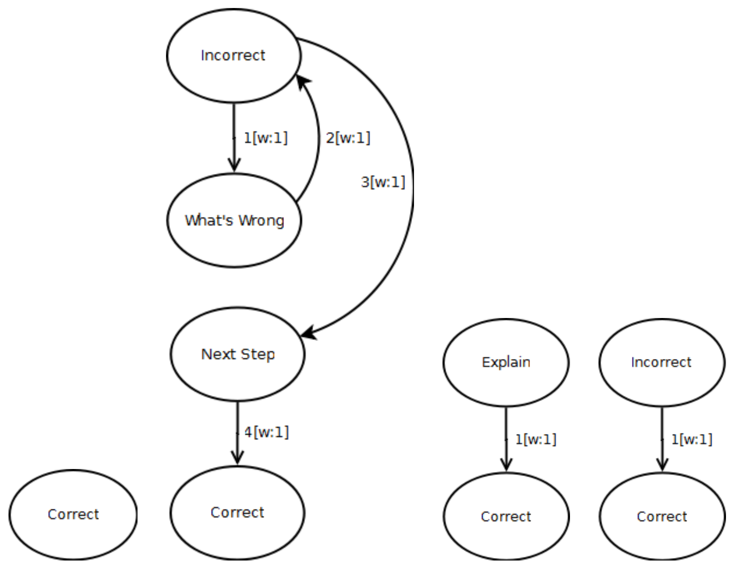
(a) Events (b) Res. (c) Sites (d) Tools (e) Users (f) Activities

Figure 5.1: Figures 5.1(a) to 5.1(f) show the different graph-based representations of test activity data listed in Table 5.1.

model construction. Live construction considers events within a specific time frame, including aging so that the latest activities are treated as most important, and out-dated activities are not depicted in the graph any more. The next subphase, *model matching*, then receives as input the model graphs and their respective counterparts in the live graph pool, and compares them. The results of graph-matching describe the degree of similarities between the model- and the live graphs. The aim of the matching process is to find similar constellations in the graphs in order to predict future activities. The information gained in the live process is again fed back into the base model, therefore, the base model is constantly updated. Thus, the longer the process runs, the more precise the model becomes.

5.1.3 Evaluation and Potential Shortcomings

This section describes potential problems coming along with the graph-based similarity model.



(a) S1T1KC1 (b) S1T1KC2 (c) S1T2KC1 (d) S2T1KC1

Figure 5.2: Figures 5.2(a) to 5.2(d) show the graph-based representations of test activity data listed in Table 5.2. As the activities of different students and of one student in different topic areas are not related, the activities must me split up into several graphs that contain related items. The subfigure designators are, for example, to be read as follows: S1T1KC1 means that the graph contains data of Student1 produced in KC1 of Topic1.

Computation Complexity

The example (a) introduced before is a very simple one. In a live system, the graphs will become disproportionately more complex as the number of users increases and the number of activities grows. Graph matching is one of the most complex problems in object recognition in computer vision [Bengoetxea, 2002], [Bienenstock and von der Malsburg, 1987]. Here we specifically face the problems of *graph and subgraph isomorphism*.

The problem of *graph isomorphism* can in general be described as a bijection between the vertex sets of two graphs G and H $f : V(G) \rightarrow V(H)$ such that two vertices u and v of G are adjacent in G iff $f(u)$ and $f(v)$ are adjacent in H, i.e. a one-to-one correspondence must be found between each node of the first graph and each node of the second graph [Conte et al., 2004].

Graph isomorphism is one of very few problems in NP that are neither known to be solvable in polynomial time, nor NP-complete [Garey and Johnson, 1979]. Yet, if it is usually not necessary to match the complete graphs but to match live graphs with subgraphs in the respective models; a weaker form of matching, *subgraph isomorphism*, is sufficient.

Subgraph isomorphism is a problem that has been proven to be NP-Complete [Garey and Johnson, 1979], although specific types of graphs can have lower complexity [Bengoetxea, 2002]. Polynomial isomorphism algorithms have been developed for specific kinds of graphs, like, for instance, trees [Aho et al., 1974], or planar graphs [Hopcroft and Wong, 1974], but none for the general case, therefore, exact graph matching is of exponential time complexity in the worst case [Conte et al., 2004].

For the graph-based similarity model presented here, it is, however, not sufficient to know whether a graph or subgraph matches another one, because an exact match is extremely rare in practice. Thus, it would be more interesting to compute the "distance" between two graphs or subgraphs in order to rate their similarity – a problem known as *inexact graph matching*.

Algorithms can be divided into optimal ones and approximate or suboptimal ones [Conte et al., 2004]. Optimal matching algorithms reliably find the solution that is the global minimum of the matching cost, i.e., if a solution exists, it will be found, whereas approximate matching algorithms only ensure to find a local minimum of the matching cost. As explained by [Conte et al., 2004], many inexact graph matching algorithms define their matching cost based on an explicit error model, assigning a specific cost to every kind of error that may occur.

In addition to these *error-tolerant* algorithms, there are algorithms defining their matching cost by graph edit operations (which are necessary to transform one of the graphs into the other). Many of these algorithms, however, face the problem of running into combinatorial explosion when the size of the graphs becomes large (see, for instance, [Ullman, 1976], [Hlaoui and Wang, 2002], [Tsai and Fu, 1979], or [Sanfeliu and Fu, 1983]), which makes them inappropriate for inexact graph matching.

A new algorithm discussed by [Hlaoui and Wang, 2002] aims at computing the distance between two graphs (i.e., the smallest matching error) and is based on iterative exploration of the best possible node mappings, selecting the best mapping at each iteration phase. The authors argue that the advantage of their algorithm is that the iterative process is often able to find the optimal mapping within a few iterations which reduces the run-time. The complexity of the algorithm depends on the number of phases. The best matching can be found within $O(n^2K^n)$ steps, with K being the number of phases and n being the number of nodes in the smaller graph. Experiments have shown that the new algorithm outperforms the error-correcting subgraph isomorphism algorithm it was compared to.

However, tests have also shown that although the new algorithm is better able to handle larger graphs, its run-time performance is still not sufficient for live application as required by the graph-based similarity model process, as the process would require permanent matching and real-time update of the models and matching results to be successfully applied in practice.

Expected Potential for Tailored Adaptation

In addition to the findings of the complexity discussion described in the previous section, some further content-related aspects have to be considered. Presuming, it would be possible to solve the run-time issue satisfiably, and given a real-time capable solution for the graph matching task would exist so that up-to-date graph distance results would be permanently available, the following questions remain:

1. How expressive are the results of graph matching regarding the approach's prediction capability?
2. How valuable is the information which can be gained through measuring graph similarities for later adaptation?

Both can only partly be answered reliably before actually having tested the approach live. Assuming that enough test data would be available to build reliable reference graphs, the approach's prediction capability would still depend on several circumstances. The following list provides the conditions under which the approach would probably be most successfully applicable in practice for predicting future developments in user behaviour:

- A user's behaviour remains relatively steady, even if situational conditions, like, e.g., a group constellation, change. This implies that the influence of external factors on users' behaviour is less powerful than the user's general attitude towards learning, group work, etc.
- Different groups behave similarly under similar circumstances (including, e.g., learning content, group tasks, tools available, etc.).
- Specific tools are used in similar ways by different users.

If these conditions hold, the system's predictions based on reference activity graphs, have the potential of being very precise because it would then be likely that relations found in one process will recur in others. For instance, this could include that users who have successfully cooperated in the past are likely to cooperate successfully again in the future, or that tools that have been used in a combined way in the past are likely to be used in a similar way again in the future.

The field of cooperation is, however, in general, one exposed to several different biasing influences. The selection of a cooperation partner might, for example, be based not only on measurable factors like

- a user's knowledge about a specific topic,
- a user's favourite role in a group,
- a user's experiences with the system in general, and a specific setting in particular, or
- a user's motivation (indicated by the level of activity),

but also on factors that are either not easy to measure or to predict, including

- coincidence,
- personal relationships,
- shared interests in scopes other than online learning, etc.

This, however, does not particularly affect the described approach but has to be considered in general when cooperation analysis should be used for predictions.

Additionally, it should be assured that adaptations based on predictions are in general potentially reasonable. For instance, if the system recognizes a pattern in the current live graph based on the users involved, how would a subsequent adaptation look like? For example, to recommend collaborators for future activities, an ideal system would strive to take into consideration any observable extrinsic factors (other than the involved users / learners) that may affect collaboration. Summing up, the described approach could potentially provide valuable results regarding predictions and subsequent adaptations if specific conditions are held. If this is not the case, the approach's success can only be measured after practical implementation.

5.2 Dependency Graphs for Classification

This section explains another theoretical concept of path-based activity classification.

5.2.1 Description

This approach is similar to the concept discussed in Section 5.1 in aiming at extracting semantic information about dependencies from activity data, but it operates on a higher level, i.e. its results can be more dependable. The kind of sequential information which is analyzed is very similar, however, the information is represented differently in order to avoid the necessity of complex graph matching tasks during a real-time process.

This approach mainly aims at gaining information about the relations between users by analyzing the sequential context of activities. Again, the high-level aim is to determine general collaboration behaviour and, based on that, later offer personalized recommendations to achieve adaptive (collaboration) support. On a level further down, the approach should facilitate the comparison of collaboration behaviour across sites and tools.

The approach is therefore only well applicable on data as in Data Set I (see Section 3.2.1). Data as in Data Set II (see Section 3.2.2) cannot directly by used here because it does not contain direct information about cooperation of users and activities across different tools or sites (i.e., contexts). If the approach described here should be applied on this kind of data, a complex intermediate step would have to be introduced during which artificial information about the users would have to be created, based on their individual behaviour. This can, however, not fully replace “real” collaboration data. The procedure which would then use this speculative models in order to define relations between users, would be a rather precarious one.

The general idea of using detailed, reliable information about an individual in order to infer the behaviour in a group, is a promising one and will be further discussed in Section 5.3. Yet, the intermediate step required here would lead to inaccurate models and subsequently also to inaccurate predictions. Thus, this approach can only be reasonably applied to cooperation data, which is why a discussion of Data Set II is omitted in this section.

Generally, the approach first splits incoming data and organizes it into sites / tools. As not all activities which occurred in the same site and tool, are potentially related, criteria to determine the probabilities for relations must be defined. These criteria can be different for different tools, e.g. for an asynchronous communication tool, the time frame in which related activities can occur is much longer than in synchronous communication tools. Thus, we find a default setting for splitting tool activities into time slots.

A default time slot does not necessarily have to be created based on time only, which would result in n equal slots of, for example, 5 minutes, but also on additional considerations like semantic similarities. For every slot, a construct depicting the relations and dependencies between the participating users is created.

Putting these constructs together, including aging by assigning different weights to the slots, leads to one common tool construct. These tool constructs which are created separately for each site, can be the basis for comparison of user collaboration behaviour in different contextual settings. The procedure is as follows:

1. monitor events (continuously, as the process can be applied at run-time)
2. organize according to sites and tools
3. divide into time slots as configured separately for every tool
4. for every time slot, retrieve participants’ basic statistical information, e.g. total number of activities, ratio of active (e.g. create, update, delete) and passive (e.g. read) activities, number of dependencies (e.g. if an entry in a forum is related to another one), etc.
5. construct two-dimensional lattice (i.e. matrix) of users, with the nodes storing relations between the users on the two axes; the two entries for the users u_1 and u_2 can differ, i.e., a user u_1 might be related to u_2 in another way than the one in which u_2 is related to u_1
6. construct a simple, easily traversable, comparable, and visualizable representation of the lattice data, e.g. graph-based
7. compare results for different tools and sites, identify recurring patterns and user behaviour

8. define “success” for different scenarios (e.g. passing a task or reaching a certain intensity of communication)

9. prepare the resulting constructs to go through classifiers

10. train the classifiers with sample constructs manually classified (e.g. as successful, i.e., if a group of users passed a test, or not)

11. continuously send resulting constructs through classifiers and determine potential success of user combinations, which then becomes the basis for recommendations of e.g. communication partners or whole group constellations

5.2.2 Process and Objectives

The process just described provides matrices and graphs as result, which provide information about the current group constellation. The high-level aims include the suggestion of group constellations based on the prediction of success or failure of arbitrary groups, again based on individual users' collaboration models. A user's collaboration model can e.g. contain information about

- a user's general level and intensity of activity,
- a user's level of active collaborative activity,
- a user's level of passive collaborative activity,
- a user's number of collaborating other users,
- the number of tools the user was active in, or
- a user's preferred category of tools (e.g. synchronous communication tools).

The task of grouping users raises some central questions like, for instance:

- What factors contribute to the success of a group?
- What group-work-related theories do we want to consider?
- What characteristics of users are relevant for the group-building process?
- What characteristics of a user influence the processes in a group most?
- What roles are most important in a group context?

Metrics and Formulae for Table Construction and Graph Conversion

The formulae described in this section are grounded on an approach used for dictionary construction, as explained by [Salton, Gerard, 1968]. Although dictionary construction does not seem to be related to the approach explained here at first glance, there are similarities between the construction of a collection of words and the construction of a collection of relations at the technical, not semantical, level that make the elements easily adaptable to what is needed here.

Salton describes the process of dictionary construction as several steps with the aim to construct a hierarchy of words. One of several possible hierarchy formation processes is based on a term-document or a term-property matrix (with a property vector describing every term). The relations between the terms in Salton's approach can be compared to those between users, based on their activities, in the data used here. Salton defines relations between terms (identified by weighted property vectors) in a dictionary as non-symmetric and lists the following possibilities:

1. two terms have different properties, i.e. they are unrelated,
2. two terms have the same properties and reasonably similar weights of the properties, i.e. the terms belong to the same class,
3. two terms have the same properties but the property weights are higher for the first term, i.e. the first term dominates the second one and is placed on a higher level in the hierarchy,
4. two terms have the same properties, but the second one dominates the first one (analogous to 3.)

Salton then uses an asymmetric similarity coefficient so that the similarity between two terms i and j is not necessarily the same as between j and i. The coefficient is computed as

$$c_{ij} = \frac{\sum_k min(v_k^i, v_k^j)}{\sum_k v_k^i} \tag{5.1}$$

with v^i and v^j being two k-dimensional property vectors representing the terms i and j. The values of the similarity coefficient are then used to fill a term-term correlation matrix. Here, a variation of Salton's approach is used to measure the relations between users based on their activities. First of all, we can define the following possibilities:

1. two users' activities are all unrelated, i.e. the users are unrelated,
2. two users' activities are (partly) related, the first user dominates the second one (i.e. more activities of the second user are related to the ones of the first user),
3. two users' activities are (partly) related, the second user dominates the first one (analogous to 3.), and

4. two users' activities are (partly) related, but none of the users dominates the other one.

We further compute the (asymmetric) Relational Collaboration Coefficient (RCC) (see Equation 5.2) based on the percentage of related activities (given the total number) and use the values to fill a user relation matrix. The RCC is in general used to measure the relation, i.e., dependency of a user u_1 to / on another user u_2. As already mentioned, the two RCC values for two users can differ, i.e., $RCC_{u_1u_2} \neq RCC_{u_2u_1}$ because one user might be strongly dependent on another user whereas in the other direction there is no dependency at all.

To compute a user's level of activity and the RCC, the following metrics are used: $TOT_{u_n(u_m)}$, i.e., the ratio between a user's number of activities depending on activities of another user and the user's total number of activities in the respective tool time slot, AVG_{u^*}, i.e., a user's expected number of activities, based on the number of users and the total number of activities in that slot, and $RELP_{u_n}(u_m)$, i.e., a user's relation (parental) to the other users, depending on how many of this user's activities follow other user's activities (i.e., are directly related to them).

$RELP_{u_n}(u_m)$ is computed as the ratio between the number of n's activities that depend on m's activities, and the total number of n's activities. An activity depending on another one, could, for example, be reading an entry in a forum. In this example, the *read* activity depends on the *new* or *create* activity of this forum entry, i.e., without the entry having been created, reading would not have been possible. The $RCCs$ for two users u_n and u_m can be computed as in Equation 5.2.

$$RCC_{u_nu_m} = \begin{cases} \frac{min(RELP_{u_n}(u_m), RELP_{u_m}(u_n))}{RELP_{u_m}(u_n)} * w & if \quad RELP_{u_n}(u_m) > 0 \quad and \\ & \quad\quad RELP_{u_m}(u_n) > 0 \\ w & if \quad RELP_{u_m}(u_n) = 0 \quad and \\ & \quad\quad RELP_{u_n}(u_m) > 0 \\ 0 & \quad\quad otherwise \end{cases} \tag{5.2}$$

where

$$w = \begin{cases} \frac{TOT_{u_n(u_m)}}{AVG_{u^*}} & if \quad TOT_{u_n(u_m)} < AVG_{u^*} \\ 1 & \quad\quad otherwise \end{cases} \tag{5.3}$$

$TOT_{u_n(u_m)}$ is determined by the total number of n's activities depending on m in the respective time slot. For computing the weight w, this result is in the standard case divided by the user's expected activity value (determined by the total number of activities in the time slot divided by the number of users active in the time slot). For instance, if one activity of user n depends on m, and the time slot contains 8 activities and 3 users in total, $TOT_{u_n(u_m)}$ would be 1 and AVG_{u^*} would be ≈ 2.67 which would result in a weight $w \approx 0.37$.

In order to provide a data representation that is easily readable for both humans and machines, the matrix is converted to a directed weighted graph which also allows for simple visualization. The matrix axes (here, the users) become nodes in a graph, the values in the matrix cells become the weights in the graph, annotating its edges. Consider activity data as shown in

Table 5.3: This table shows simple example activity data for one time slot of tool *forum* in site s_1. In addition to index, tool, activity, user and site, it also contains a reference to a resource, which is either a previous item, identified by its index, or $[idx_0]$ if no parent exists (e.g. a new, independent forum topic).

Index	***Tool***	***Activity***	***Resource***	***User***	***Site***
3	forum	new	$[idx_0]$	u_1	s_1
5	forum	read	idx_3	u_2	s_1
6	forum	new	idx_3	u_2	s_1
7	forum	new	$[idx_0]$	u_1	s_1
8	forum	read	idx_7	u_3	s_1
11	forum	read	idx_6	u_3	s_1
13	forum	new	idx_6	u_1	s_1
14	forum	read	idx_{13}	u_3	s_1

Table 5.3 for an example. The matrix in Figure 5.3 can be visualized by the graph in Figure 5.4.

In this example, the activities of user u_1 are not strongly dependent on others, whereas the activities of both u_2 and u_3 tend to be related to those of u_1. This can be seen by the high values of $RCC_{u_3u_1}$ and $RCC_{u_2u_1}$, and the low values of $RCC_{u_1u_2}$, $RCC_{u_1u_3}$ respectively. The mean of incoming weights in the graph for a user node (equivalent to the vertical average of a user's corresponding RCCs in the matrix) indicates this user's level of activity, combined with the influence on the other users. A high value marks this user as important for the collaboration processes of the group, a low value marks this user as a rather passive collaborator.

This additional metric will be further referred to as $RCC^+_{u_n}$ for a user u_n. Given a total number of m users in this slot, the value is computed as shown in Equation 5.4. The corresponding converse value $RCC^-_{u_n}$ (Equation 5.5), representing a user's dependence on other users' activities, is computed as the mean value of a node's outgoing edges' weights in the graph (equivalent to the horizontal average of a user's corresponding RCCs in the matrix).

$$RCC^+_{u_n} = \frac{\sum_1^m RCC_{u_m u_n}}{m-1} \tag{5.4}$$

$$RCC^-_{u_n} = \frac{\sum_1^m RCC_{u_n u_m}}{m-1} \tag{5.5}$$

The values for users u_1, u_2 and u_3 of our example scenario (Table 5.3, Figures 5.3 and 5.4) are listed in Table 5.4.

Given these metrics, we finally want to use the model to infer information about the behaviour of the group. This could, for example, be achieved via classification (see also Section 4). Classification can be done in different ways, e.g., we want to be able to predict a group

	u_1	u_2	u_3
u_1	0.00	0.12	0.00
u_2	0.75	0.00	0.00
u_3	0.75	0.38	0.00

Figure 5.3: Matrix storing the *RCC* values for three hypothetical users in the tool *forum* (in general, a tool matrix contains all users with activities in that specific tool).

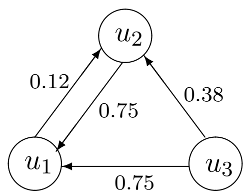


Figure 5.4: Graph representation of the matrix in Figure 5.3. The 0.00-values can be omitted, as an edge weighted 0.00 does not exist.

Table 5.4: This table shows the RCC^+ and RCC^- values for the three users of the example. These values can be indicators for predicting the potential "success" of specific group constellations. For instance, grouping many users of a collaboration behaviour similar to u_3 will most probably not lead to a high intensity of collaborative activities, as these users may all tend to be rather passive and depend on other users' initiative.

User n	$RCC^+_{u_n}$	$RCC^-_{u_n}$
u_1	0.75	0.06
u_2	0.25	0.38
u_3	0.00	0.57

constellation to be potentially successful or the opposite. "Success" regarding group work can be defined in several ways, e.g.

- passing an assignment or test,
- obtaining a certain grade in an assignment or test,
- reaching a certain level of communication intensity,
- reaching balanced communication behaviour among the students, or
- reaching a certain level of task sharing and role allocation

As classification should be performed on the basis of interrelated activity sequences, the activities cannot be sent through the classification process as described in Chapter 4. Another approach would be to include several items' data as features in a data set which then goes to the classifiers. However, as the number of items in every time slot can differ drastically, classification on the basis of time-slot-data is not possible. Yet, a fixed number of items can be used for one data set and include the corresponding time slot index as an additional feature in the feature set for classification data.

Another possibility is to build the feature set based on the metrics described before, i.e. activity data will not be sent through the classifier in its original form, but be preprocessed first, creating a new data set for e.g. every time slot, containing elements like mean, standard deviation, variance, minimum and maximum RCC^+ and RCC^-, number of elements in this time slot, etc. Furthermore, standard matrix and graph metrics like sparsity or connectivity can be included.

Given the results of the classifiers, one can extract information about what leads to successful collaboration and what does not seem to influence the success of a group at all. This information can then be used to add contextual information to the knowledge gained by classification of independent activities. Furthermore, it could potentially reveal new information about collaboration in general and collaborative learning in specific.

Tool Implications and Additional Relations

The example introduced before is based on data produced in one specific tool, the forum. However, the approach cannot be applied in exactly the same way for all kinds of tools. First, information about relations between items differs for different tools, and second, the relations themselves may differ. Therefore, the concept must be slightly refined for different kinds of tools.

For instance, we need to know the $RELP$-values if we want to compute RCC values for activity items. There are tools where it is not possible to derive the parental relationship between items as easily as for the forum. Our computation until now is based on the fact that every item stores a "resource" the respective item is related to, i.e. the parent activity. This information is not available for items of every tool. For instance, the relations between chat activity items can hardly be established without a semantic analysis of the message contents. Thus, only the relations based on chronological order are obvious in this case.

In general, this means, the relations have to be defined manually. Thus, tool categories are introduced in order to group tools on the basis of the nature of the relations between their items and the approximate size of the time slots. The most significant distinction regarding the way of organizing items can be made between synchronous and asynchronous tools. Therefore, this becomes the first-level categorization although one may consider the distinction between communication facilities and tools supporting learning semantically more important.

Further, synchronous tools will be referred to as category S and asynchronous ones as category A. Category S includes tools like chat and audio / video conferencing. The tools in category A can be further divided into communication facilities (A_c) – including forum, announcements, email, private messages, or roster (i.e. user profiles in Sakai), and learning support tools (A_l), including assignments, resources, wiki and syllabus.

Tool Category A_c These tools have in common that their items contain information about their respective parent. This makes it easy to determine the RCC values. Therefore, only a reasonable time slot must be defined. A time slot in this category can be quite long, as related activities can occur with a significant delay.

Tool Category A_l The tools in this category are to be treated similar to the ones in A_c. However, the time slots have to be even longer as learning activities can theoretically be related with months between them (e.g. a user reading a document which has been uploaded at the beginning of a semester and is relevant for a final exam at the end of the semester). In this category there can be several different configurations for different tools.

Assignments, for example, are likely to have a shorter time slot than resources, wiki or syllabus. For an assignment, there is a specific time period during which the assignment must be delivered. This period can be defined by the responsible lecturer or tutor. This predefined time is used for a time slot in the assignments tool, which means that it can vary.

For the syllabus, one may want to consider all activities within the duration of a course which usually is one semester, or a project, which can vary. The resources tool can be treated similar to the syllabus.

For the wiki, however, more individual treatment has to be considered, as it is a more interactive tool which potentially provides more details about users' behaviour. As the wiki can be revised and extended by all participants, it can reveal fine-grained information about multiple aspects of group-work-related attitudes. For instance, a user can act entirely passively, or, to the contrary, actively contribute a lot. Other users may tend to correct other users' contributions rather than just reading them. Possible metrics are:

- the number of **original** contributions to a wiki (i.e. contributions not related to previous ones)
- the number of **independent** contributions to a wiki (i.e. contributions possibly related to, but not revising, previous ones),
- the number of **revising** contributions,
- the average time between an original contribution and the revision, or
- the ratio between revising own and revising other contributions.

Tool Category S These tools generally require a fine-grained elaboration. Some of them do not store parental information in their items at all, simply because there is no context defining a clear parent. For instance, items in a chat tool are usually structured according to chronology only. In order to find a basis for RCC computation we have to add an additional layer of relation. All possible ways of approaching this depend on assumptions on a certain level.

All of a user's items are treated as potential parents if they occur between the last one and the same user's last-but-one activity. Additionally, we use a limited number of potential parent users, considering all different users whose activities occurred within the defined time. If the number of users there is higher than the limit, they are selected in a descending order, starting from the most recent item.

The time slots in this category are much shorter than the ones in A_c, because related activities mostly occur with a short delay only, otherwise they would not be considered related any more. As the number of items drastically depends on interaction intensity here, also variable time slots should be considered. In audio / video conferencing, the borders of a session are mostly clearer than in chat. Therefore, it may not be necessary to have time slots but given sessions can be used instead.

5.2.3 Evaluation and Potential Shortcomings

Compared to the concept described in Section 5.1, this approach is similar in several ways:

- Cooperation data is needed in order to apply the approach. Individual user data sequences like ITS activities are not sufficient.
- The approach aims at extracting information about relations.
- Information about relations should be used as a basis for predictions and subsequent adaptations.

Despite these similarities, the approach is different from the one described before in several ways:

- Information, although based on activity sequences, is not modeled by potentially very complex graphs any more.
- It can more easily be applied at real-time.
- Relevant information is extracted in an earlier phase of the adaptation cycle.
- Supplemental information is added to what is extracted from plain activity data.

In general, the characteristics of group activity data and their relevance for predictions identified in Section 5.1.3 for the graph-based similarity approach also apply for the approach described here. Thus, also for this approach, the quality of the analyses and inferences based on them cannot be proven without practical implementation which would have to address the following elements of the approach:

- validation of the model (with more complex and numerous data than presented here for exemplification),
- validation of the theoretical metrics and formulae proposed,
- investigation of concrete, yet parameterized constraints to be used to establish time slots, and
- validation of the way of deducing the quality of collaboration within a time slot from given relationships and dependencies between, e.g., users or tools.

However, as the dependency graphs approach does not rely on complex computations that might render it inapplicable for run-time scenarios, it overcomes one potential shortcoming. Appending supplemental information can potentially make the approach more resistant against external influences because behaviour is not the only basis for inferences any more. This could, if it is overrepresented compared to the activities themselves, also turn out to be a constraining factor.

5.3 Activity Sequence Graphs

This section describes an approach to modeling sequential user activity data that is applicable to different kinds of data, including individual users' activity sequences within an ITS. Parts of the section have been also published by [Köck and Paramythis, 2011].

5.3.1 Description

The first part of the proposed approach is concerned with modeling activity sequences in a way that allows for analyzing and reasoning over sets of activities performed by different users. However, in order to overcome the shortcomings previously identified, the approach should mainly be able to model the activities of an individual user in a way that allows for later conclusions and inferences regarding the individual's behaviour in a group.

Here, several alternatives to representing sequences have been considered and evaluated, concentrating on the interwoven questions of comparability and generalizability. Specifically, a formalism should be found, that would allow for the comparison of activity sequences that might differ only little (e.g., situations where one sequence might contain more repetitions of one activity than those found in another), but also for comparing sequences with only small amounts of overlap. In general, the modeling of sequential data faces the challenge of not losing information about relations and dependencies between the individual items, in this case, activities.

Different machine learning approaches for modeling sequences were considered. [Dietterich, 2002] lists the most important research issues in sequential supervised learning as follows: loss functions, feature selection and long-distance interactions, and computational efficiency. Although, as further described later, the approach selected here aims at information extraction via clustering, i.e. unsupervised learning, most of these issues are relevant.

Feature selection, for example, plays a crucial role in the process, as discussed in Section 6. Too many features can inhibit the identification of the most significant properties and thus distort the picture, whereas too few features may easily cause total loss of relevant information.

Computational efficiency is also a very important factor for the scenario here – a sequence modeling approach suitable for the given requirements should be applicable at run-time and avoid loss of information. Dietterich lists several machine learning techniques suitable for modeling sequential data: the Sliding Window method, Recurrent Sliding Windows, HMMs and related methods, Conditional Random Fields (CRFs), and Graph Transformer Networks (GTNs).

The *Sliding Window* method (see different applications in, e.g. [Sejnowski and Rosenberg, 1987], [Quian and Sejnowski, 1988], or [Fawcett and Provost, 1997]), converts a sequential learning problem into a classical learning problem. The method uses a window classifier that is trained with input data that has been converted into windows (each representing and treated as a sequence). The sliding window method is not bound to specific algorithms but can use any machine learning technique. The windows are not necessarily static but could also be of dynamically adapted size [Ortiz et al., 2011]. However, the method is not capable of identifying dependencies between the outcome values, a potential shortcoming which is overcome by the *Recurrent Sliding Windows* technique.

The *Recurrent Sliding Windows* technique feeds the predicted value for a specific data instance into the system to help make the prediction for the next instance, i.e. the most recent predictions are used as inputs (the size of this "window" depends on the respective application scenario). [Lichtenwalter et al., 2009], for example, describe an approach using recurrent sliding windows for musical classification. [Bakiri and Dietterich, 2002] apply recurrent sliding windows in combination with a decision tree algorithm to the English pronunciation problem.

In their evaluation, the recurrent sliding window technique drastically improved the results of the original sliding window method.

Markov Models are probabilistic models similar to finite state machines consisting of a set of states $S = \{S_1, S_2, ..., S_n\}$, an NxN matrix containing state transition probabilities $A = \{a_{ij}\}$, and a vector of initial state probabilities $\pi = \{\pi_i = P(q_1 = S_i)\}$. This model is then used to compute the probabilities for specific output sequences.

HMMs (see, for instance, [Rabiner, 1989]) are special cases of Markov models because the states are hidden, i.e., not observable. The hidden states form a traditional Markov model and can produce a set of different outputs, i.e., observable effects. HMMs are a popular way of modeling sequential data in order to be able to provide predictions for specific activity sequences, see, for example, [Beal et al., 2007], [Seymore et al., 1999], [Soller et al., 2005], [Soller, 2007], or [Soller and Lesgold, 2007].

However, [Dietterich, 2002] identifies a principle drawback of this methodology and states that the "structure of the HMM is often a poor model of the true process producing the data", a problem which originates in the Markov property (i.e. the probability of a system being in a particular state S_j at time t does not depend on the entire history, but only on the previous state at time $t-1$); a relationship between two different y values, for example, y_1 and y_3, must be communicated via the intervening ys.

A Markov model where the probability $P(y_t)$ only depends on y_{t-1} cannot generally capture these relationships. This problem is generally addressed by sliding window techniques. Using sliding windows for HMMs is however difficult, because an HMM generates each x_t from the corresponding y_t only. [Dietterich, 2002] argues that this problem could theoretically be overcome by replacing the output distribution $P(x_t|y_t)$ by a more complex distribution $P(x_t|y_{t-1}, y_t, y_{t+1})$, which would allow an observed value x_t to influence all three y values but is difficult to put into practice because it is not clear how to represent this complex distribution compactly.

[Dietterich, 2002] lists the following approaches to overcome these limitations: *Maximum Entropy Markov Models (MEMMs)* (see, for example, [McCallum et al., 2000]), *Input-Output HMMs (IOHMMs)* (see, for example, [Bengio and Frasconi, 1996]), and *CRFs* (see, for example, [Lafferty et al., 2001] or [Vail et al., 2007]). All of these approaches are conditional models that, unlike standard HMMs which try to explain how observed sequences are generated, represent conditional distributions of output sequences given input sequences, i.e. they try to predict output values given input values.

IOHMMs and *MEMMs* are quite similar in the way they are trained and both suffer from the same issue called *label bias problem*, i.e. there is a bias toward states with fewer outgoing transitions (states with a single outgoing transition ignore their observations), see a more detailed description in [Lafferty et al., 2001].

CRFs, which are mostly used for labeling sequences, are an approach to overcome the label bias problem (see [Lafferty et al., 2001] or [Vail et al., 2007]). In the CRF, the way in which adjacent y values influence each other is determined by the input features [Dietterich, 2002]. In the experiments presented by [Lafferty et al., 2001], the CRF outperforms HMMs and MEMMs regarding modeling accuracy, but it is fairly slow in comparison to the other approaches.

GTNs (see, for example, [Bottou et al., 1997] or [Bottou and Le Cun, 2005]) are neural-network-based models that transform input graphs into output graphs [Dietterich, 2002]. For example, an input graph consisting of a sequence of inputs x_t is transformed into a graph of u_t outputs, where every x_t is a feature vector attached to an edge of the graph, and every u_t is a pair of a class label and a score. The graph of the u_t scores is then analyzed with the aim of finding the path with the lowest total score. Also this methodology aims at solving complex supervised learning problems rather than unsupervised ones.

In general, most techniques for modeling sequential data as presented here, are tailored to the use in classification tasks, i.e. supervised learning. Here, however, sequences should also be modeled for clustering, i.e. unsupervised learning. Thus, a way to represent sequences must be found that allows for simple transformation into another format processable by unsupervised learning algorithms. The reasons for the decision to use DMMs can be summarized as follows:

1. Markov models have been successfully used in the past for similar purposes in the context of modeling activities [Soller and Lesgold, 2007] [Soller, 2007] (see also Section 2.2).
2. The states themselves are observable (see the description below), therefore there is no need to use hidden models.
3. Traditional statistical representations are likely to lose information bound to not the activities themselves but the relations and dependencies between them (see the example below).
4. The approach must be suitable for unsupervised learning and models must be convertible to other formats that can be fed into a clusterer, or serializable without information loss.
5. The modeling process itself should not be too expensive concerning its run-time behaviour.

5.3.2 Process and Objectives

To better motivate the choice of representation, a concrete example of modeling problem-solving sequences is considered. As a first step, the activity sequences must be converted into DMMs, thus identifying states, state transitions and transition probabilities, and initial state probabilities. Again, Data Set I and Data Set II introduced in Section 3.2 are used to provide examples a and b. Regarding example a, the resulting models themselves are quite similar to the models discussed in Section 5.1. Comparing the models for Data Set I, depicted in Figure 5.1, Figure 5.5 respectively, only the following differences can be identified:

- the edges in Figure 5.1 have been turned into transitions in Figure 5.5 and the edge indices disappeared
- the edge weights in Figure 5.1 have been turned into transition probabilities (i.e., values between 0 and 1) in Figure 5.5
- the nodes in Figure 5.1 have been turned into states in Figure 5.5 (although the states have not been augmented with initial probabilities as the DMM would suggest – due to the fact that the data does not entail the necessary information)

Table 5.5 provides the activities Figure 5.5 is based on. However, although the models seem similar, the further process differs drastically: while the approach described in Section 5.1 aims at identifying similarities in activity sequences by measuring the similarities in the corresponding graphs, the approach described here aims at having similarities and differences in the models determined automatically by means of unsupervised learning, i.e., clustering.

Furthermore, it becomes obvious that the DMM approach, which relies on information about its states and transitions, is not best applicable for this example because the activities are continuous and can thus not be put into predefined frames which could allow for the determination of the initial probabilities. This would only be possible if we could determine the beginning and end of a sequence, which is not possible if sequences are considered across different tools, users, etc.

For this approach should, as already explained, be particularly well applicable for modeling individual users' activity sequences produced, for instance, in an ITS, the relations between different users, tools, resources or sites are only marginally relevant here. Thus, Data Set II provides the better example (b) in this case and will further be concentrated on.

As already introduced in Section 3.2.2, in the Andes system, a problem-solving sequence contains all of a user's activities related to any unique step associated with the problem (KC). Thus, a solving sequence for a specific problem looks different for each user. Consider the following scenario: two hypothetical users U_1 and U_2 are working on the same topic T_1 which consists of three KCs KC_1, KC_2 and KC_3. The activity sequence of U_1 is also shown in Table 5.6.

User U_1 solves problem KC_1 correctly (further referred to as C, i.e., correct) at first attempt, but fails (further referred to as I, i.e., incorrect) first at KC_2. Next, the user requests a hint of type "what's wrong" (further referred to as H_1) that is followed by two hints of type "next step help" (further referred to as H_2), and then solves the problem correctly at the second attempt. This results in the activity sequence $I \rightarrow H_1 \rightarrow H_2 \rightarrow H_2 \rightarrow C$. A similar pattern is observed for KC_3: $I \rightarrow H_1 \rightarrow H_1 \rightarrow C$. For user U_2 we observe: KC_1: $I \rightarrow H_1 \rightarrow I \rightarrow H_2 \rightarrow C$, KC_2: $H_3 \rightarrow C$, KC_3: $H_1 \rightarrow I \rightarrow H_2 \rightarrow C$.

Table 5.5: This table shows a small test data set containing different events that occurred within an e-learning platform in the order as listed here. A data instance is described by an index, the tool in which the event occurred, the type of event (i.e., activity), the related resource, the user responsible for the event, and the site in which the event occurred. Indices 4 and 5 do not provide a site because they occurred in an overview-page that is not handled as a course page.

Index	*Tool*	*Activity*	*Resource*	*User*	*Site*
1	calendar	new	resource4	user3	site3
2	announcement	new	resource5	user3	site3
3	content	new	resource6	user3	site4
4	roster	view	resource7	user4	null
5	forum	new topic	resource8	user4	null
6	announcement	new	resource9	user4	site5
7	announcement	new	resource10	user4	site6
8	calendar	new	resource11	user5	site6
9	announcement	new	resource12	user5	site6
10	content	new	resource13	user5	site7

Describing these sequences with basic statistical means, one may obtain results such as these: both users have successfully completed the topic; user U_1 submitted two incorrect answers in total and requested five hints, user U_2 submitted three incorrect answers and requested five hints. A comparison of these results might lead to the conclusion that the performance of U_1 and U_2 at T_1 was similar. Even if the comparison considered the level of steps, the result for the two users at KC_3 would be equal although the actual sequences were different, i.e., one dimension of the information is lost.

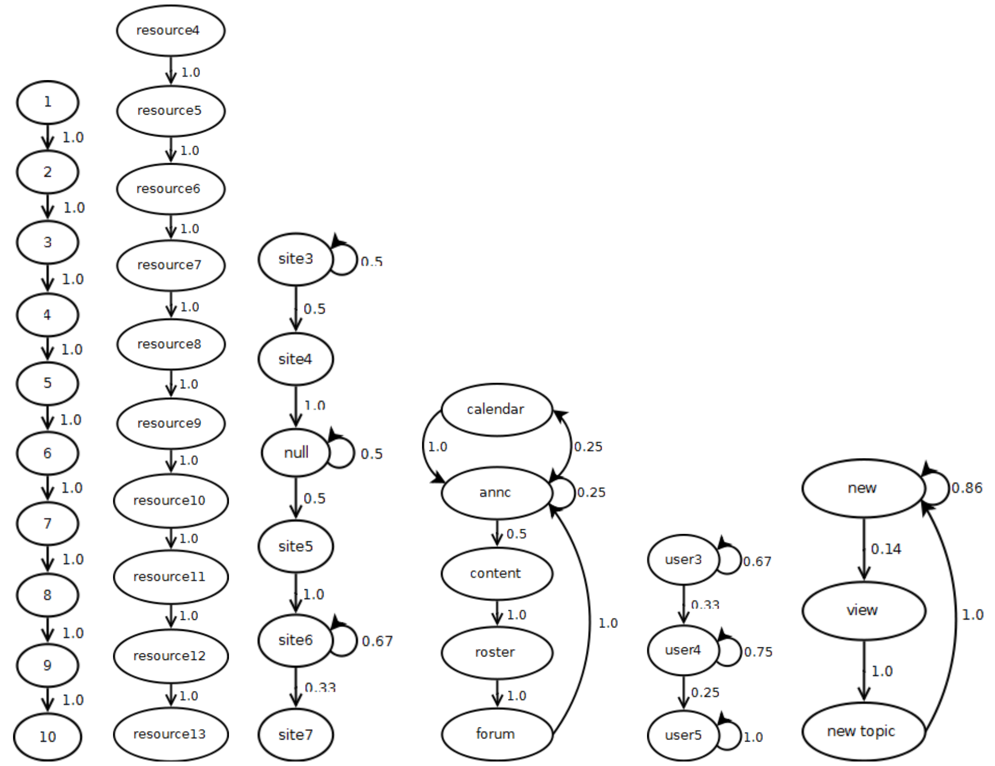
Table 5.6 lists example activities based on Data Set II described in Section 3.2. The resulting models are shown in Figures 5.6 and 5.7. The activities listed in Table 5.6 are slightly different to the ones used for the graph-based similarity model described in Section 5.1. Table 5.2 in Section 5.1 comprises activities across different topics, whereas Table 5.6 only contains activities of one student (S_1) within the scope of the same topic (T_1) due to the fact that this approach aims at creating one DMM for every student / topic combination. Therefore, activities within two topics would lead to two different models. Here, Topic1 was chosen for demonstration purposes.

As already discussed, the premise of the presented work is that retaining this kind of sequential activity information in the modeling process can enhance several stages of the adaptation cycle by offering fine-grained user model input on a behavioural level.

As mentioned earlier, Markov models are a convenient way to represent sequences in a way similar to finite state machines. They can be applied for situations where the states are known, but also if the system can only work with observations and does not have information about

Table 5.6: This table shows a small test data set containing different events that occurred within an ITS in the order as listed here. A data instance is described by an index, the student who caused the event, the type of the student's response (e.g., Hint Request), the more detailed type of the student's response (e.g., the request to get information about what was wrong with the student's answer), the topic, the so-called "Knowledge Component" (i.e., the concrete problem), and the unique step (i.e., a unique part within a specific problem).

Index	*Student*	*Student Response Type*	*Student Response Subtype*	*Topic*	*Knowledge Component*	*Unique Step*
1	Student1	Answer (Correct)	Answer (Correct)	Topic1	KC1	S1
2	Student1	Answer (Incorrect)	Answer (Incorrect)	Topic1	KC2	S1
3	Student1	Hint Request	What's Wrong	Topic1	KC2	S2
4	Student1	Hint Request	Next Step Help	Topic1	KC2	S3
5	Student1	Hint Request	Next Step Help	Topic1	KC2	S4
6	Student1	Answer (Correct)	Answer (Correct)	Topic1	KC2	S5
7	Student1	Answer (Incorrect)	Answer (Incorrect)	Topic1	KC3	S1
8	Student1	Hint Request	What's Wrong	Topic1	KC3	S2
9	Student1	Hint Request	What's Wrong	Topic1	KC3	S3
10	Student1	Answer (Correct)	Answer (Correct)	Topic1	KC3	S4



(a) Events (b) Resources (c) Sites (d) Tools (e) Users (f) Activities

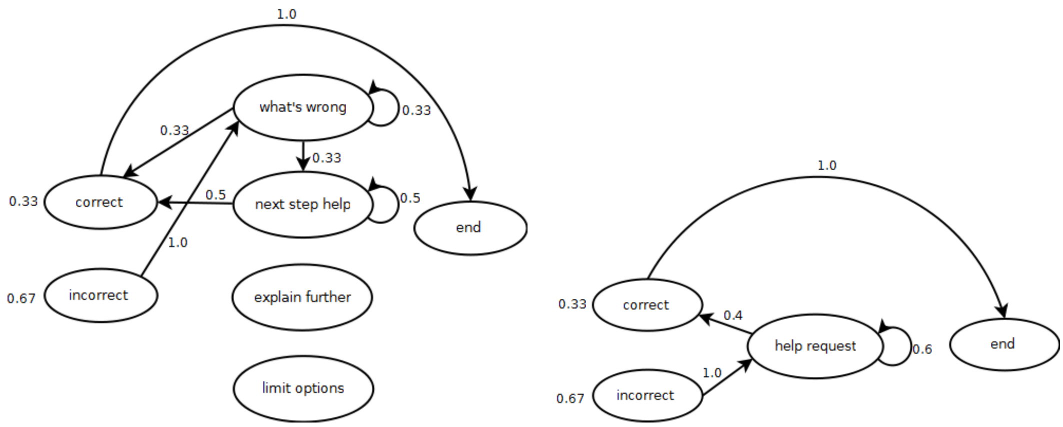
Figure 5.5: Figures 5.5(a) to 5.5(f) show the different graph-based representations of test activity data listed in Table 5.5.

the underlying states that have produced an observation sequence. In a case study, the data clearly suggested a certain configuration of states, therefore there is no need to use HMMs here.

Referring back to the example above, a student has basically two important possibilities of interacting with the system: submitting an answer, or requesting a hint. The system offers four different categories of hints (what's wrong, next step help, explain further, limit options), thus we can differentiate between four different help states in the corresponding DMM.

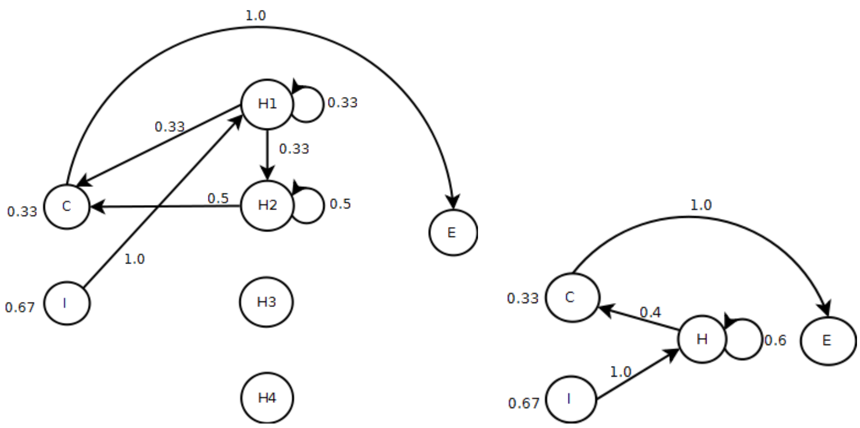
To be able to examine at a later point whether the distinction between hint types influences behaviour analysis, two DMM settings were created for all further experiments, the first with one aggregated hint state and the second with the initial four (different kinds of hint states). Figure 5.6 shows sample DMMs for the two settings, modeling the behaviour of user U_1 solving topic T_1 from the example above.

In addition to the obvious states "correct", "incorrect" and "hint", an artificial "end" state is added. The end state is needed in order to distinguish between the transitions within a single step and the transition to a new one. If the user starts a new step, the system inserts a



(a) Extended (b) Aggregated

Figure 5.6: Figures 5.6(a) and 5.6(b) show two DMMs produced by the activity sequence in Table 5.6. Probabilities with 0-values are omitted here. The numbers next to nodes denote their prior probability, the numbers next to transitions denote the transition probability.



(a) Extended (b) Aggregated

Figure 5.7: Figures 5.7(a) and 5.7(b) show the two DMMs of Figure 5.6, using more general identifiers for the states.

transition from the current state to the end state, thus completing the step. Figure 5.7 shows the same models using the more general names for the different types of help that are also further used in the next chapters.

The DMM-based problem-solving sequence models were subsequently serialized and converted to the common Attribute-Relation File Format (ARFF [1]). Serialization in this case means that for all elements of a DMM a feature was created (e.g., for the prior probability of the state C, or the transition probability from one state to another).

[1]See more information about ARFF at `http://weka.wikispaces.com/ARFF`

In addition to the aforementioned activity sequence information, basic statistical data was used, in order to be later able to compare the clustering performance not only for different settings and different years but also according to different clustering aims and with different aspects of the same raw data.

Using these data sources in isolation and in combination gave rise to a total of three data sets that were used for clustering later: SET_MARKOV, including only the information provided by the learned Markov models (i.e., prior probabilities for the states and transition probabilities between the states), $SET\_STATISTICAL$, including very basic statistical information (i.e., the percentage of incorrect attempts, the percentage of help requests and the percentage of unfinished steps), and SET_BOTH combining the previous two.

All features that were later used in the context of the description of the experiments and results are listed and explained in Section 3.2.2.

5.3.3 Evaluation and Potential Shortcomings

In general, the activity sequence graphs approach brings along the following characteristics:

- It is well applicable for ITS data treating users as individuals.
- It provides detailed information about individual users' behaviour.
- External factors influencing a user's behaviour are reduced to a minimum.
- The resulting models are not complex at all and can be easily converted to a format processable by a clusterer.
- The clustering process does not have to be permanently applied at run-time; information acquisition and adaptation are two different phases only one of which needs to be integrated at run-time.

As the approach does not rely on the existence of cooperation data, it is applicable for different scenarios and platforms. The information gained by the analysis of individual users' activities can be used not only as a basis for adaptations in the area of this user's personal learning environment but also for collaboration support. As there is only little influence of external factors that are not obvious to the system (like, for instance, shared private interests of users), the information gained is likely to be comparatively precise.

Yet, the fact that the approach "works" with individuals' data here could be problematic in cases where one is exclusively looking for a solution that analyzes collaboration data. The models as presented in this section would have to be revisited in order to accommodate collaboration data, which would result in new data sets that then could be treated and analyzed in a way similar to the one described here. This is, however, not the main goal pursued in this thesis – here, information relevant for adaptive support should be extracted from users' activities or activity sequences, which could well also be individual users' data. Thus, a practical implementation of this approach would have to ascertain that in the context of the analysis of individual users' activities, information that is relevant not only for individual user support but also for group or collaboration support, can be extracted.

The fact that the resulting simple models are easily convertible to a format processable by a clusterer is beneficial because then unsupervised learning can be applied in a relatively straightforward way (see Chapter 6).

However, note that the modeling choices made here (especially the use of DMMs), as well as the selection of features with which to populate the data sets used in the clustering stage, have been tailored to the specific needs of modeling problem-solving activity sequences. Chapter 9 discusses factors researchers may want to consider when applying the proposed approach to other domains of learning.

5.4 Comparison and Selection

This section provides a brief discussion on the selection of one of the approaches that were presented, for implementation. The graph-based similarity model presented in Section 5.1 is further referred to as approach 1, the dependency graphs approach described in Section 5.2 is further referred to as approach 2, and the activity sequence graphs approach introduced in Section 5.3 is further referred to as approach 3.

Approaches 1 and 2 belong to the same category of approaches that uses multiple users' cooperation data in order to retrieve information that can be used as a basis for predictions and adaptations. Approach 3 represents the second category of approaches that does not rely on cooperation data being available but can also use individual users' activities in order to infer potential collaborative behaviour that is then used as an adaptation basis.

Approaches 1 and 2 are thus both limited to a specific kind of environments, i.e., environments where users are aware of each other and can cooperate. As already discussed before, this does not only exclude other environments but is potentially also prone to external factors influencing users' behaviours which could make inferences inaccurate. Predictions and adaptations based on individual users' data could however involve another danger: even though individual users, if not influenced by others, tend to reveal their general attitudes towards learning or problem-solving, inferences on potential collaboration behaviour are speculative.

Therefore, all three approaches have in common that, as a final step, practical implementation is needed in order to prove that the original assumptions are correct. Thus, the most promising approach is selected and its practical implementation is described in the following chapters.

Regarding approach 1, in addition to the fact that is is not applicable to, for example, ITS data, computational complexity represents a problematic factor that most probably renders the approach inapplicable or only partly applicable for run-time settings, which would, however, be desirable for this scenario.

Although the models created by approach 2 differ drastically from the ones created by approach 1, the general idea is rather similar. Approach 2 relies on the same kind of data but overcomes several shortcomings.

First, due to the way of modeling sequences, resulting in very simple models, followed by this approach, the problem of computational complexity does not arise, which is why the approach is better suitable for run-time application.

Second, as already mentioned in Section 5.2.3, approach 2 is a bit less likely to become inaccurate due to external influencing factors because of the supplemental information considered during the modeling process. Thus, approach 2 can be regarded an improvement over approach 1.

Both approaches however rely on information about a specific group constellation given a concrete setting, topic, etc. The information gained about collaboration might be reasonable but it might also be context-specific and not fully relevant for different settings.

Approach 3 overcomes the problem of constrained applicability as it is well implementable for ITS data produced by individual users. It is further computationally cheap and potentially provides reliable, unbiased information about users' behaviour.

To sum up, for the rest of the thesis and thus also for implementation, approach 3 was chosen due to the following reasons that are particularly relevant in the context of e-learning:

- The approach is not limited to a specific setting (with multiple users being able to interact with each other) but can theoretically be applied on all kinds of data.
- It allows to identify salient e-learning behaviour of the individual.
- It is computationally feasible.
- The influence of external factors on the system's inferences and resulting models is minimized, thus the gained information is potentially highly reliable.

5.5 Summary

As the loss of an information dimension through treating user activities as independent from each other has earlier been identified as a potential danger, this chapter discussed several ways of modeling user activity sequences.

For the reasons just named in Section 5.4, the activity sequence graphs approach was selected as the most promising one out of the three that were proposed.

This activity modeling approach will be used later to transform real-world learner activity data into models that keep information about relations between activities and that are processable by the clustering unit described in the following chapter.

The sequence models, each storing one activity sequence that contains all activities a particular student has performed during solving a particular problem, are, however, not only machine-processable but also interpretable for humans. Thus, the form of the final clustering results (see Chapter 6) facilitate human intervention as it is required at the end of the process.

Chapter 6

A Multi-Targeted Clustering Approach

This chapter[1] describes the proposed three-level clustering approach and experiments that were conducted to demonstrate its feasibility, using real-world data.

As introduced in Section 1.2, the clustering process aims at the extraction of information at the level of behavioural patterns in learners' activity data. It operates on activity sequence models that have been generated from raw activity data as described in Section 5.3. The clustering results should then provide information about the behaviour of learners that can be used as a basis for adaptive interventions in the learning process.

The general overall approach, as depicted in Figure 6.1, can be split into the following different phases:

- pre-processing,
- experimental clustering (i.e., the process of configuration of clustering parameters),
- clustering, and
- analysis.

Pre-processing includes model definition, depending on the nature of the base data and the possible observable activities, a data conversion step that converts the activity sequences to models (here DMMs as described in Section 5.3), and the definition of evaluation metrics which should be able to measure cluster quality.

The *experimental clustering* phase includes a feature selection step that is responsible for the evaluation of the features' characteristics (for instance, their discriminatory capacity) and subsequent feature selection. Next, the optimum cluster setting, i.e., the optimum number of clusters for a specific case, is determined by considering and combining a number of (partly domain-specific) indicators.

Subsequently, the *clustering* phase starts with the determination of clustering goals. As introduced in Section 1.2, we want to generally be able to identify patters on different levels

[1] Parts of the chapter, including most of the formulae, tables and figures, have been also published by [Köck and Paramythis, 2010], [Köck and Paramythis, 2011]. Small portions of it were written by the second author Alexandros Paramythis and were left in this thesis for reasons of completeness.

in learner behaviour (i.e., learner interactions with the system) in order to offer more fine-grained individual adaptations and support later. Thus, a clustering goal can be to identify

1. a predefined problem-solving style, or
2. a predefined learning / problem-solving dimension, or
3. new, previously unknown potential dimensions and styles.

Regarding 1, the concrete style that should be identified by the clustering process, must be defined, and the most relevant features and their respective expected values must be selected, before a data set can be created. This data set is then fed into the clustering process. The initial definition of a style is usually based on a description well supported by the literature.

Regarding 2, the dimension that should be identified must be defined, and again the features with potential to be relevant, must be selected. Here, however, the features themselves are relevant, independent of their values. Again, the initial definition of a dimension should be based on descriptions in the literature.

Regarding 3, no features (and corresponding values) are preselected. Instead, constraints like the maximum number of features that should go into a data set, are identified. This step results in a variety of different data sets that are automatically created and subsequently all processed to the clustering phase.

In all three cases, the phase of *cluster analysis* follows the clustering process, providing the information basis for later adaptations.

6.1 The General Process

Chapter 5 discussed different ways of modeling learner activity sequences in the area of problem-solving. Here, the proposed modeling approach is used to provide a representation of the sequences that can be further analyzed by clustering in order to discover patterns that could be characteristic of

- attributes of the person showing the behaviour, or
- attributes of the context in which the activities take place.

The first is more relevant in this case, as it focuses on the identification of learning or problem-solving styles. Towards this end, the activity models derived before, are clustered, possibly in combination with additional monitored data that usually relates to the activities themselves.

The approach proposed here involves a clustering process that is dynamically (regarding determination of cluster quality and establishment of termination conditions) controlled in order to function towards different analysis goals. Thus, we not only have to determine these goals,

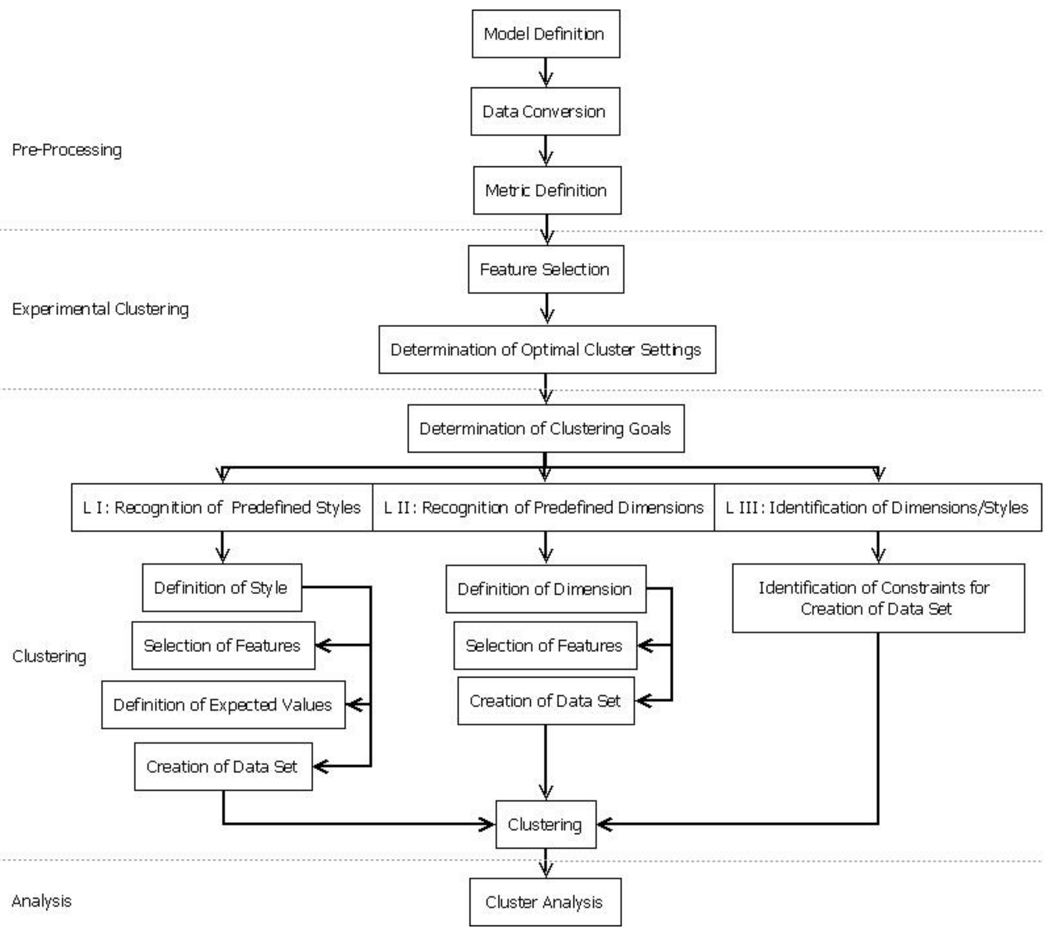


Figure 6.1: Overall process of the proposed approach to sequence modeling and subsequent clustering.

but additionally also need to establish metrics that can be used for this kind of dynamic control. Such metrics can be domain-specific when based on activity semantics, or more general, which renders them applicable for the analysis of different kinds of activities.

The following sections discuss the process briefly lined out before and concentrate on its application in the domain of problem-solving.

6.2 Metrics for the Domain of Problem-Solving

The case study described here aims at the two-fold identification of problem-solving styles via an unsupervised learning process. The first variant comprises providing a description of predefined styles to the clusterer and subsequently detecting these styles in users' activity data, the second (as further described in Section 6.4) aims at the autonomous detection of new styles. Another goal involved in the clustering process is the prediction of success. "Success" in this context implies the completion of a problem by a student through a correct answer.

It is the purpose of the clustering unit to not only feed data sets containing a certain number of data instances (i.e., for example, a serialized version of a DMM) to the clusterer (here, the k-means algorithm [Jain et al., 1999] is used), but to also subsequently analyze the results. The analysis process can vary, complying with the respective clustering goal.

This section and the following ones will describe experiments that were conducted to demonstrate the full process, from the preprocessing phase to the cluster analysis phase. The experimental setup includes a repetition of the process for n clusters, where $2 \leq n \leq 20$. During every "run", the changes in the clustering behaviour and the performance are being assessed, with the ultimate aim to identify the optimal value for n, which can subsequently be utilized for the dynamic control of the clustering process.

The following metrics are used for the evaluation of the clustering results (with n being the number of clusters):

1. *Average Student Entropies ($SE(C_n)$)*, measuring the distribution of students in n clusters (i.e., the distribution of students' problem-solving sequences in n clusters),
2. *Average Problem Entropies ($PE(C_n)$)*, measuring the distribution of problems in n clusters (i.e., the distribution of different students' solving sequences to the problems in n clusters),
3. *Average Variance ($V(C_n)$)* in n clusters, computed by the average standard deviations for the attributes, and
4. *Average Expected Prediction Error ($EPE(C_n)$)*, averaging over n clusters' capabilities of correctly predicting success.

These metrics and their characteristics are described in more details in the following sections.

6.2.1 Entropy

Entropy, before it has been introduced in the field of information theory, has been discussed in several other areas, like, for instance, physics / thermodynamics (see, e.g. [Darrow, 1944], [Haddad et al., 2005]) or social sciences (see, e.g. [Bailey, 1990]).

Entropy is often associated with the the certainty / uncertainty (see a short discussion in [Tribus and McIrvine, 1971]) of an event, or with the order / disorder in a system ([Wright, 1970], [Lambert, 2002]), which is, however, discussed controversially (see,e.g. [Styer, 2000]). Although a universal definition of the term "entropy" still does not exist, there is common consensus regarding its implications and interpretation.

Figures 6.2(a) to 6.2(c) explain the concept of order and disorder in the area of physics / thermodynamics.

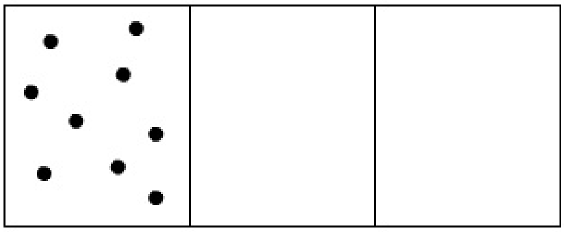
In information theory (see, for instance, [Gray, 2009] for an extensive discourse), there is a slightly different understanding of entropy, compared to physics / thermodynamics. The concept of entropy in this domain was coined by [Shannon, 1948], who defined the entropy of a discrete stochastic variable with a finite set of possible outcomes (the "alphabet") as the expected value for its information content. Thus, the Shannon entropy H (see Equation 6.1, where K is a positive constant) can also be described as a metric of uncertainty.

$$H = -K \sum_{i=1}^{n} p_i * log_a(p_i) \tag{6.1}$$

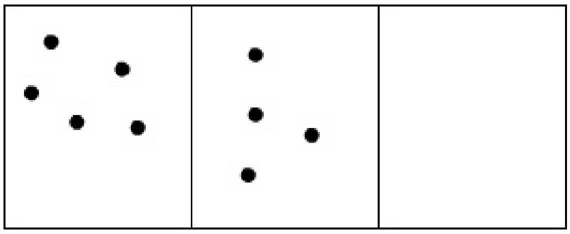
Shannon defines H as the "entropy of a set of probabilities p_1,...,p_n" and introduces a simple example with two possibilities, i.e., a two-letter alphabet (probabilities p and $q = 1 - p$ respectively), resulting in an entropy $H = -(p * log(p) + q * log(q))$ (omitting K as it is only a basis for the determination of a unit of measure, which is of no relevance for the example). In this example, the entropy would be 1 in case $p = q$, i.e., a uniform distribution of probabilities for the two possibilities. This would apply, for example, for a coin flip using an unbiased fair coin. Likewise, in case of $p = 0$ or $p = 1$, the entropy would become 0, i.e., the value for a random variable is *certain.*

Here, entropy-based indices are used to determine the distribution of problem-solving sequences related to particular objects to clusters. An "object" in this context can either be a student or a problem, thus two different entropy-based metrics are introduced, one measuring the consistency of students' problem-solving approaches (across different problems), and the other one measuring the consistency of solving approaches for specific problems (considering different students).

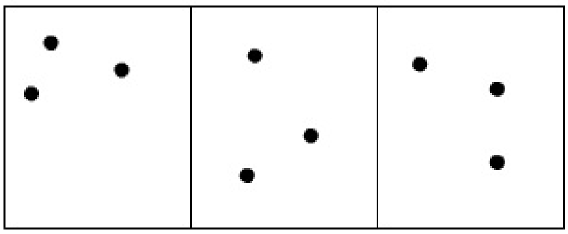
Given, for example, a student showed consistent behaviour during different problem-solving sessions, an *optimal* cluster setting would associate this student's problem-solving sequences with one or few clusters only, resulting in a low value for $SE(C_n)$. However, we cannot presume students' behaviour to be that consistent in practice.



(a) $N = \frac{9!}{9!} = 1$



(b) $N = \frac{9!}{(5!*4!)} = 126$



(c) $N = \frac{9!}{(3!*3!*3!)} = 1680$

Figure 6.2: This figure explains the concept of order and disorder using the example of 9 particles distributed to 3 places, drawn from the domain of thermodynamics. If all particles can be in one place only, the system is in order, i.e. entropy is 0. The more states are possible (i.e., places a particle could theoretically be in), the higher the entropy gets. If n is the number of particles, the number of the states can be computed as $N = \frac{n!}{(n_1!*n_2!*...n_k!)}$. 6.2(a) shows a system with 0-entropy because there is only one state the particles can be in: $N = \frac{9!}{9!} = 1$. In 6.2(b), the entropy gets higher, as the number of states increases: $N = \frac{9!}{(5!*4!)} = 126$. In 6.2(c), the number of states is computed as $N = \frac{9!}{(3!*3!*3!)} = 1680$, which results in an even higher entropy. Simplified, it can be stated that in 6.2(a), the system is in order, while in 6.2(c), it is in disorder to a certain degree.

Likewise, we can state that in an *optimal* cluster setting, the problem-solving sequences of different students that, however, show similar behaviour, should also be grouped in the same cluster.

Similar analyses can be done for different aims, e.g., at the level of problems in order to find out if the distribution of problems to clusters satisfactorily identifies or isolates classes of similarly structured problems. Therefore, likewise conditions apply to $PE(C_n)$.

The combination of students and problems can help to not only find out which approach a student shows, but also if the approach differs for different types of problems. Both entropy indices, however, tend to naturally increase as the number of clusters increases in the concrete scenario, and are, therefore, not sufficient in themselves for characterizing the results of the clustering process (i.e., a counterbalance is needed). This increase originates, in this case, from clusters often being homogeneous along one dimension but inhomogeneous along others. For instance, regarding the distribution of students to clusters, we have to consider that problem-solving behaviour consists of several components that could influence the assignment of a student's problem-solving sequence to a cluster. The more clusters are introduced, the finer gets the granularity of the analysis, and the more factors could end up being emphasized by the representation in a cluster. Thus, it is practically impossible to receive the *optimal* cluster setting mentioned before. However, a good cluster setting would be able to group a student's problem-solving sequences into the same cluster if they are similar at least along one dimension.

Equation 6.2 depicts the computation of $SE(S_x)$ for a student S_x that is based on the Shannon standard entropy measure H. The logarithm base, here and in the following formulae denoted as a, is not decisive for the results. At a later point, the values are thus normalized.

S_{c_i} is the number of a specific student's problem-solving sequences that can be found in cluster i. With $|S|$ being the overall number of this student's problem-solving sequences, $\frac{S_{c_i}}{|S|}$ is the ratio of the student's problem-solving-sequences in cluster i and this student's overall number of sequences, thus, the probability of this student's problem-solving sequence being assigned to cluster i. $SE(C_n)$ is the respective average over all students, as shown in Equation 6.3.

$$SE(S_x) = -\sum_{i=1}^{n} \frac{S_{c_i}}{|S|} * log_a(\frac{S_{c_i}}{|S|}) \tag{6.2}$$

$$SE(C_n) = \frac{\sum_{x=1}^{n} SE(S_x)}{n} \tag{6.3}$$

Similarly, Equation 6.4 shows the computation of $PE(S_x)$ for a problem P_x, where $PE(S_x)$ is again based on the standard entropy measure described before.

P_{c_i} denotes the number of solving sequences for a particular problem that can be found in cluster i. $|P|$ is again the overall number of solving sequences for this specific problem, thus, $\frac{P_{c_i}}{|P|}$ denotes the probability of a problem's solving sequence (independent of the student it was produced by) being assigned to cluster i. $PE(C_n)$ is the average over all problems, as shown in Equation 6.5.

$$PE(P_x) = -\sum_{i=1}^{n} \frac{P_{c_i}}{|P|} * log_a(\frac{P_{c_i}}{|P|}) \tag{6.4}$$

$$PE(C_n) = \frac{\sum_{x=1}^{n} PE(P_x)}{n} \tag{6.5}$$

6.2.2 Variance

As the clustering process aims at maximizing the distances between clusters and minimizing the distances within clusters, in a good cluster setting, the variance of attribute values is kept low; in an *optimal* cluster setting, $V(C_n)$ would thus tend to 0. This would also imply that similar values for attributes can be found in the same clusters.

However, similarly to the entropy-based metrics where values naturally increase with an increasing number of clusters, the variance automatically decreases with an increasing number of clusters – in a 1-cluster setting, the variance would be maximal.

Equation 6.6 depicts the computation of $V(C_n)$ for a cluster setting with n clusters with $\sigma^2(C_i)$ (see Equation 6.7) being the mean standard deviation over all m attributes in a cluster i.

$$V(C_n) = \frac{\sum_{i=1}^{n} \sigma^2(C_i)}{n} \tag{6.6}$$

$$\sigma^2(C_i) = \frac{\sum_{j=1}^{m} \sigma^2(A_j)}{m} \tag{6.7}$$

6.2.3 Prediction Capability

As a fourth metric, a cluster's prediction capability is introduced. Here, the aim is the prediction of whether a problem-specific solving sequence will lead to successful completion of the problem or not. The metric can thus be regarded an "outcome-metric" indicating whether a concrete sequence is rather "good" or "bad". This metric is, furthermore, the one most strongly related to a specific domain, compared to the previously described ones. Therefore, if a similar approach should be applied to a different domain, the general idea of clusters' prediction capabilities could still be used but the way of computing it would have to be adapted.

Towards the aim of predicting whether a sequence ultimately leads to "success", the clusters are analyzed regarding the distribution of "successful" sequences. The further a result is from equal distribution of success and failure, the more accurate are the cluster's potential predictions. The expected prediction error $EPE(C_n)$ would thus be minimized in an *optimal* cluster setting, indicating that the clusters precisely model the characteristics of sequences that lead to success or failure.

Equation 6.8 shows how $EPE(C_n)$ is computed for a cluster setting with n clusters.

$$EPE(C_n) = \frac{\sum_{i=1}^{n} err(C_i)}{n} \tag{6.8}$$

where

$$err(C_i) = \begin{cases} \frac{co(C_i)}{tot(C_i)} & if \quad \frac{co(C_i)}{tot(C_i)} \leq 0.5 \\ 1 - \frac{co(C_i)}{tot(C_i)} & otherwise \end{cases} \tag{6.9}$$

with $co(C_i)$ denoting the number of completed steps in cluster i and $tot(C_i)$ denoting the number of total steps in cluster i.

As the there are only two possible outcomes for the prediction, i.e., 0 and 1, the expected prediction error ranges between 0 and 0.5. For instance, if the result for $\frac{co(C_i)}{tot(C_i)}$ would be 0.4, the predictor would suggest 0, i.e., failure, with an expected error of 0.4. In general terms, we can state that the expected prediction error grows with growing distance of $\frac{co(C_i)}{tot(C_i)}$ to the closer ones of the boundaries 0 and 1.

6.2.4 Metric Combination

So far, the following individual cluster setting evaluation metrics were introduced: the *student entropies*, the *problem entropies*, the *variance*, and the *expected prediction error*. The entropy-based metrics aim at capturing intra-personal similarities and the effects of problem types on the behaviour of learners. Regarding student entropies, patterns are indicative of students showing stable problem-solving behaviour. Regarding problem entropies, recognized "patterns" are either indicative of problems of the same structure or of independent approaches people share.

As already mentioned, the behaviour of $V(C_n)$ and $EPE(C_n)$ is complementary to the one of the entropy-based metrics. Therefore, these four metrics are ideal for being combined in a balanced optimization formula (see Equation 6.10).

Regarding $SE(C_n)$ and $PE(C_n)$ we can observe a logarithmic ascent, whereas regarding $V(C_n)$ and $EPE(C_n)$ a gradual descent can be observed. Experimentally comparing $V(C_n)$ and $EPE(C_n)$, $V(C_n)$ shows more fluctuations than $EPE(C_n)$, and for $EPE(C_n)$, the descent is more significant than for $V(C_n)$.

The four indices are then combined in order to identify the optimal cluster setting, i.e., number of clusters. Before feeding the respective resulting values into a combined optimization formula, the following two intermediate steps are performed:

1. the resulting values for the indices are normalized to a range between 0 and 1 (including the boundaries), and

2. weights are introduced for the different indices.

The first step is beneficial because by normalization the dependency on irrelevant factors (here, the logarithm base) can be repealed. The second step allows later optimization of clustering towards a specific aim (like, for example, the minimization of error).

Figures 6.3, 6.4 and 6.5 show the normalized graphs for an example data configuration (data sets SET_MARKOV and SET_BOTH). They are to be understood as follows: clustering was performed on the respective data sets several times with different cluster configurations (using $2 \leq n \leq 20$ clusters). For every configuration, the values for the four metrics used for the indication of quality were computed and subsequently normalized to a range between 0 and 1. The results are depicted by the different graphs in each of the figures.

For the process described in this thesis, we define the main optimization goal as follows: in a good cluster setting, the spread between the ascending ($SE(C_n)$ and $PE(C_n)$) and the descending ($V(C_n)$ and $EPE(C_n)$) graphs should not exceed a certain threshold (i.e., the convergence point of the different graphs, considering the normalized values). This optimization goal can be defined by combining the following aspects of a good cluster setting:

- As we expect a student to apply at least a certain basic problem-solving approach globally, *student entropies* should not exceed a certain threshold, because this would indicate that the definitions of problem-solving styles would be distorted by too many details.

- As problems can be of different nature and we expect the problem-specific characteristics to, at least marginally, influence the problem-solving behaviour, *problem entropies* should not exceed a certain threshold.

- The *variance* in general should be kept low (as a good cluster setting is indicated by minimal distances within clusters in combination with maximal distances between the clusters). However, as the number of clusters increases, the variance naturally becomes lower because more clusters lead to a smaller number of instances within the cluster. In the extreme case, a cluster would hold only one data instance, leading to zero variance, which is in this case, however, not desirable, because we cannot infer, from such a setting, any information about the data. Thus, the variance should not deceed a certain threshold.

- The *expected prediction error* can be assessed in a way similar to what was described for variance. Generally, a low value for the expected prediction error is good. However, if the number of clusters becomes too high, thus minimizing the number of data instances in a cluster and thus also the expected prediction error, the resulting clusters are not informative in any way any more. Thus, again, the value for the metric should not deceed a certain threshold.

Equation 6.10 shows the computation of the optimized value for a configuration with n clusters.

$$Opt(n) = \left| \frac{\frac{no(SE(C_n))*w_s+no(PE(C_n))*w_p}{2} - \frac{no(V(C_n))*w_v+no(EPE(C_n))*w_e}{2}}{w_s + w_p + w_v + w_e} \right| \tag{6.10}$$

where $no(N)$ normalizes the values in N to a range between 0 and 1 (again including the boundaries). Normalization is done by assigning the lowest available value the new value 0, changing the highest available value to 1, and updating the values in between accordingly.

The best result for $Opt(n)$ is then the value closest to 0. For the example used in figure 6.4(a), the unweighted optimization process would identify 5 as the best number of clusters, as can also be read from Table 6.1 and Figure 6.6.

Table 6.1: This table compares the optimization results for different data sets (based on Data Set II, as described in Section 3.2.2) with different weight configurations for the Andes interaction data in the physics course of the year 2008 (using the extended help state setting). The numbers in the data sets' rows show the optimum number of clusters found for the respective data set and weight configuration.

Weights **(*s*, *p*, *v*, *e*)**	***Markov***	***Statistical***	***Both***
1 | 1 | 1 | 1	5	6	7
1 | 1 | 1 | 3	8	6	9
1 | 1 | 3 | 1	7	7	11
3 | 1 | 1 | 1	4	4	4
1 | 3 | 1 | 1	4	3	4
1 | 1 | 2 | 2	8	9	10
1 | 2 | 1 | 2	5	5	6
1 | 2 | 2 | 1	5	5	7
2 | 2 | 1 | 1	4	4	4
2 | 1 | 1 | 2	5	6	6
2 | 1 | 2 | 1	5	5	7

The results clearly show that the optimum number of clusters changes with a changing clustering purpose (indicated by changing weights). The first row contains the results of the process using uniformly distributed weights, thus not optimizing for a specific aim.

Generally, the optimum number of clusters ranges between 5 and 7. The configurations depicted in rows 2 to 5 focus on one particular criterion and thus use (equally) low priorities for the other criteria.

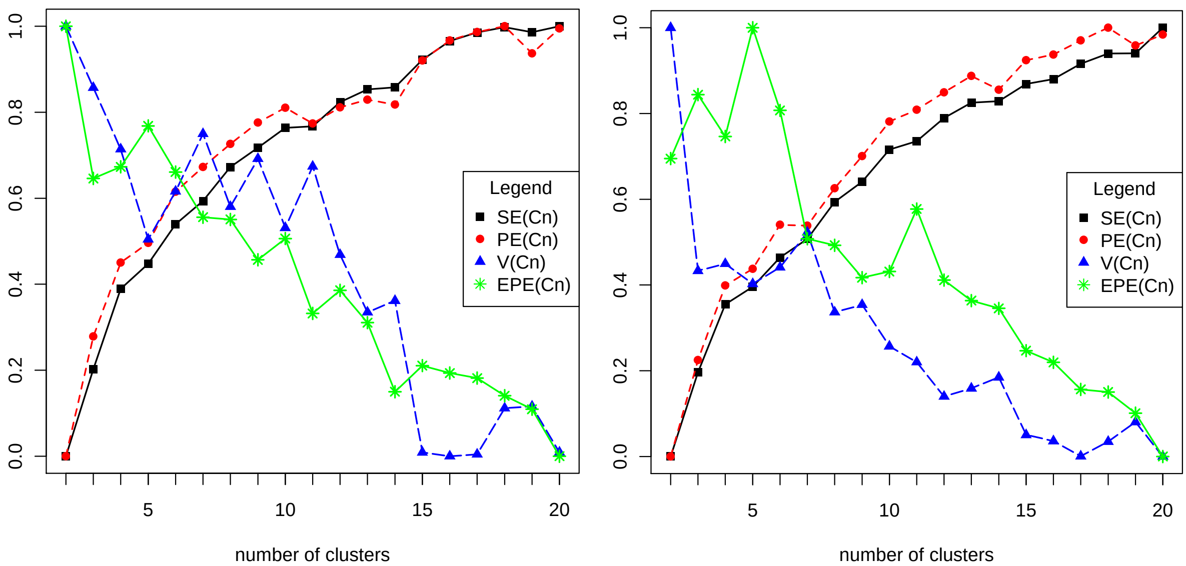
Comparing the different criteria, we can declare that, if weighed high, $SE(C_n)$ and $PE(C_n)$ both suggest a lower number of clusters whereas $V(C_n)$ and $EPE(C_n)$ suggest a higher number of clusters in this case. Thus, we can draw the conclusion that if, for instance, $EPE(C_n)$ should be minimized, a number of clusters $n > default$ should be chosen.

The remaining rows in Table 6.1 show the behaviour of the optimum number of clusters n when two criteria should be combined as a basis for optimization. Figure 6.6 illustrates the changing optimum number of clusters for the first 5 weight configurations listed in the table.

The optimization approach as a whole and the individual metrics can be evaluated best by assessing the results of the experiments and the resulting clusters themselves. As the proposed approach relies on human expertise at a late stage of the process (see Section 6.4), it is not possible to quantitatively prove its validity. However, the clustering results (and the suggested optimized setting in specific) provide evidence for the existence of patterns in the behaviour of learners that are well interpretable and assessable by humans.

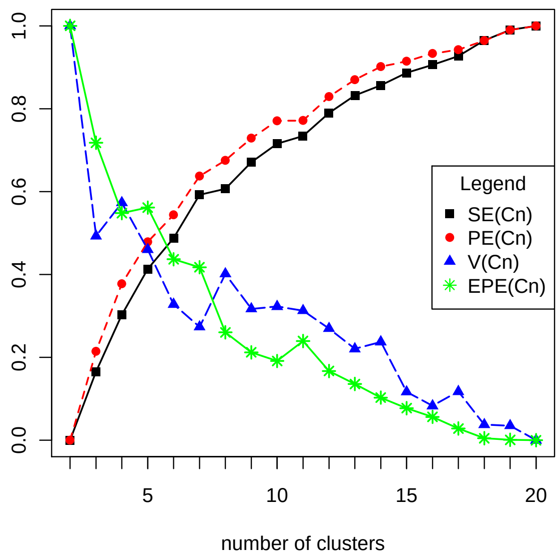
An evaluation must thus concentrate on the clusters and their descriptiveness rather than on the process leading to these clusters. A quantitative evaluation could theoretically be done regarding the pattern detection capabilities of the approach by manually assessing the problem-solving data, assigning them to the patterns they depict, and run the process in order to see whether the patterns are detected correctly.

However, the proposed approach does not concentrate on the detection of predefined patterns only, but should be able to, after a clustering-based analysis, suggest new, potentially meaningful patterns to a human expert who then judges them. The task of assessing whether a pattern exhibited in a problem-solving process is meaningful or not, cannot be fully automated in this case, as it relies on profound knowledge in the area of (learning) psychology and an understanding of human behaviour in general that cannot be attained algorithmically.

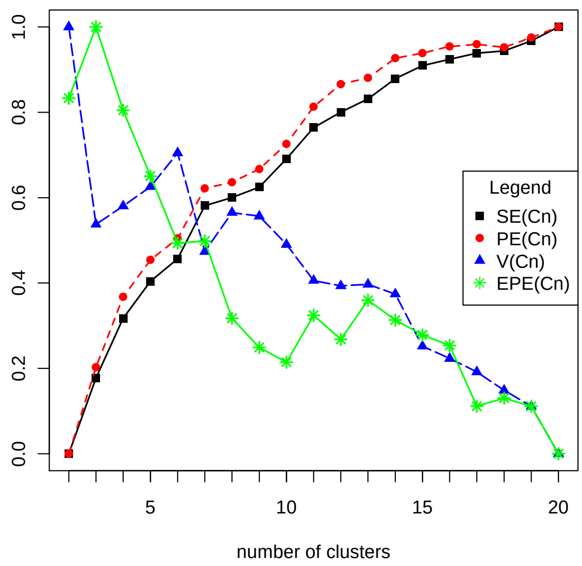


(a) *SET_MARKOV* (b) *SET_BOTH*

Figure 6.3: This figure shows the normalized graphs for the data sets *SET_MARKOV* (a) and *SET_BOTH* (b) in the extended hint processing on data of the year 2007.

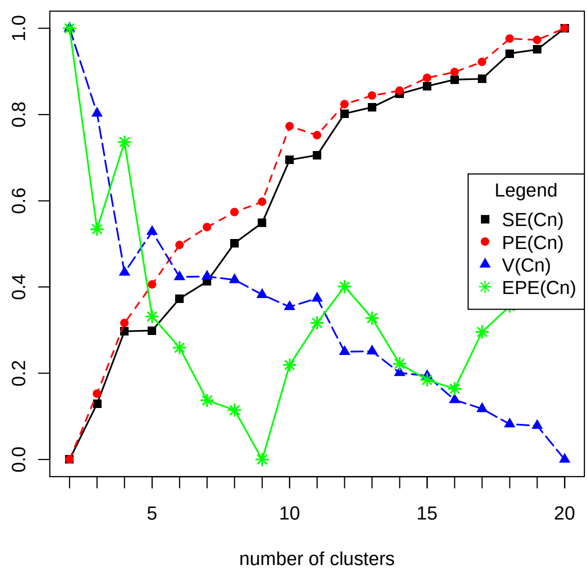


(a) *SET_MARKOV*

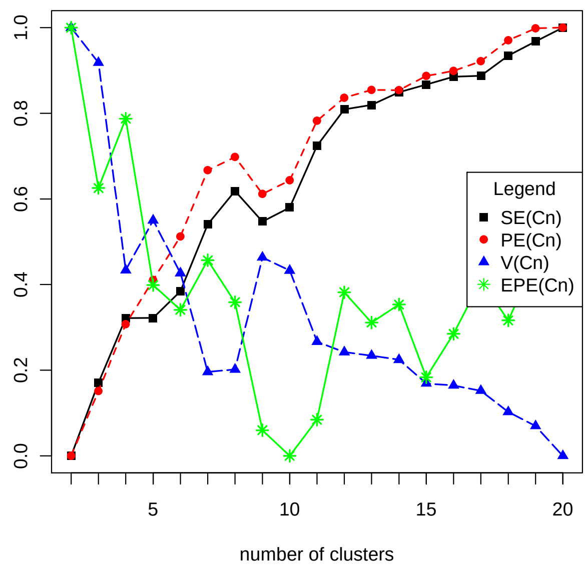


(b) *SET_BOTH*

Figure 6.4: This figure shows the normalized graphs for the data sets *SET_MARKOV* (a) and *SET_BOTH* (b) in the extended hint processing on data of the year 2008.

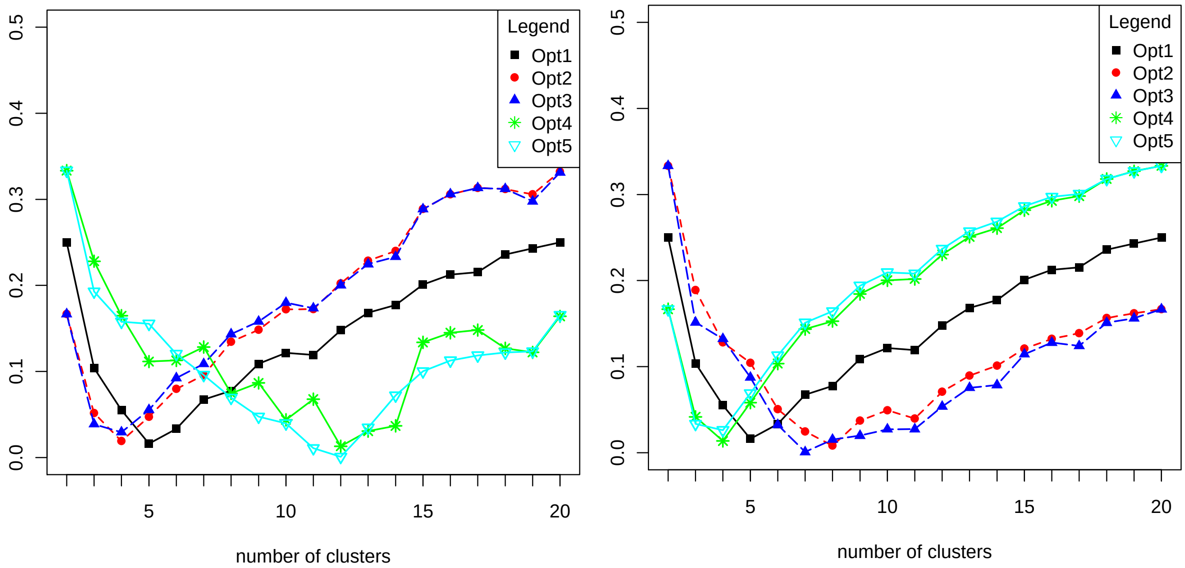


(a) *SET_MARKOV*



(b) *SET_MARKOV*

Figure 6.5: This figure shows the normalized graphs for the data sets *SET_MARKOV* (a) and *SET_BOTH* (b) in the extended hint processing on data of the year 2009.



(a) 2007 (b) 2008

Figure 6.6: This figure shows the optimization results for the first 5 weight configurations listed in table 6.1 for the data set *SET_MARKOV* of the years 2007 (a) and 2008 (b).

6.3 Comparison of Different Data Sets and Settings

In order to gain results that reliably indicate possible trends and characteristics in user behaviour, the clustering process was conducted with different data sets (all based on Data Set II introduced in Section 3.2.2) and configurations (i.e., aggregated or extended help processing, also see Section 3.2.2), considering also different academic terms. This practice also opened the opportunity to compare the results of the clustering process for different students solving the same problems, and to determine whether the identified trends are of a rather global nature or, to the contrary, dependent on a specific setting.

Towards these ends, as a first step, the clustering results for each of the four metrics ($SE(C_n)$, $PE(C_n)$, $V(C_n)$ and $EPE(C_n)$), were compared for different academic terms, settings and data sets. For instance, the results for the particular data set *SET_MARKOV* (see Section 5.3.2, with one of the two possible help processing configurations) could be compared for the three different academic terms (Spring 2007, 2008 and 2009). The results of this step are reported in Figures 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, and 6.14.

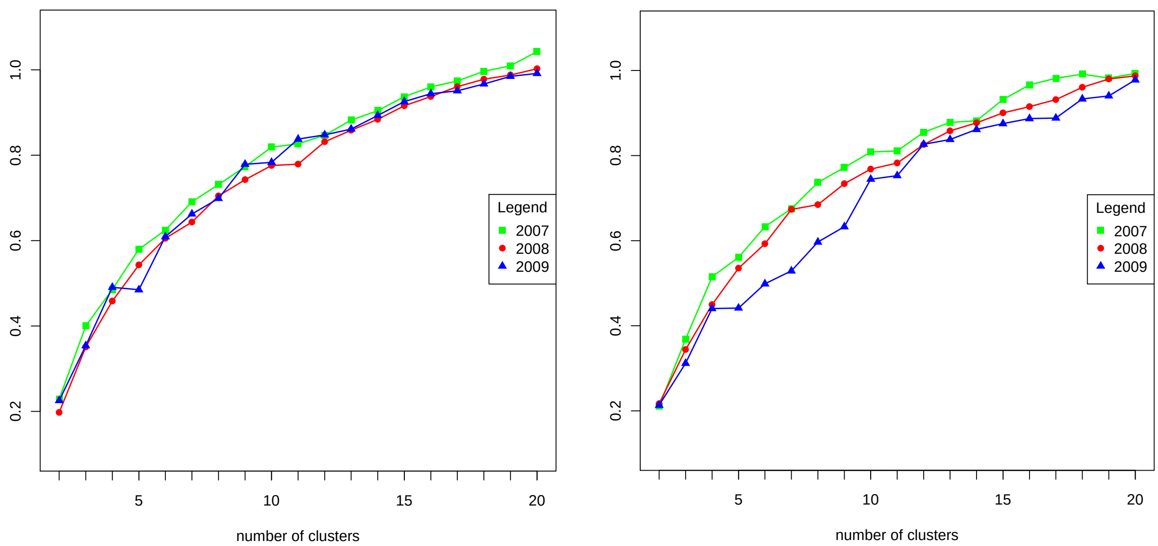
If a sufficiently high amount of activity sequences is available, it can be assumed that the different implementations of the (clustering) process would lead to similar results. Thus, the results and comparisons just described, generally aim at the verification of the overall process.

Regarding the data that was available for the experiments, we can declare that

- the number of activities in the physics spring course of 2007 is relatively high,
- the number of activities in the physics spring course of 2008 is relatively high,
- the number of activities in the physics spring course of 2007 and 2008 is about equal, and
- the number of activities (and students, in general,) in the physics spring course of 2009 is significantly lower, compared to 2007 and 2008.

Based on these general observations it can be assumed that the clustering results for the years 2007 and 2008 will be similar, whereas the results for the year 2009 may differ (and may in general be less reliable).

A comparison of, for example, the results reported in Figures 6.7, 6.8, 6.13, and 6.14 supports these assumptions.

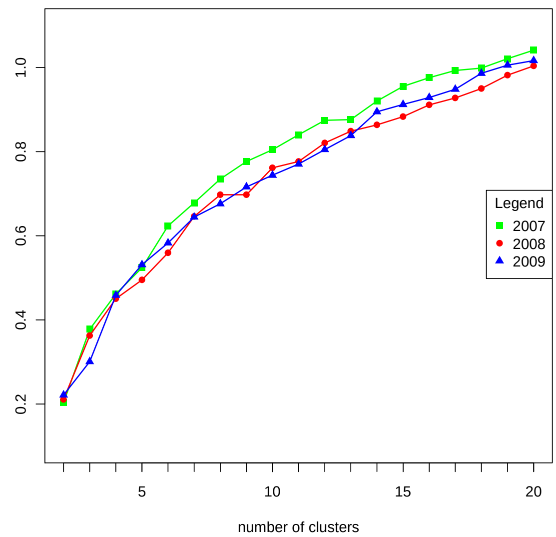


(a) Aggregated help states

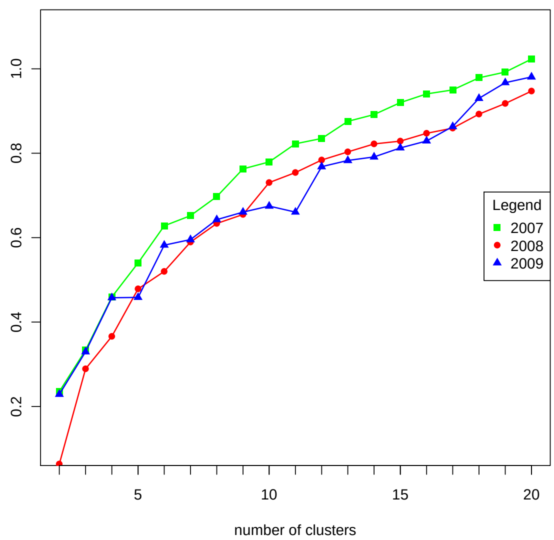
(b) Extended help states

Figure 6.7: This figure shows the $SE(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.

As shown in Figures 6.7 to 6.14, initial experiments were run on both help state settings ("aggregated" and "extended") in order to determine which of them provides more expressive results for the respective purpose. Additionally, tailoring the data set itself to the purpose is recommendable. For instance, if the clustering purpose would be to specifically analyze users' preferences regarding different types of help, the "extended" help state setting would be the better choice, and the data set should contain only features that are relevant for this purpose (see also Section 6.4).

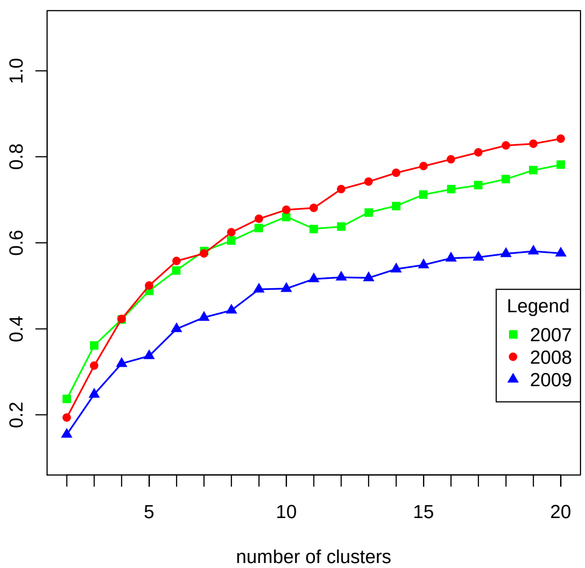


(a) Aggregated help states

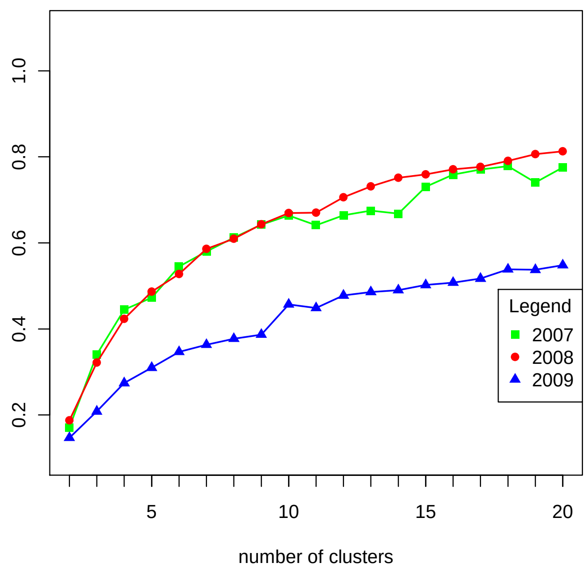


(b) Extended help states

Figure 6.8: This figure shows the $SE(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.

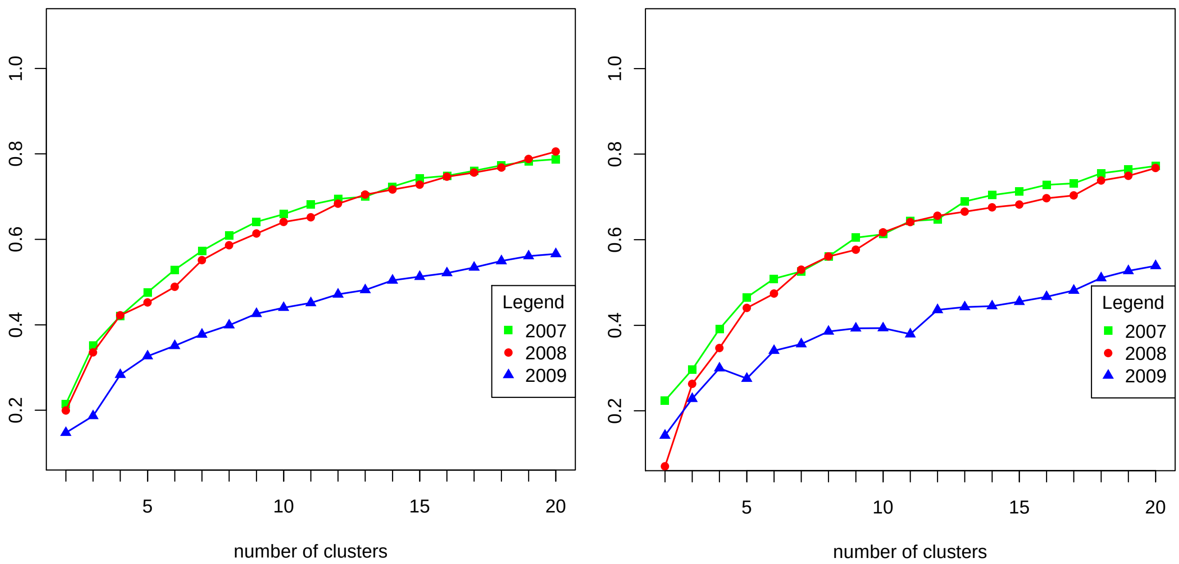


(a) Aggregated help states



(b) Extended help states

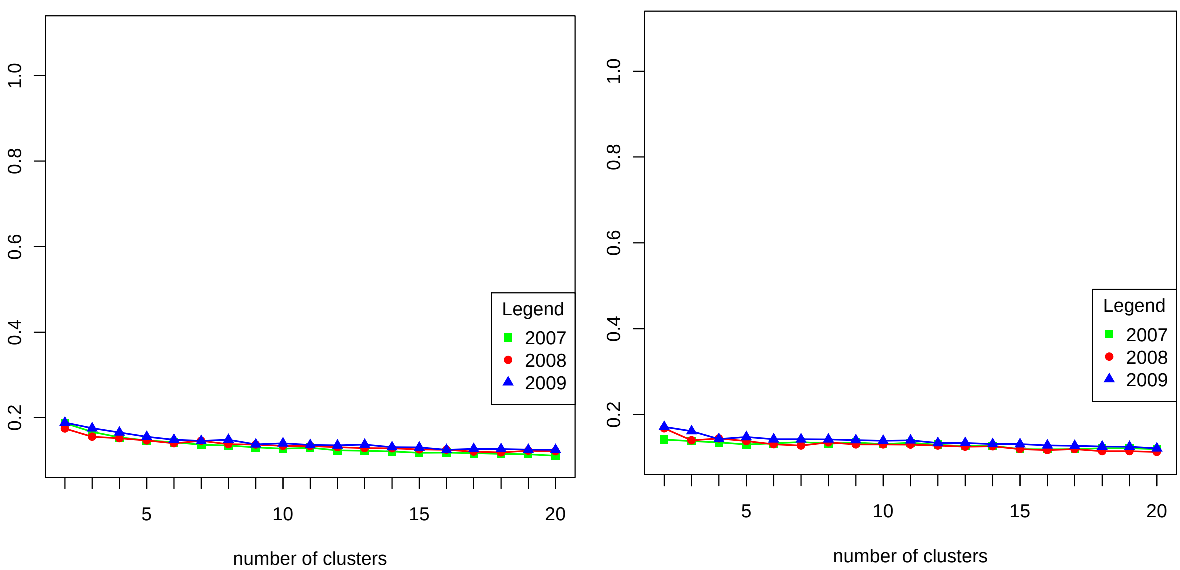
Figure 6.9: This figure shows the $PE(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.



(a) Aggregated help states

(b) Extended help states

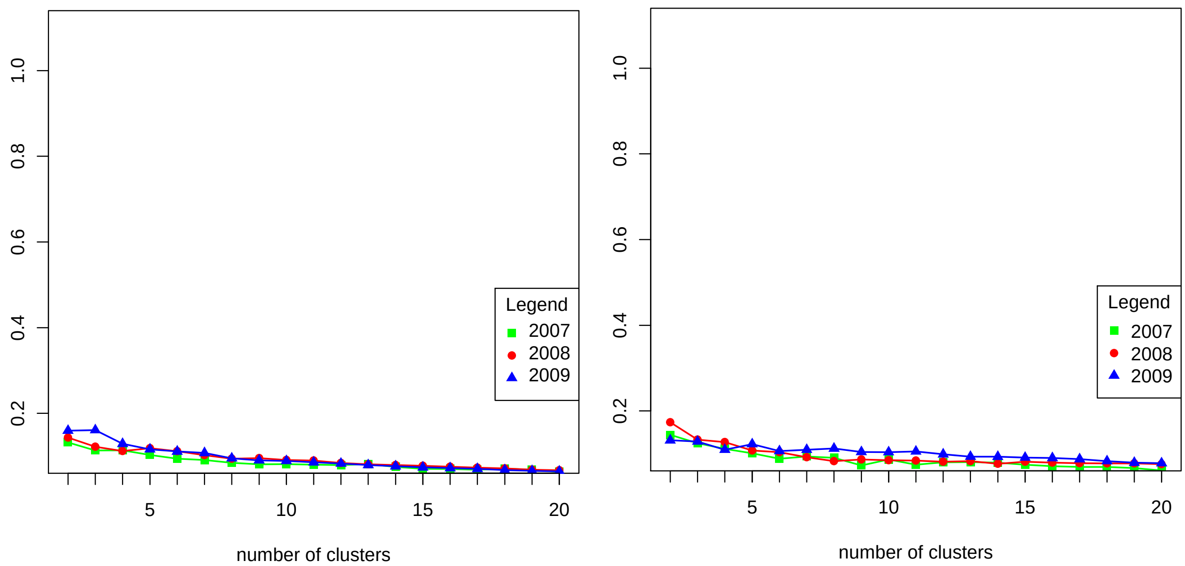
Figure 6.10: This figure shows the $PE(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.



(a) Aggregated help states

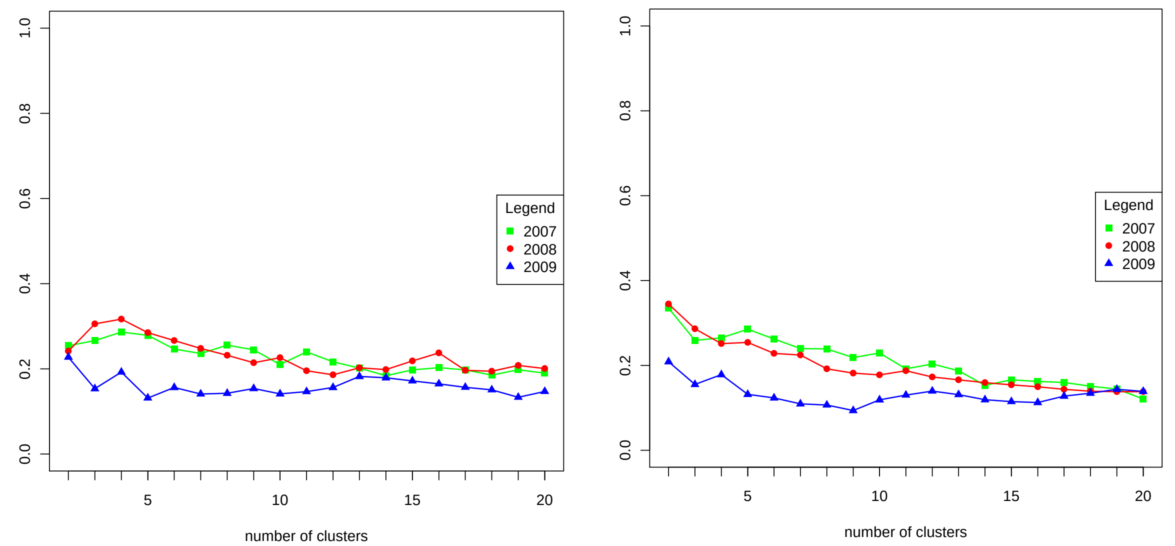
(b) Extended help states

Figure 6.11: This figure shows the $V(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.



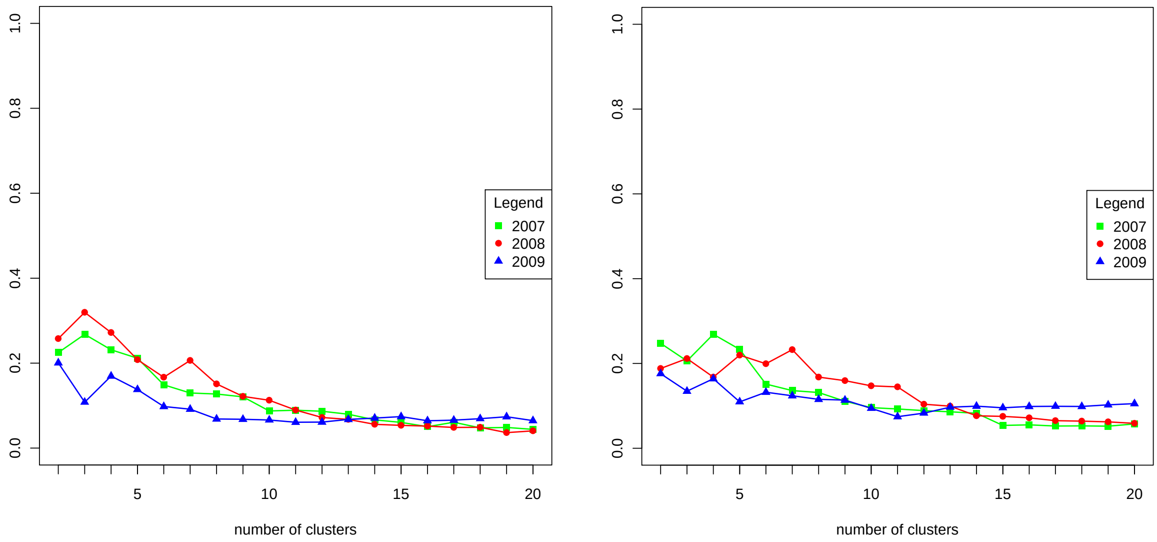
(a) Aggregated help states (b) Extended help states

Figure 6.12: This figure shows the $V(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.



(a) Aggregated help states (b) Extended help states

Figure 6.13: This figure shows the $EPE(C_n)$ results for the data set SET_MARKOV in aggregated (a) and extended (b) help processing settings for all three academic terms.



(a) Aggregated help states

(b) Extended help states

Figure 6.14: This figure shows the $EPE(C_n)$ results for the data set $SET\_STATISICAL$ in aggregated (a) and extended (b) help processing settings for all three academic terms.

Figures 6.15 and 6.16 show the trends for $PE(C_n)$ and $V(C_n)$, $SE(C_n)$ and $EPE(C_n)$ respectively, using a data set that only contains help-related features.

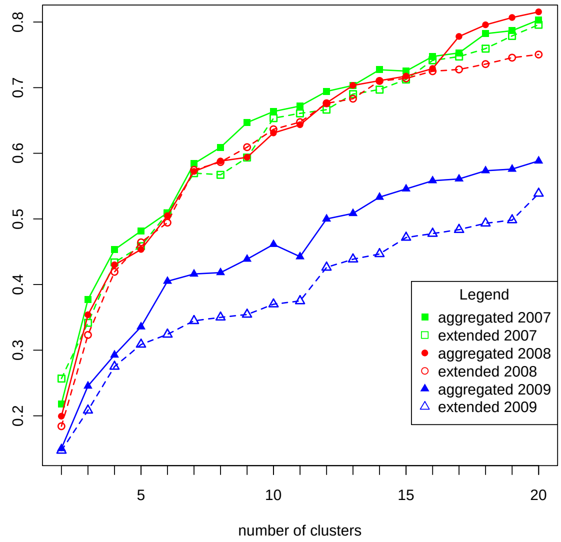
For example, Figure 6.15(a) indicates that the results for the aggregated and extended help processing configurations are rather similar in the cases of the 2007 and 2008, whereas for 2009, the extended help processing configuration provides the better results. In Figure 6.15(b), we can generally observe better results for the aggregated help processing configuration.

As already mentioned before, the data from 2007 and 2008 can be considered more reliable, compared to the data from 2009. As the results for both years, 2007 and 2008, were either relatively equal for the "extended" and the "aggregated" help state configuration, or better for the aggregated one, it was decided to further use this setting, whenever both would theoretically be applicable. The further results reported in this chapter are thus based on the "aggregated" help state setting.

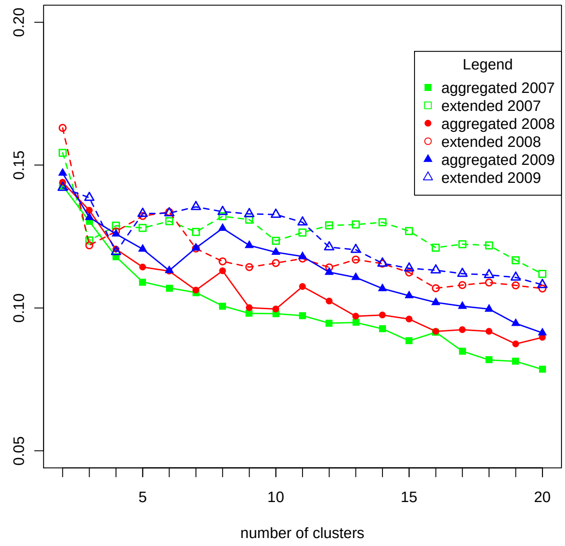
Regarding the different data sets, different trends could be observed:

1. the best results for predicting success (indicated by a low value for $EPE(C_n)$) are provided by the analysis of the data set $SET\_STATISTICAL$, but

2. the best results for detecting patterns are provided by the analysis of the data set SET_MARKOV. The quality of a data set for pattern detection cannot be determined directly by only analyzing the values for the four metrics but requires an observer to examine the resulting clusters in order to identify the requested patterns the process

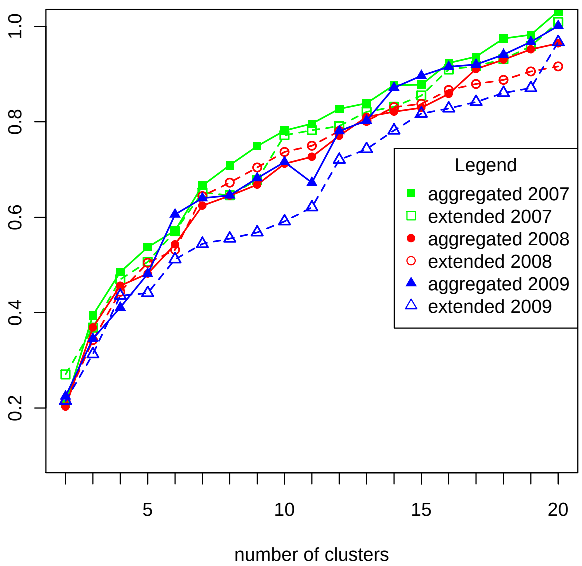


(a) $PE(C_n)$

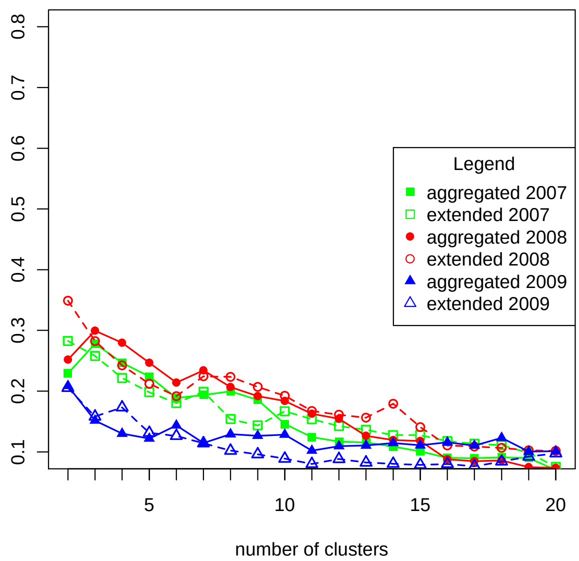


(b) $V(C_n)$

Figure 6.15: This figure shows the $PE(C_n)$ (a) and $V(C_n)$ (b) clustering results for a data set containing help-related features in aggregated and extended help processing settings for all three academic terms.



(a) $SE(C_n)$



(b) $EPE(C_n)$

Figure 6.16: This figure shows the $SE(C_n)$ (a) and $EPE(C_n)$ (b) clustering results for a data set containing help-related features in aggregated and extended help processing settings for all three academic terms.

clustered for. This is done by a human observer here but could partly be automated in the future.

As different data sets proved to be well applicable for different purposes, a combined data set was introduced to be used as a basis for the further process including the automatic creation of specifically tailored new data sets.

6.4 Three-Level Clustering and Cluster Analysis

This section concentrates on the clustering phase of the overall process as depicted in Figure 6.1 and discusses three levels of clustering, each tailored to a specific aim. Again, the experiments described here, were run on data of the nature of Data Set II (see Section 3.2.2), based on the Andes physics course of spring 2008. The results presented in this section thus correspond to this data set also. Subsequently, the experiments were repeated with the data of 2007 and 2009 in order to confirm the results.

6.4.1 Level I (Pattern-Driven)

On the first level of clustering, descriptions of predefined patterns are created before the clustering process starts, in order to identify these patterns in students' problem-solving activity sequences. Here, the well-established problem-solving style *Trial and Error* [Jarvis, 2005], [Thorndike, 1903] (also referred to as *Trial and Success*) was chosen for the demonstration of level I clustering. Learners exhibiting the *Trial and Error* style usually tend to guess the correct answer in the beginning and, by learning from mistakes, later approach the correct answer systematically by excluding the incorrect answers already tried.

Unfortunately, it is not easy to find a recent psychological description of this problem-solving style; however, the recent literature suggests that there is a common understanding on its definition (see different references by, for example, [Kanninen, 2008], [Brown et al., 2007], [Butler and Pinto-Zipp, 2006], [Cassidy, 2004], [Kolb, 1984], [Schaller et al., 2009], [Liu and Dean, 1999], [Dewar and Whittington, 2000], [Terrell, 2005], [Ballone and Czerniak, 2001], [Felder and Silverman, 1988], [Simon, 2000], or [Richmond and Cummings, 2005]).

As depicted in Figure 6.1, level I clustering involves, after the definition of the style to be clustered for, a feature selection step. Towards this end, the available attributes of the overall data set combining SET_MARKOV, $SET\_STATISTICAL$, and SET_BOTH were evaluated regarding their relevance to the *Trial and Error* problem-solving style.

Generally, a person exhibiting this kind of behaviour, can be expected

- to have high prior probabilities for incorrect attempts,
- to have low-to-medium prior probabilities for correct attempts,

- to have a low help-request rate,
- to have low transition probabilities from an incorrect attempt to a help-request, and
- to generally have a high rate of incorrect attempts.

The reason for not expecting about equal probabilities for correct and incorrect guesses is rooted in the nature of the data; as the ITS in this case usually provides several possible answers or leaves them open entirely, there is just one correct option as opposed to several possible incorrect answers.

Based on these considerations, features were evaluated and a set of features that was expected to be relevant for the identification of the *Trial and Error* problem-solving style, was compiled. The resulting new data set SET_TRIAL_ERROR was then fed into the clustering process.

Table 6.2 contains the clustering results for this data set, based on an 8 cluster configuration. The number of 8 clusters was chosen because the unweighted optimization configuration (i.e., using equal weights of 1 for all metrics) has shown that the optimum number of clusters is below 8 for all three data sets that were tested (see Table 6.1). Later, the hypothesis of the optimum number of clusters for the new SET_TRIAL_ERROR data $n \leq 8$, was confirmed by further experiments (an optimum of 6 was found in this case).

The results indicate that the style clustered for could be identified in clusters 1 and 6. Both show characteristics expected for *Trial and Error* problem-solvers: a high prior probability for incorrect attempts, a low rate of help-requests, a prior probability for help-requests of ~ 0, and in general, a high percentage of incorrect attempts.

Level I clustering as just described can be repeated with any other problem-solving style that can be sufficiently well defined to select the relevant features. At this point, given detailed definitions of patterns to be identified in a set of activity sequences, the application of a supervised learning instead of a clustering approach would also be possible, presumed a certain number of preclassified training examples would be available. However, this approach would not only involve more human intervention, but, more importantly, could not scale to the following two levels of discovery.

6.4.2 Level II (Dimension-Driven)

This level goes a step further, compared to the first one, and does not only cluster for predefined concrete styles, but aims at recognizing different patterns (i.e., styles) within predefined "dimensions" of user behaviour. It is thus now the task of the clustering unit to cluster for a specific dimension, i.e. a more general kind of behaviour, that may again entail different concrete problem-solving styles.

Again, an example was chosen for demonstration purposes, in this case, *Help-Seeking* behaviour [Nelson-Le Gall, 1985], [Aleven et al., 2003] as it is a very well known learning dimension.

Table 6.2: This table shows the clustering results on *SET_TRIAL_ERROR* with an 8 cluster configuration. The numbers in parentheses after the cluster id show the number of problem-solving instances in the respective cluster.

Attribute	C_0(1506)	C_1(456)	C_2(2503)	C_3(489)	C_4(1770)	C_5(1606)	C_6(464)	C_7(671)
PRIOR_PROB_C	0.9923	**0.1219**	0.239	0.4187	0.7405	0.6201	**0.2347**	0.3079
PRIOR_PROB_I	0	**0.7896**	0.1408	0.2045	0.2324	0.304	**0.7422**	0.0045
TRANS_PROB_I_I	0.1429	**0.5568**	0.2377	0.0772	0.0348	0.5452	**0.0991**	0.1429
TRANS_PROB_I_H1	0.1429	**0**	0	0	0	0	**0**	0.1429
TRANS_PROB_I_H2	0.1429	**0.0082**	0.0951	0.0336	0.0251	0.0301	**0.0087**	0.1429
TRANS_PROB_I_H3	0.1429	**0.032**	0.0767	0.6779	0.0262	0.052	**0.0231**	0.1429
TRANS_PROB_I_H4	0.1429	**0.0044**	0.0399	0.0124	0.0163	0.011	**0.0056**	0.1429
PERC_HELP_STEP	0.0146	**0.0918**	0.658	0.5533	0.0652	0.1432	**0.0465**	0.728
PERC_INCORRECT	0	**0.6345**	0.1081	0.1257	0.1989	0.3877	**0.4659**	0

Help-Seeking behaviour has in the past been defined in different ways, for example, as described by the following statement: „.. many scholars consider that the ability to utilize adults and peers appropriately as resources to cope with difficulties encountered in learning situations..“ [Nelson-Le Gall, 1985], which is based on, among others, [Anderson and Messick, 1974] or [Nelson-Le Gall, 1981].

[Aleven et al., 2003] discuss a framework to understand *help-seeking* which was first presented by [Nelson-Le Gall, 1981] and later adapted by [Newman, 1994] and [Ryan et al., 2001], containing an analysis of the following tasks involved in the *help-seeking* process:

1. Become aware of need of help.
2. Decide to seek help.
3. Identify potential helper(s).
4. Use strategies to elicit help.

This sequence of tasks is generally well transferable to other scenarios, including the one discussed here.

Similar to what was described for level I, as a first step in the *clustering* phase, the behaviour elements expected to be defining for help-seeking, were identified:

- the general rate of help-requests,
- the transition probabilities from an incorrect attempt to a help request,
- the prior probability for help requests, and
- the help-internal transition probabilities (i.e., the probabilities for transitions from one help request to another one).

After the subsequent identification of the relevant features (see the selection in Table 6.3), a new data set $SET\_\ HELP\_SEEKING$ containing only these attributes was created and fed into the clustering process. As for level I, the process successfully discovered the kind of behaviour (i.e., "dimension") it clustered for. In this case, several variations of *help-seeking* behaviour could be detected that were subsequently analyzed towards the aim of identifying concrete styles. Table 6.3 shows the results and provides a clear picture of the four concrete *help-seeking* styles that could be recognized.

Problem-solvers of type H_1 show *help-seeking* behaviour comparable to the one of the *Trial and Error* type as discussed before for level I. These learners furthermore tend to request hints in sequences.

Problem-solvers of type H_2 show quite different behaviour: they tend to make sure not to submit incorrect answers. However, the results suggest that this is mostly achieved by requesting a huge amount of help, in many cases before having tried to submit an answer at all. Thus, it must be considered that these problem-solvers might also replace appropriate preparation by use of the help functionality.

Problem-solvers of type H_3 show a tendency towards the use of help in sequences (i.e., if help is requested, the probability of the next activity being another help-request, is high). Generally, these problem-solvers do not request help very often, and not right in the beginning, i.e., they usually try to solve a problem by themselves first before they ask for help. These facts suggest the assumption that H_3 problem-solvers are highly interested in understanding a problem before continuing.

Problem-solvers of type H_4 behave in a way similar to H_2 learners. Thus, if a cluster configuration with fewer clusters would have been used, the types H_2 and H_4 would most probably have been combined. Generally, the number of clusters chosen for a cluster setting, is dependent on the aspired level of granularity regarding the clustering results. In this case, the aim was not to identify rough types that learners could be assigned to, but to distinguish between subtypes belonging to the same category, thus a cluster configuration with more clusters was the better choice.

[Aleven et al., 2006] discuss a *help-seeking* model comparable to the results presented so far that involves a taxonomy of "help-seeking bugs" in student behaviour, listing the following variations:

- Help Abuse,
- Help Avoidance,
- Try-Step Abuse, and
- Miscellaneous Bugs.

Table 6.3: This table shows four problem-solving styles in the *help-seeking* dimension discovered by the clustering process (using an 8 cluster configuration). The remaining clusters not shown here contain non-*help-seeking* behaviour. The syntax is to be read as follows: the percentage results have been abstracted to the five categories *very low, low, medium, high, very high*, which are represented by the more easy to read identifiers $--$, $-$, o, $+$, $++$.

Style	**Size**	PRIOR_I	PRIOR_H*	TRANS_I_H*	TRANS_H*_H*	PERC_I	PERC_H*
H_1	- -	o	o	o	+	o	o
H_2	+	- -	+	-	++	- -	+
H_3	o	-	-	-	+	-	o
H_4	o	- -	+	-	++	- -	+

Students exhibiting *Help Abuse* generally tend to request a disproportionate amount of help, even in cases where they would theoretically be sufficiently skilled to solve a problem without help. The H_2 and H_4 types discussed before show similar behaviour regarding some aspects. For instance, a H_2/H_4 problem-solver might also tend to clicking through hints instead of spending enough time to understand the problem.

However, the process described here is able to distinguish between help-requests that take place before the submission of an answer and help-requests that follow (probably incorrect) attempts. The first is in many cases undesirable behaviour that can also be compared to the *gaming the system*-problem discussed by [Baker et al., 2006]. Actually, [Muldner et al., 2011], who aim at detecting abuse in the same ITS as used here, explicitly regard evidence for such behaviour as indication for “gaming”.

Another parallel can be drawn between the H_1 or *Trial and Error* style and *Try-Step Abuse* behaviour, as all tend to submitting answers too early, often before being sufficiently skilled.

Furthermore, we can compare the behaviour of a H_3 problem-solver with *Help Avoidance* behaviour as described by Aleven et al. In both cases, students tend to keep the amount of requested help low. As *Help Avoidance* behaviour concentrates on not using help when dealing with unfamiliar steps, it could also be considered a subtype of *Try-Step Abuse* behaviour.

However, although *Help Avoidance* and *Try-Step Abuse* seem to be related and H_3 and *Help Avoidance*, H_1 and *Try-Step Abuse* respectively, show significant similarities, we can clearly distinguish between H_1 and H_3. The latter type could also be described as learners’ endeavour not to submit incorrect answers.

Thus, we can conclude that not only did level II clustering confirm the taxonomy of “help-seeking bugs” of Aleven et al., it also added some distinct aspects to it.

6.4.3 Level III (Open Discovery)

Clustering on level III again goes one step further, compared to level II: it no longer aims at the identification of concrete styles within a predefined dimension. Instead, the process autonomously clusters for (new) dimensions in order to subsequently identify concrete styles within these dimensions (in this case, again, in the scope of problem-solving).

In contrast to the previously described levels of clustering, this level is driven by the system and does no longer rely on definitions of either concrete styles of dimensions that have to be passed on to the clustering phase. However, the process still involves human intervention for the analysis and interpretation of the results at the end of the process.

We can summarize the system's tasks on this level as follows:

- automatic selection of feature combinations with potentially high discriminatory capacity,
- creation of a new data set for each of one feature combinations,
- clustering on each of the new data sets, and
- analysis of the resulting clusters regarding significant trends, aiming at the autonomous detection of problem-solving styles.

Thus, level III clustering involves an additional feature selection unit.

Automated Feature Selection and Combination

Feature selection is based on an initial data set containing all available attributes. The experiments presented here used the data set SET_BOTH with data of the year 2008, containing 20 features. As learned from the analysis of the previous levels of clustering, dimensions as well as concrete styles can be defined by a significantly lower number of features.

The feature selection unit first randomly selects subsets of the initial feature set in order to create a new data set for each of the resulting subsets later. The random process at this stage considers all possible combinations of features, thus resulting in a number of $\sum_{k=x}^{y} \binom{n}{k}$, i.e., $\sum_{k=x}^{y} \frac{n!}{k!(n-k)!}$ data sets, with n being the number of available features and x and y denoting the lower and upper limits for the number of features in a data set. The limits are introduced due to the exponential complexity of the feature combination task and the resulting very high number of possible feature combinations (i.e., the power set, containing 2^n subsets) and due to the fact that the previous levels showed that concrete problem-solving styles as well as dimensions can be well described by a much lower number of features.

As the previous experiments have indicated, the number of features defining a concrete style or dimension, is usually rather low (for instance, the *Help-Seeking* dimension used 6 features not all of which were significant, as depicted in Table 6.3), the level III clustering experiments used the following configuration for the lower and upper limits: $x = 1$, $y = 7$. These limits however, are tailored to the requirements of the concrete scenario and must most probably be modified for different ones.

Creation of New Data Sets

As a next step, based on the original data set (here, SET_BOTH), new data sets are created for all feature combinations found before, each containing only the selected features. The resulting data sets thus illustrate different characteristics of the same basic activities produced by the learners when interacting with the ITS. In general, we have to consider that the amount of data can become very high, depending on the selected upper limit of features, and provide sufficient storage facilities and computation capacity. Another approach to deal with the high amount of data would be to only temporarily create the data sets, and sequentially analyze them.

Clustering on the New Data Sets and Cluster Analysis

After having selected / combined the features and stored them in new data sets, these sets are passed on to the clustering unit that compares the results in order to provide a ranking at the end of the process. The ranking is based on an average cluster quality metric $Q(FS_i) = \frac{D_b * w_b}{D_w * w_w}$ for a feature set FS_i, where D_b is the average distance between the cluster centroids, D_w is the average distance between the elements within a cluster, averaged again over the clusters, w_b and w_w are weights (with $0 < w_* \leq 1$). For the experiments reported here, all weights were equally initialized with $w_* = 1$.

The quality metric is based on Linear Discriminant Analysis (LDA) (see [Martínez and Kak, 2001]), which generally aims at maximizing the distance between clusters (i.e., their centroids) and at the same time minimizing the average distance between the items within the clusters. Here, the Euclidean Distance (see [Black, 2004]) was used for measuring distances.

LDA is appropriate for the requirements of the setting in this case because it preserves the original features and thus also information about their significance. Alternative approaches like, for example, Principal Component Analysis (PCA) (see [Martínez and Kak, 2001] for a comparison of LDA to PCA) create new features, combining the original ones, instead of selecting the most relevant ones.

The quality analysis outputs a feature set ranking, recommending the top ranked sets as potentially reasonable dimensions (see Tables 6.4 and 6.5 for the results). At this point in the process, human expertise is involved for the analysis of the system's recommended feature sets. The finally selected feature sets are then passed on to level II, aiming at the identification of concrete problem-solving styles within the respective dimension. Here, again SET_BOTH with the aggregated help state configuration was used, storing the data of 2008.

Table 6.4: This table shows the feature combinations ranked 1^{st} to 11^{th} and their respective $Q(FS_i)$ results.

Rank	$Q(FS_i)$	***Features***
1	5.8086	TRANS_PROB_H_H
2	4.0364	TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
3	3.9017	TRANS_PROB_H_H TRANS_PROB_H_E
4	3.7714	TRANS_PROB_C_H TRANS_PROB_H_E PERC_HELP_STEP
5	3.7149	TRANS_PROB_H_I TRANS_PROB_H_H
6	3.5501	TRANS_PROB_H_E PERC_HELP_STEP
7	3.5124	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
8	3.4429	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP
9	3.4331	TRANS_PROB_C_H PERC_HELP_STEP
10	3.4076	PRIOR_PROB_H TRANS_PROB_C_H PERC_HELP_STEP
11	3.3662	TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP

Table 6.5: This table shows the feature combinations ranked 12^{th} to 20^{th} and their respective $Q(FS_i)$ results.

Rank	$Q(FS_i)$	***Features***
12	3.3302	TRANS_PROB_H_H TRANS_PROB_E_H
13	3.2656	PRIOR_PROB_H TRANS_PROB_H_H TRANS_PROB_H_E
14	3.2652	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_E_H PERC_HELP_STEP
15	3.2270	PRIOR_PROB_C
16	3.2250	TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
17	3.2092	TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP
18	3.2090	TRANS_PROB_C_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
19	3.2026	TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H PERC_HELP_STEP
20	3.1810	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP

As can be seen in the ranking tables, feature sets with 6 or 7 features are not among the highest-ranked combinations. The highest-ranked 6-feature setting (holding the attributes $PRIOR\_PROB\_H$, $TRANS\_PROB\_C\_H$, $TRANS\_PROB\_H\_I$, $TRANS\_PROB\_H\_H$, $TRANS\_PROB\_E\_H$ and $PERC\_HELP\_STEP$) can be found on position 39, position 79 holds the best-ranked 7-feature combination (including $PRIOR\_PROB\_H$,

$TRANS\_PROB\_C\_H$, $TRANS\_PROB\_H\_I$, $TRANS\_PROB\_H\_H$, $TRANS\_PROB\_H\_E$, $TRANS\_PROB\_E\_H$, and $PERC\_HELP\_STEP$).

Although the positions 39 and 79 do not seem promising at first glance, regarding the total number (in this case, ~ 140000), these feature combinations can still be considered potentially relevant.

The highest ranked feature sets seem to involve variations of the *Help-Seeking* dimension discussed before. For level II, this dimension was manually defined, using recent descriptions reported in related literature as a basis. Thus, the results of clustering on this level provide clear indications for success in both directions; on the one hand, the open-ended clustering process of level III confirmed the assumptions of level II, on the other hand, level III clustering results are supported by profound, well-established definitions.

Table 6.6 takes up again the results listed in Tables 6.4 and 6.5 and sorts them according to the number of features in the feature sets, thus providing a better way to compare and analyze them.

After the most promising feature combinations have been chosen for every number of features, experimental clustering is performed on these data sets (see the results in Table 6.7).

Furthermore, the concrete types identified within the dimensions found by level III clustering, are summarized in Table 6.8 and can be described as in the following paragraphs.

Dimension 1: $Rank_G = 1, n = 1$
The first dimension contains only one feature and concentrates on a user's tendency to request help in sequences. As indicated by the clustering results, the selected feature is of high discriminatory capacity. Thus, distinct kinds of user behaviour could be identified (compare, for example, clusters 2 and 4), resulting in the following concrete types:

- $T_{1.1}$, preferring help requests in sequences (as can be seen in clusters 1, 3 and 4),
- $T_{1.2}$, not requesting help in sequences (as can be seen in cluster 2), and
- $T_{1.3}$, requesting help in sequences occasionally (as can be seen in cluster 0).

Dimension 2: $Rank_G = 1, n = 2$
Here, the dimension is described by two features, adding the tendency to request help as a last activity in a problem-solving sequence (often without having solved the problem) to the previously described tendency to sequentially request help. The following types were identified:

- $T_{2.1}$, preferring sequential help requests and not requesting help as a last activity in a sequence (see clusters 1, 2, and 4), and
- $T_{2.2}$, occasionally requesting help in sequences and as a last activity (see clusters 0 and 3).

Dimension 3: $Rank_G = 1, n = 3$
Here, again the previously described two features become part of the new dimension that adds as a third feature a user's general percentage of help requests. Several significant concrete types could be identified:

- $T_{3.1}$, not requesting help in sequences, not concluding problem-solving sequences with help requests, and generally requesting little help (see cluster 2),
- $T_{3.2}$, also not concluding problem-solving sequences with help requests, but requesting help in sequences and generally requesting a high amount of help (see clusters 1 and 3),
- $T_{3.3}$, tending to close problem-solving sequences with help requests (without having completed the task), generally not requesting much help, and not tending to request help in sequences (see cluster 4),
- $T_{3.4}$, not tending to use help at all (see cluster 0). Note, that the percentage of requested help is ~ 0.00, which renders the values indicating tendencies to request help in sequences or to close problem-solving sequences with help requests, irrelevant. Thus, we can consider this dimension more expressive, compared to the previous ones, and conclude that adding a basic statistical metric on the general use of help is a valuable indicator on the expressiveness of dimensions in general.

Dimension 4: $Rank_G = 1, n = 4$
Here, the probability of submitting a wrong answer directly following a hint request, is added to the features that were already part of the previous dimensions. In this dimension, a type already discovered in the previous one, could again be identified: $T_{3.4}$ (see cluster 0), in addition to the following:

- $T_{4.1}$, closing problem-solving sequences with a hint request in 100% of the cases and, in contrast to the behaviour of type $T_{3.3}$, not showing a very low help rate in general (see cluster 3),
- $T_{4.2}$, showing a very high help request rate, generally tending to request help in sequences and showing a low rate of incorrect attempts or closings after a hint request (see cluster 1),
- $T_{4.3}$, showing rather similar behaviour, compared to type $T_{4.2}$, but showing a lower help request rate and a slightly higher rate of quits after a hint request (see cluster 2), and
- $T_{4.4}$, showing a medium rate of help requests, incorrect submissions, and help requests directly following a help request, and generally not tending to quit after a help request (see cluster 4).

Type $T_{4.1}$ is however, only exhibited by a low number of students, as it could be identified in only about 1% of the problem-solving sequences analyzed.

Dimension 5: $Rank_G = 1, n = 5$
This dimension is mainly defined by the general help rate, the tendency for sequential help

requests, the rate of help requests after the submission of a correct answer, and the probability for a help request being the first activity within a problem-solving sequence. Yet, the feature describing the help rate directly after a correct attempt, proved not to be sufficiently discriminatory and is thus only marginally relevant for the recognition of different types. The following new types could be identified within this dimension:

- $T_{5.1}$, showing a high general help request rate, a tendency to request help in sequences, and a high rate of help requests as a first activity within a sequence (see clusters 1 and 2),
- $T_{5.2}$, exhibiting a tendency to request help in sequences and a medium general help rate, and not tending to request help as a first activity within a sequence (see cluster 3), and
- $T_{5.3}$, taking up the characteristics of $T_{5.2}$ regarding the prior probability for help requests, but showing a lower general help rate and a lower tendency to request help in sequences.

In addition to these types, again type $T_{3.4}$ could be recognized in cluster 0.

Dimension 6**:** $Rank_G = 1, n = 6$
Here, the probability of incorrect attempts directly following a help request, is added to the features already described, to form a new dimension. Clustering along this dimension again led to the recognition of type $T_{3.4}$ (see cluster 1). Additionally, the following types could be identified:

- $T_{6.1}$, showing a high general help rate, a high probability for help requests as a first activity within a sequence, and a tendency to request help in sequences (see clusters 0 and 1),
- $T_{6.2}$, showing a high tendency to request help in sequences (see cluster 2), and
- $T_{6.3}$, not tending to request help in sequences but showing a high probability of incorrect attempts directly following a help request (see cluster 3).

Types $T_{6.2}$ and $T_{6.3}$ are similar regarding the general help rate (which is low in both cases), the (low) prior probability for help requests, and the (very low) probability of help requests directly following a correct attempt.

Dimension 7**:** $Rank_G = 1, n = 7$
Here, the probability of a help request being the last activity within a sequence, is added to the list of features already described. Again, type $T_{3.4}$ could be found by the clustering process along this dimension. One additional new type could be identified:

- $T_{7.1}$, tending to close a problem-solving sequence by requesting help (see cluster 3; as already described, this kind of behaviour is undesirable because it indicates that a learner has given up before having solved the problem).

The results of level III clustering indicate that two factors have to be taken into account when choosing a certain number of features:

1. dimensions with a very low number of features (or even only one) might be indicative of concrete problem-solving types but results could be distorted, whereas
2. dimensions with a very high number of features might be better suitable for the identification of a range of "subtypes" (many of which could potentially be combined without causing information loss) than for the recognition of the most significant concrete types.

Thus, a medium number of features is usually the best choice for scenarios like the one presented here, aiming at the identification of (new) dimensions. More concretely, based on the findings of the different clustering levels, a number ranging between a fourth and a third of the overall number of features in the base data set, can be suggested. Level III clustering in this case suggests a domination of the *Help-Seeking* dimension, as indicated by the results reported in Table 6.7, although already the 5th-ranked configuration for the group with $n = 7$ provides evidence for a different dimension consisting of the following features:

- $PRIOR\_PROB\_C$
- $PRIOR\_PROB\_H$
- $TRANS\_PROB\_C\_H$
- $TRANS\_PROB\_H\_I$
- $TRANS\_PROB\_H\_H$
- $TRANS\_PROB\_E\_H$
- $PERC\_HELP\_STEP$

The clustering process along this dimension led to the identification of the following types characterized by

1. a high prior probability for correct attempts, a very low help request rate, and the tendency not to request help in sequences for the first type, and
2. a medium prior probability for correct attempts, a low prior probability for help requests, and a generally rather low help request rate for the second type.

The second type can be compared to the *Trial and Error* type previously discussed in the context of level I clustering.

6.5 Summary

This chapter presented a clustering approach facilitating the detection of patterns in learner activities on different levels. Generally, the approach was applied on the models that user interaction data has been transformed into in a preceding sequence modeling step (see Chapter 5).

On the first level, the process aimed at the recognition of predefined problem-solving “styles” defined by specific values for the different components of the models. On the second level, the aim was to identify predefined “dimensions” of problem-solving behaviour in user activity data that could entail different manifestations, i.e. concrete styles. On the third level, the system was not given definitions of what should be found beforehand; the process aimed at the discovery of potentially meaningful “‘new” dimensions and styles. For the evaluation of the significance of the findings of third-level clustering, human expertise was involved.

The results of all levels of clustering can become a basis for the adaptive system interventions suggested in the following Chapter 7, where potentially reasonable system reactions to user behaviour (manifested in concrete “styles” within different “dimensions”) are identified.

Table 6.6: This table shows the feature combinations ranking split into groups containing sets with equal number of elements n. The $Gr.$-column shows the number of features in the respective group, $Rank_G$ is the rank within a group. For every $RANK_G$ in a specific group, the following information is provided: the features in the set, the overall ranking (not group-related), and the $Q(FS_i)$ results.

Gr.	$Rank_G = 1$	$Rank_G = 2$	$Rank_G = 3$	$Rank_G = 4$
$n = 1$	TRANS_PROB_H_H	PRIOR_PROB_C	TRANS_PROB_E_C	TRANS_PROB_I_C
	5.0806	3.2270	2.9853	2.6309
	1	15	37	134
$n = 2$	TRANS_PROB_H_H TRANS_PROB_H_E	TRANS_PROB_H_I TRANS_PROB_H_H	TRANS_PROB_E_H PERC_HELP_STEP	TRANS_PROB_C_H PERC_HELP_STEP
	3.9017	3.7149	3.5501	3.4331
	3	5	6	9
$n = 3$	TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_H_H TRANS_PROB_H_E
	4.0364	3.7714	3.4076	3.2656
	2	4	10	13
$n = 4$	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H PERC_HELP_STEP	TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_E_H PERC_HELP_STEP
	3.5124	3.4429	3.3662	3.2651
	7	8	11	14
$n = 5$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H PERC_HELP_STEP
	3.1810	3.1348	3.1138	3.0901
	20	23	25	26
$n = 6$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E PERC_HELP_STEP
	2.9672	2.9378	2.8925	2.8755
	39	44	52	59
$n = 7$	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_C_H TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_I TRANS_PROB_C_H TRANS_PROB_H_H TRANS_PROB_H_E TRANS_PROB_E_H PERC_HELP_STEP	PRIOR_PROB_H TRANS_PROB_C_H TRANS_PROB_H_C TRANS_PROB_H_I TRANS_PROB_H_H TRANS_PROB_E_H PERC_HELP_STEP
	2.7744	2.6684	2.6400	2.6299
	79	116	128	136

Table 6.7: This table shows experimental clustering results based on the top ranked feature sets (TRFS) listed in Table 6.6. In order not to neglect variations of similar types of behaviour, the relatively high number of 5 clusters was used. The results listed here are an example of what a human observer would see when applying level II clustering on the dimensions suggested by level III. Of course, a human observer would be provided not only one TRFS but several. The values for the clusters denote the mean for the respective feature in this cluster.

TRFS	***Features***	$Cluster0$	$Cluster1$	$Cluster2$	$Cluster3$	$Cluster4$
$n = 1$	TRANS_PROB_H_H	0.25	0.69	0.03	0.52	0.77
$n = 2$	TRANS_PROB_H_H	0.25	0.69	0.75	0.15	0.51
	TRANS_PROB_H_E	0.25	0.06	0.02	0.27	0.01
$n = 3$	TRANS_PROB_H_H	0.25	0.74	0.01	0.60	0.05
	TRANS_PROB_H_E	0.25	0.03	0.00	0.06	0.88
	PERC_HELP_STEP	0.00	0.67	0.10	0.34	0.28
$n = 4$	TRANS_PROB_H_I	0.25	0.03	0.08	0.00	0.23
	TRANS_PROB_H_H	0.25	0.74	0.69	0.00	0.34
	TRANS_PROB_H_E	0.25	0.04	0.26	1.00	0.08
	PERC_HELP_STEP	0.00	0.71	0.48	0.24	0.23
$n = 5$	PRIOR_PROB_H	0.00	0.76	0.45	0.11	0.09
	TRANS_PROB_C_H	0.00	0.01	0.04	0.01	0.02
	TRANS_PROB_H_H	0.24	0.72	0.70	0.66	0.18
	TRANS_PROB_E_H	0.30	0.12	0.14	0.09	0.18
	PERC_HELP_STEP	0.00	0.72	0.59	0.34	0.09
$n = 6$	PRIOR_PROB_H	0.75	0.00	0.45	0.11	0.07
	TRANS_PROB_C_H	0.00	0.00	0.05	0.01	0.01
	TRANS_PROB_H_I	0.04	0.25	0.07	0.10	0.59
	TRANS_PROB_H_H	0.73	0.25	0.67	0.52	0.01
	TRANS_PROB_E_H	0.71	0.03	0.42	0.10	0.09
	PERC_HELP_STEP	0.72	0.00	0.58	0.15	0.07
$n = 7$	PRIOR_PROB_H	0.00	0.43	0.74	0.31	0.09
	TRANS_PROB_C_H	0.00	0.04	0.04	0.05	0.01
	TRANS_PROB_H_I	0.25	0.07	0.04	0.01	0.18
	TRANS_PROB_H_H	0.25	0.71	0.73	0.13	0.48
	TRANS_PROB_H_E	0.25	0.04	0.04	0.83	0.03
	TRANS_PROB_E_H	0.03	0.41	0.71	0.32	0.08
	PERC_HELP_STEP	0.00	0.57	0.72	0.28	0.26

Table 6.8: This table provides a concise overview about the types discovered on the third level of clustering, including a detailed description.

Type	***Description***
$T_{1.1}$	Requests help in sequences.
$T_{1.2}$	Does not request help in sequences.
$T_{1.3}$	Occasionally requests help in sequences.
$T_{2.1}$	Tends to request help in sequences and does not conclude problem-solving sequences with help requests.
$T_{2.2}$	Occasionally requests help in sequences and occasionally concludes problems with help requests.
$T_{3.1}$	Does not request help in sequences, does not end a problem-solving sequence with help requests, requests only little help.
$T_{3.2}$	Tends to request help in sequences, does not end problems with help requests, tends to request a lot of help.
$T_{3.3}$	Does not request much help and when so, not in sequences, shows a strong tendency to end problem-solving sequences with hints.
$T_{3.4}$	Does not use help at all.
$T_{4.1}$	Stops problem-solving sequences with help requests in 100% of the cases.
$T_{4.2}$	Shows a very high help request rate, a strong tendency to request help in sequences and a very low rate of incorrect submissions or quits after a hint.
$T_{4.3}$	Behaves in a similar way as $T_{4.2}$, shows a slightly lower rate of help requests and a medium rate of quits after a hint.
$T_{4.4}$	Shows a medium rate of help requests, a medium rate of incorrect attempts or further help requests after a help request, and a low rate of quits after a help request.
$T_{5.1}$	Shows a high help rate, a high prior probability for the use of help and a tendency to request help in sequences.
$T_{5.2}$	Shows a high help sequence rate, a medium overall help request rate and a relatively low prior probability for help.
$T_{5.3}$	Is similar to $T_{5.2}$ but shows a lower rate of help sequences and a lower overall help rate.
$T_{6.1}$	Shows a relatively high prior probability for help requests, a high general help rate and a tendency to help request sequences.
$T_{6.2}$	Like $T_{6.1}$, and shows a high help sequence rate.
$T_{6.3}$	Like $T_{6.1}$, and shows a very low help sequence rate and a relatively high percentage of incorrect attempts after a hint.
$T_{7.1}$	shows a strong tendency to close a problem-solving sequence with a help request.

Chapter 7

Closing the Circle – Adaptive System Interventions

Chapter 6 discussed a clustering approach aiming at the identification of learning behaviour on different levels. This chapter builds upon these findings and describes ways how the information can be re-integrated into the adaptation cycle by enhancing the user models, which then become a basis for adaptive system interventions. A previous version of this chapter also appeared in [Köck and Paramythis, 2010].

As introduced in Section 1.1.2, adaptive support in the area of e-learning can be categorized into *learning support* and *collaboration support*. While the first focuses on content and navigation and mostly targets the individual learner, the second puts emphasis on the collaboration process within a group. Thus, we could also distinguish between *individual learner support* (see, for instance, [Koedinger and Aleven, 2007]) and *collaboration support* (see, for instance, [Soller et al., 2005] or [Walker et al., 2009]).

The case study reported in this thesis mainly focused on the identification of styles and dimensions in the domain of problem-solving. This chapter takes up the types and dimensions found by the different levels of the clustering process and demonstrates how the adaptation cycle can be closed by the system reacting to such behaviour in a personalized way.

More concretely, the chapter will again discuss the *Trial and Error* problem-solving type, the different types within the *Help-Seeking* dimension, i.e., H_1, H_2/H_4 and H_3, and the "open dimensions". Note, that the decision for a specific kind of personalized support strongly depends of the didactic approach that is applied in the particular case. Thus, the examples described here should not blindly be considered a sample solution for *all* similar scenarios.

7.1 Supporting Individual Users

This section discusses how the individual learner, exhibiting one of the identified styles, can be supported in order to optimize the personal learning process.

7.1.1 Trade-Offs and Decisions

Focusing on the provision of help, we can distinguish between different system activities, depending on the behaviour of the respective user. In many cases, the main question is whether to provide or not provide help for a particular student solving a particular problem in a specific situation.

Thus, one faces the trade-offs between giving and withholding information, a problem [Koedinger and Aleven, 2007] define as *assistance dilemma*. This issue is further also addressed by [Rummel and Krämer, 2010] and [Borek et al., 2009], who discuss the decision of a system when and to what degree a student should be provided with (additional) information or assistance. The potential decisions reach from not providing help at all to providing full solutions to a problem [Razzaq and Heffernan, 2009], depending on the particular circumstances.

Having decided for the level of help a student should generally receive in a particular situation, a system, for example, an intelligent tutor in an ITS, has to determine the content of the help. Usually, this decision is based on the tutor's *production rule model* which represents the competences students should acquire with the help of the tutor [Koedinger and Aleven, 2007]. A tutor is thus able to autonomously solve the problems provided to the students.

In addition to the production rule model, an ITS usually implements two algorithms enabling the tutor to interpret students' activities and to determine the optimal portion and granularity of help:

1. *model tracing*, a process that uses the model mentioned before for the interpretation of activities, and
2. *knowledge tracing*, a process that aims at estimating how well a student has understood a particular part of the content.

Both are used subsequently to individualize feedback, instruction and help for the students.

The Andes tutoring system [VanLehn et al., 2005] that provided the data set (Data Set II as described in Section 3.2.2) used for the case study reported in this thesis, for example, uses a variant of the *model tracing* algorithm.

As indicated by the algorithms mentioned above that are usually applied in ITSs, tutoring systems mostly rely on knowledge-driven models. However, one could also consider combining these models with activity models derived from patterns that have been detected before. Such a combined version could well be considered here, because the findings of the different clustering levels discussed in Section 6.4 would provide a profound basis for the activity-driven part.

Although focusing on optimally supporting the learner, a system has to consider that students might not in all cases exhibit constructive behaviour only, and thus provide means to suppress undesired behaviour like *gaming the system* as described by [Baker et al., 2006] and [Baker et al., 2008].

Different types of system decisions regarding interactivity are also listed by [Koedinger and Aleven, 2007], including feedback content, hint content and timing. For the rest of this chapter, we concentrate on the following approaches to personalize help for a user that are most relevant for the types of behaviour identified before:

- Hint Tailoring,
- Hint Withholding, and
- Proactive Hint Delivery.

7.1.2 Hint Tailoring

Hint tailoring involves the limitation of the available help as a reaction to student behaviour and can be approached in different ways:

1. by reducing or increasing the number of hints, or
2. by reducing or increasing the granularity of information within the hints.

The first option might be the better choice for the types H_2 and H_4 within the *Help-Seeking* dimension. Students of this type seem to have a natural aversion for the submission of incorrect answers. However, these students also tend to use a disproportionate amount of help. In such cases, the system might aim at generally encouraging a more independent problem-solving approach, which could be done by limiting the available hints for these students. Additionally or alternatively, the system could also consider tailoring the information within the hints, which might be the better choice in cases where the limited number of hints discourages the students.

Furthermore, hint tailoring can be well applied for students of the types in the open dimensions $T_{2.*}$ and $T_{7.*}$. These students tend to quit after a hint request, i.e., without having solved the problem. A potentially successful approach to support these students could be to increase the amount of information with an increasing number of hint requests in a sequence. Additionally the system might consider sporadically providing very extensive hints in order to guide the students to the correct answer proactively. In this case, the amount of information in the hints should be reduced again later, after the student has successfully completed a certain number of problems and thereby gained confidence and motivation.

The $T_{4.*}$ dimension also contains types where hint tailoring is potentially beneficial. This dimension needs to be addressed particularly carefully because the concrete types identified there are quite distinct and require different treatment. The types $T_{4.2}$ and $T_{4.3}$ show a high help request rate and a strong tendency to request help in sequences. In such cases the system could decide to lower the amount of information within the hints if too many hints are requested, or to limit the number of available hints for these students.

The $T_{4.1}$ type is certainly the most problematic one, although rather rare – students of this type quit a problem-solving sequence in 100% of the cases after having requested a hint. In this case, the reason for this kind of behaviour should be identified before a decision about how to support these students, is made. If an analysis would, for example, result in the assumption that an extremely low level of tolerance is the reason for the exhibited behaviour, the content of the hints could be tailored to the student's needs by adding a motivating message or by providing an "almost-solution" to the problem. Another possible reason for $T_{4.1}$ behaviour could be insufficient knowledge about the system and its functionalitites. In this case, hints could be tailored by adding information about how to use the system in general and that particular hint in specific.

Regarding the types in the $T_{6.*}$ dimension, where a student's probability of submitting a wrong answer directly after having requested a hint, becomes an important criterion, the system could, in cases where the students generally use a disproportionate amount of help, decide to limit the available hints. These students are potentially likely to misuse help (related types are also discussed by [Aleven et al., 2006].

7.1.3 Hint Withholding

Hint withholding actively repeals the accessibility of (particular) hints for specific users at a specific time. This seems to be a relatively drastic intervention at first glance but is reasonable in cases where students request hints before having tried to understand the learning content or where an obviously disproportionate amount of help is used.

Hint withholding is applicable for the types identified within the $T_{5.*}$ dimension where students tend to request a hint as a first activity within a problem-solving sequence. This kind of behaviour indicates that a student is most probably not well prepared or even trying to abuse help in order to reduce the learning efforts. In such a case, the system could decide not to provide initial hints at all, thus enabling the help functionality at a later point in time when the students have acquired a certain amount of knowledge.

7.1.4 Proactive Hint Delivery

Proactive hint delivery aims at encouraging the use of help by actively offering it, i.e., even without users having requested it, or by promoting the help request functionalities more obviously. An approach like this is mostly best applicable in cases where students show a low tendency to use help, tend to make uneducated guesses, or where a hint would enable students to solve a problem but is not requested.

Proactive hint delivery is, for example, well applicable for students showing the *Trial and Error* style or students of the type H_1 in the *Help-Seeking* dimension. These students should be encouraged not to make uneducated guesses but instead process some additional information.

The hints provided in such a case should not necessarily include all information and details available but rather be sufficient to bring the student on the right track.

For students of type H_3 in the *Help-Seeking* dimension, showing a high inhibition threshold regarding the request of help, proactive hint delivery is also a well applicable approach for support. These students should be encouraged to find a more balanced use of help, because they tend to request help not very often, but if they do, they request it in long sequences. Thus, the system has to decide whether a hint is helpful in the particular case, which could be done, for example, based on the time a student spends on a problem.

Furthermore, proactive hint delivery can be suitable for the types in the “open dimension” $T_{3.*}$. These students show a tendency not to use help at all. This kind of behaviour can be problematic in different ways, for example:

- when students would be able to solve a task with a hint but quit instead of requesting one, or
- when students who are already well skilled would benefit from additional knowledge through a hint.

7.2 Implications for Collaboration

This section discusses how collaboration could be adaptively supported in the context of e-learning. In general, collaborative learning can be split into two categories as explained by [Dillenbourg et al.,]:

1. an amalgamation of independent cognitive systems with message exchange, and
2. a single cognitive system with its own properties.

The first understanding suggests the individual as the unit of analysis whereas the second one suggests the group for this purpose. The approach introduced in this thesis can generally be applied from both perspectives.

However, as the case study treated users as individuals, the information gained serves best when designing collaboration support based on the characteristics of the individual learners.

[Paramythis, 2008] lists the following set of requirements as prerequisites for adaptive collaboration support:

- capability to automatically collect / infer user- and learner profile data of individual learners,
- capability to collect / infer and model collaboration activity data for individual learners,
- capability to represent and employ algorithms / strategies that govern how learner information is used to identify appropriate collaboration partners,

- the opportunity to allow for alternative policies for and approaches to group initiation.

It has already been shown that the approach presented here is clearly capable of the first requirement. As data collection and modeling as described here for individual users' activities can likewise be applied for collaboration data (these processes are rather specific to the format of the data than to the source it originates from), it can be assumed that the approach is capable of the second requirement also. The new model information should be used to provide adaptive collaboration establishment support, including the third requirement, keeping the implementation sufficiently generic to allow for the fourth requirement.

Furthermore, Paramythis splits collaboration support into the two phases

1. adaptive support for collaboration establishment and
2. adaptive support during the collaboration process.

Again, the proposed approach is generally applicable for both, however, the nature of the information analyzed here is better suited for collaboration establishment support, usually based on learner's learning characteristics that could be either explicitly provided by the user or observed by the system by analyzing the interaction process [Paramythis, 2008], [Carro et al., 2003b], [Quignard and Baker, 1999].

7.2.1 Adaptive Collaboration Establishment Support

As mentioned previously, there are additional factors influencing the decision for a particular kind of adaptive support, like the didactical approach and pedagogical strategies, teaching concepts and learning theories applied. This is also true for the concrete case of adaptive collaboration establishment support, which includes encouraging students to cooperate with others, or recommendations of tools to use for collaboration, or partners to collaborate with [Carro et al., 2003a].

Group synthesis recommendations are usually based on specific rules that consider, for instance, users' preferences, backgrounds, interaction behaviour, etc. It may be generally desirable for a system to group students that could potentially benefit from the cooperation. In this case, criteria like complementarity or competitiveness could be taken into consideration [Alfonseca et al., 2006].

Examples for analysis of individual learner behaviour that could later become the basis for collaboration support are provided by Alfonseca et al. and by Liu et al., who model learning styles based on the Felder and Silverman model [Felder and Silverman, 1988], [Felder and Brent, 2005]. The Felder and Silverman model uses a categorization of learning styles along five dimensions:

- active / reflective,
- sensing / intuitive,

- visual / verbal,
- sequential / global, and
- inductive / deductive.

Alfonseca et al. and Liu et al. conclude that

- learning styles affect the performance of students when working together,
- for the dimensions *active / reflective* and *sensing / intuitive*, the mixed pairs tend to work better,
- heterogeneous groups in general get better results, and
- students themselves tend to group randomly without respect to their learning styles.

The results reported there indicate that utilizing the models as a basis for group synthesis recommendations is a reasonable goal, and, additionally, that learning styles are a relevant criterion to base grouping algorithms on. When deciding on the synthesis of groups in the scope of problem-based collaborative learning, taking into account individuals' problem-solving styles, effects might be even more pronounced.

7.2.2 Adaptive Support During the Collaboration Process

Regarding the second phase of adaptive collaboration support mentioned by [Paramythis, 2008], i.e., adaptive support during the collaboration process, a similar kind of analysis of activities in group settings is required in addition to the analysis of individual users as described before.

Data can be monitored by any available kind of collaborative environment and may include activities within tools for multi-user communication and cooperation, like, for example, a chat, a forum, a wiki, or audio / video conferencing facilities. A statistical analysis can provide a basis for the user model, including a user's level of activity in the group, tendencies to correct other users' contributions or to initiate new ones.

Regarding the approach presented in this thesis, the analysis of group activity sequences is even more important. As proposed by the related literature, computational models for analyzing such sequences to determine metrics such as the *centrality* of group members or the *cohesion* of a group (see, for example, [Suh and Lee, 2006]), might be well applicable.

However, more "dialogical" forms of analysis have typically been constrained to group discussions, and require annotation of activities by experts, or content analysis of exchanged messages (again, see [Suh and Lee, 2006] for an example).

The application of the proposed approach for the analysis of group activity sequences could potentially lead to

- the detection of behavioural patterns of individual learners, regarding also their contact within collaborative learning settings, and
- the detection of patterns emerging in the behaviour of a group as a whole.

The first type of information, based on the analysis of individual learners' activities can potentially enhance the collaboration process because individual users' characteristics contribute to the group model and influence the collaboration behaviour.

The second type of information cannot only be fed back into the user models and become basis for further adaptive collaboration support, but potentially form the basis for novel kinds of adaptation for collaboration. Each of these cases would require a different representation of activity sequences.

7.3 Summary

This chapter presented a selection of adaptive system interventions specifically tailored to the kinds of user behaviour identified before in Chapter 6. Interventions were suggested at two different levels: first, to support the individual user's learning process, and second, to support the learning process of a user group.

Both the data acquisition and modeling process described previously in Chapters 4 and 5 as well as the suggested adaptive system interventions described in this chapter may trigger concerns regarding system security and user privacy issues. Thus, these issues are further discussed in the following Chapter 8.

Chapter 8

Security in Personalized Systems

This chapter discusses security and privacy in adaptive systems. The approach introduced in the previous chapters aims at creating a model of the user that provides enough information about the user's behaviour that the system can offer adaptations based on them. This approach, and personalized systems in general, strongly depend on the collection and analysis of user activity data in order to determine users' preferences and characteristics related to the respective field of interest, like in this case, problem-solving and learning.

Admittedly, this leads to potential security risks which have to be considered during the full process of the adaptation cycle. Thus, security arose as a very important branch of research in the area of adaptive systems and got even more important as new mining and analysis techniques lead to more and more fine-grained results. Personalized systems that also subsist on users having faith in their reliability and trustworthiness therefore need to carefully address these concerns.

Security in general comprises several goals, namely, authenticity, data integrity, confidentiality, availability, and anonymity / pseudonymity. In the context of adaptive systems, privacy can be considered as an "umbrella", involving, for example, anonymity / pseudonymity or confidentiality. Although these issues are often mentioned in the same context, they should be addressed individually.

In the most general sense, some security issues pertain technical aspects of and demands on a system, whereas authenticity or anonymity, at a higher level, concern the user directly, i.e. the system may specify if, and to what degree a user's personal "space" is being intruded. Thus, we can conclude that the lower-level security issues are needed to ensure higher-level ones.

The following sections discuss in more details different security issues in personalized systems and their implications on design and development of such environments in general, and in the context of the proposed approach in specific.

8.1 Data Integrity, Authenticity and Confidentiality

This section discusses general security issues, and particularly considers the context of adaptive systems.

8.1.1 Definitions and Description

The OECD Guidelines for the Security of Information Systems [OECD, 1992] provide a basis for deliberations on security in the domain of user modeling. [Summers, 1997] and [Schreck, 2001] identify the most important factors and discuss them as follows:

1. *accountability*, signifying that "all parties concerned with the security of information systems (owners, providers, users, and others) should have explicit responsibilities and accountability",
2. *awareness*, signifying that "all parties should be able to readily gain knowledge of security measures, practices, and procedures",
3. *ethics*, signifying that "information systems and their security should be provided and used in ways that respect the rights and legitimate interests of others",
4. *multidisciplinary principle*, signifying that "security measures should take into account all relevant viewpoints, including technical, administrative, organizational, operational, commercial, educational, and legal",
5. *proportionality*, signifying that "security measures should be appropriate and proportionate to the value of and degree or reliance on the information systems and to the risks of harm",
6. *integration*, signifying that "security measures should be coordinated and integrated with each other and with other measures, practices, and procedures of the organization so as to create a coherent system of security",
7. *timeliness*, signifying that "parties should act in a timely and coordinated way to prevent and to respond to security breaches",
8. *reassessment*, signifying that "security should be reassessed periodically as information systems and their security needs change", and
9. *democracy*, signifying that "the security of information systems should be compatible with the legitimate use and flow of information in a democratic society".

Schreck further splits security into three main factors: secrecy, integrity, and availability (also see [Summers, 1997]). The first factor could also be considered a privacy issue, especially as Schreck points out anonymization as a requirement for secrecy. We will concentrate on the second factor here. Integrity can in general be understood in this context as the consistency of processed data with the world it describes [Schreck, 2001]. It can be further split up into external (i.e., from the client's perspective) and internal (i.e., from the developer's perspective) integrity.

8.1.2 Potential Threats

As already mentioned, publicly accessible adaptive systems may entail security risks. This concerns all kinds of adaptive systems, although recommender systems are probably the most popular form of personalized environments.

Due to (more or less, see Section 8.2) anonymous / pseudonymous access, potential attackers cannot always be easily distinguished from ordinary users in these systems. Thus, it is relatively easy for attackers to inject multiple and / or biased profiles in order to force a system to adapt in a way advantageous to them [Mobasher et al., 2007b].

These kinds of attacks are labeled *profile injection attacks* or *shilling attacks* and constitute one of the most publicized security risks in adaptive systems. Discussion will concentrate on Collaborative Filtering (CF) (see, for instance, [O'Sullivan et al., 2002], [Herlocker et al., 2000], or [Sarwar et al., 2001]) based systems, as most recommender systems, and also most adaptive multi-user systems rely on CF. However, in the context of adaptive e-learning, we have to consider additional issues other than those related to CF, as discussed later.

CF techniques utilize multiple users' data in order to predict the interests of an individual user and offer recommendations based on this predictions, as an alternative or supplement to pure content-based recommendation [O'Donovan and Smyth, 2005]. The latter is, however, also highly relevant in the context of adaptive e-learning.

Although fake profiles could theoretically be injected into every system managing users, these attacks are naturally only effective in environments where the system's behaviour can be influenced by users' behaviour, i.e., adaptive ones.

Profile injection attacks can be categorized based on the knowledge required by the attacker, the intent of an attack, and the size of the attack. Intentions can be categorized as follows [Mobasher et al., 2007b]:

- A *push attack* means that an attacker inserts a profile in order to make items more likely to be recommended.
- A*nuke attack* means that an attacker aims at making items less likely to be recommended.
- A *vandalism attack* occurs if an attacker simply aims at making the system function poorly without any further personal or general motivations. This kind of attack is rare as most attackers are in some way economically motivated.

Push attacks can be further divided into sampling attacks [O'Mahony et al., 2004], random attacks [Lam and Riedl, 2004], average attacks [Lam and Riedl, 2004], consistency attacks [Burke et al., 2005b] and segmented attacks [Burke et al., 2005a]. A sampling attack requires the attacker to have access to the rating database and is therefore mostly only of theoretical relevance. Both random and average attacks are based on randomly assigned item ratings in the attack profile. While with the random attack, the ratings are based on the overall distribution of user ratings, the average attack is based on average item ratings for all users.

A consistency attack is an attack that does not aim at manipulating items' absolute values but rather the consistency of ratings for different items. A segmented attack is an attack in which the attacker concentrates on a set of similar items (similar regarding their content) that have high visibility [Burke et al., 2005a].

Equivalently to push attacks, nuke attacks can be further categorized into random and average attacks.

8.1.3 Coping Strategies

This section discusses ways and strategies of dealing with profile injection attacks by prevention or detection.

Prevention of Attacks

The "success" of profile injection attacks is in general dependent on the respective recommendation algorithm applied. [Mobasher et al., 2007b] analyze several algorithms regarding their vulnerability to attacks. In general, CF algorithms can be divided into a user-based and an item-based category.

User-based CF relies on user-to-user similarity [Herlocker et al., 1999]. The first step after similarity computation is the selection of the most similar users and filtering of users with significantly low similarity values, before the actual prediction value for a specific item is computed.

Item-based CF is based on a comparison of items based on their rating patterns across users [Mobasher et al., 2007b]. The further process is similar to the one in *user-based CF*: first, the similarities between items are computed, next, the most similar items are selected and items with very low similarity values are removed. The item-based version has been considered more robust to attacks compared to the user-based variant [Lam and Riedl, 2004], is, however, still vulnerable.

Therefore, Mobasher et al. recommend a hybrid approach called *semantically enhanced CF* [Mobasher et al., 2004], [Jin and Mobasher, 2003], extending the item-based approach and relying on a combination of rating similarity and semantic similarity measures. This is possible for all systems that do not depend on profile data only, which is the case when the collaboration component is only one of multiple recommendation components.

Additional semantic information can be used to infer a user's interest in a particular item, which is particularly helpful in, for example, cases where not many ratings are available for an item (e.g., if it is a new item). The approach presented by Mobasher et al. involves (semantic) domain-specific information that has been retrieved from the web and combines it with user-item mappings so that predictions are ultimately based on a conjunction of both. As experiments with attacks on user-based, item-based and semantically enhanced CF have shown, the hybrid (i.e., semantically enhanced) approach is able to reduce the impact of attacks.

A different aspect on security in personalized environments is introduced by [Lum, 2003] and [Lum, 2007]. There, scrutable user models in decentralized adaptive systems are discussed. Lum states that due to the fact that user models contain personal data, users should be able to access their respective models at any time and to maintain their models regarding what parts of it should be made public and what not. In order to achieve this goal, the author suggests a decentralized way of storing the user models, i.e., the client would play a more important role than in traditional user modeling systems. This approach is, however, mainly relevant in the context of the sampling attack model where the attacker needs to gain access to the system's database(s).

Concluding, Mobasher et al. argue that as long as it is possible for users to create new profiles themselves and to affect the system's output, profile injection attacks are not avertible. Therefore, it is not only necessary for the developer of a system to make the algorithms used as robust and stable as possible, but also to integrate methods for the *detection* and *neutralization* of attacks.

Although simply making the creation of a profile more difficult can significantly contribute to the discouragement of profile injection attackers, this is not a fully appropriate solution in some cases for various reasons. First, it may keep users from participating, it further involves increased efforts for the system owner, and it is mainly potentially successful where the system relies on explicitly provided data to a great extent.

Detection of Attacks

Different approaches of attack detection can, for example, be based on *profile classification* as introduced by [Mobasher et al., 2007b], or *anomaly detection* as described by [Mobasher et al., 2007a].

Anomaly detection focuses on the discovery of items with suspicious trends. The process includes the selection of interesting features and estimation of the features' further distribution. Control charts can then be used to examine further features' development. For instance, if a new item's average rating falls outside the limits in the control chart, this might be indicative of a possible attack involving this item. However, Mobasher et al. point out that not all kinds of items must necessarily have the same rating distribution. The results of their analysis show that this approach works well on some kinds of items while it turned out to be less successful on others.

Profile classification aims at detecting suspicious profiles and is strongly dependent on reliable definitions of a "suspicious profile". Mobasher et al. list different categories of detection attributes for *profile classification* as follows: generic attributes for detection, model-specific attributes, and intra-profile attributes. Generic attributes rely on the assumption that the "overall statistical signature of attack profiles will differ from that of authentic profiles" [Mobasher et al., 2007b].

An attacker is usually unlikely to have complete knowledge of the ratings in a system (see also [O'Mahony et al., 2004], or [Mobasher et al., 2005]) and therefore attacking profiles should deviate from real ones according to the rating patterns exhibited. Generic attributes that should be able to capture this kind of distribution differences can be the following [Mobasher et al., 2007b], [Chirita et al., 2005]:

- *Rating deviation from mean agreement* examines a profile's average deviation per item (and by that identifying attackers).

- *Weighted degree of agreement* captures the sum of the differences of a profile's rating from the respective item's average rating divided by the item's rating frequency.

- *Weighted deviation from mean agreement* aims at helping to detect anomalies and puts high weight on rating deviations for sparse items.

- *Degree of similarity with top neighbours* determines the average similarity of a profile's neighbours because attackers' profiles are expected to show higher similarities to their neighbours than real profiles do.

- *Length variance* determines the discrepancy between the length of a specific profile and the average profile length.

Although generic attributes can successfully contribute to the identification of attack profiles in some cases, they are insufficient in others (e.g., for the distinction between attack profiles and eccentric real profiles [Burke et al., 2006a], [Burke et al., 2006b], [Mobasher et al., 2006], [Mobasher et al., 2007b]). Therefore, additional model-specific attributes are introduced that aim at the recognition of the distinctive signature of a specific attack model. Furthermore, intra-profile attributes complete the set of attributes. They do not focus on a specific profile's characteristics but on statistics across profiles [Mobasher et al., 2007b]. The results can be of high importance because often attackers do not inject one single profile but a number of automatically generated ones that target the same items.

8.1.4 Implications for the Proposed Approach

This section discusses the relevance of the presented security threats and strategies for adaptive e-learning systems in general and for the proposed approach in particular.

Effects on Adaptive E-Learning Systems

The previous sections concentrated on (CF-based) recommender systems as the best known form of personalized environments. Although one might, due to the popularity of systems like Amazon [Amazon.com, 2010], first think of e-commerce in this context, adaptive e-learning can be treated in a similar way.

On the one hand, CF is a technique employable everywhere where recommendations / predictions should be provided to individual users, based on what the system has learned from multiple users. Thus, every personalized multi-user environment, one of which a(n) (collaborative) e-learning system definitely is, can utilize CF as a basis for its actions. On the other hand, the context of adaptive e-learning involves additional risks regarding undesirable user behaviour (known as "gaming the system", as already introduced and further discussed later in this section).

The environment does not necessarily have to allow direct interaction between the users, which is not, for example, the case in a non-collaborative e-learning system. As introduced in Section 1.1, adaptive e-learning aims at providing personalized support for its learners, based on the underlying user model. Adaptive support can include different aspects like the recommendation of literature to read, of material to use, or of users to cooperate with (in case of a collaborative environment).

Thus, the named security threats and strategies generally apply to adaptive e-learning systems. However, some domain-specific characteristics have to be considered:

1. In many e-learning systems learners cannot create profiles themselves – this is done by instructors or administrators, or, alternatively, at least authorized by them. This mainly applies in e-learning environments that are bound to a specific institution; open learning environments on the web usually allow arbitrary users to register.

2. Regarding the different types of attacks, a push attack is more likely to happen in an e-learning environment than a nuke or vandalism attack. Learners are often in a less competitive situation than, for example, traders in an e-commerce system. There, it is desirable to promote the own product in comparison to other, similar ones, which might cause an attacker to perform a nuke attack on a competing product. In an e-learning context, it is more likely to influence one's own user profile in a positive way than to manipulate other users' in a negative way. A vandalism attack in the context of adaptive e-learning could be based on a user's intention of disrupting the system's (adaptation) functionality. This could, e.g., be caused by randomly reading or rating content in order

to influence future recommendations to other users. A vandalism attack is however generally rather rare, which is also true for e-learning systems.

The first characteristic shows that e-learning systems are somewhat less likely to become targets of profile injection attacks than other (recommender) systems. The second characteristic shows that attention must be specifically turned to the push attack here, although attack prevention would be relatively similar for nuke attacks.

Comparing user-based CF, item-based CF, and hybrid CF-approaches, the general trends also apply in the case of e-learning. However, concerning user-based CF, an additional problem must be considered: an e-learning system, especially if it is a non-open-access one, tends to have significantly fewer users than an open recommender system. Thus, the inferences based on user-to-user similarities might be less reliable than desirable. Therefore, the item-based version is better applicable than the user-based one, and, as described for CF-based systems in general, its performance can be improved by adding domain-specific information.

Retrieving useful domain-specific information is relatively simple in the case of e-learning content, as there mostly exist similar learning units for one topic, often even following universal standards or specifications that can be used as a knowledge base. Regarding communication and cooperation activities, however, the extraction of "domain-specific" information is rather difficult.

Regarding attack detection, the application area of e-learning does not differ much from the general case. *Profile classification* would, in an e-learning context, aim at the identification of unusual, irregular or unlikely learner profiles. Such a profile could contain facts like, for example,

- a student solving all tasks in a time deviating from the optimum drastically – in both directions,
- a student performing implausibly equally on different kinds of tasks in different areas of knowledge,
- a student performing implausibly differently of similar kinds of tasks in similar topics,
- a student whose online times don't conform to the completed tasks,
- etc.

In addition to the CF-related issues just discussed, the context of e-learning involves additional risks that have to be considered. As shortly introduced in Section 6.4.2, learners can show undesirable behaviour like "gaming the system" (see, for example, [Baker et al., 2006], [Baker et al., 2008] or [Muldner et al., 2011]).

Muldner et al. describe this kind of gaming behaviour as exploiting "properties of an instructional system to make progress while avoiding learning", based on the definition of [Baker et al., 2009] who describe gaming as "attempting to succeed in a learning environment by exploiting properties of the system rather than by learning material" . An adaptive e-learning environment must thus be able to detect or prevent student activities that are likely to be part of gaming, although detection is in this case probably more relevant than prevention.

Regarding *detection* of gaming behaviour, a description of possible manifestations of this kind of behaviour must be available. One manifestation could be that students excessively utilize a system's help functionalities before having had a look at the learning content. Given such descriptions, an e-learning system should, in order to detect gaming behaviour without much human intervention being required for the analysis of student activities, be able to autonomously analyze log data, searching for predefined behavioural patterns.

Regarding *prevention*, different interventions for discouraging gaming can be considered, for example, [Muldner et al., 2011]:

- introducing a mandatory delay before the system's help functionality can be used [Aleven, 2001], [Murray and VanLehn, 2005],
- introducing additional supplementary exercises [Baker et al., 2006],
- introducing agents pointedly reacting to gaming behaviour [Baker et al., 2006], or
- providing visualizations of student behaviour including information on gaming the system [Walonoski and Heffernan, 2006], [Arroyo et al., 2007].

Independent on what prevention strategy or combination of strategies is chosen, a subsequent analysis should be conducted in order to ensure that it does not negatively affect the learning process itself in any way.

The strategies introduced here for attack *prevention* can also be applied as reaction to an attack that has been *detected*.

Effects on the Proposed Approach

This section discusses the implications of security threats and strategies on the approach proposed in this thesis, based on information about patterns in students' learning behaviour (in this case signifying problem-solving styles) gained by clustering their activity sequences (in this case problem-solving sequences in an ITS, see Chapter 6). This information should then become a basis for individual learner and collaborative learning support, as described in Chapter 7.

Individual learner support does not require learners to be able to interact with each other. Thus it could be integrated in the ITS like Andes (which does not include collaboration or awareness facilities) directly. In this case, learners are not (or not directly) in a state of competition with each other which lowers the probability of a profile injection attack.

Collaborative learning support can only take effect if groups are available, i.e., students are in contact with other students in order to cooperate. There, a state of competition could theoretically arise, which, in comparison to a setting with only individual learner support, could increase the probability of a profile injection attack happening. However, compared to domains other than e-learning, this probability is still significantly lower.

A profile injection attack is unlikely to be successful if details about the algorithm are unknown to the attacker. However, it should, at least to a certain degree, be transparent to the user why the system behaves as it behaves (e.g., whether recommendations are based on a user's cooperation behaviour or the learning performance).

If the full algorithms and criteria for the system's decisions were known, it would be relatively easy to feed a user profile with fake data. For example, a student could, in order to receive additional hints right from the beginning, decide not to ask for hints for a while and accept a potentially bad performance meanwhile. Yet, this risk mainly applies for systems where the performance is not graded, i.e., does not affect the student's progress and assessment.

Generally, we must state that the proposed approach can be vulnerable to profile injection attacks. However, for the reasons already mentioned, profile injection attacks are not particularly likely to occur in the context of e-learning. Therefore, strategies to deal with them should concentrate on attack detection instead of attack prevention. The more likely an attack is, the more efforts should be put into attack prevention strategies (i.e., the pessimistic approach), the less likely it is, the more effort should be put into attack detection (i.e., the optimistic approach).

If an attack has been identified, there are several ways of reacting to it, including the limitation of the user's access to the system or interaction with the system. However, an alternative and probably the best way to deal with an attack would be to change the system's behaviour back to non-personalized for this user. This would not affect other learners' work with the system, and it would not keep the attacker from using the system in general, which is, in an e-learning environment, not as desirable as in, for example, an e-commerce environment.

The concrete ITS used here, does not rely on adaptive features, they could thus be considered an additional service to the students using the system that can easily be turned off without compromising the system's functionality in general and for other users.

Regarding the aspect of gaming the system, as discussed more generally for adaptive e-learning systems in the previous section, it has already been proven in Section 6.4.2 that the proposed approach is well able to detect such undesirable behaviour. Possible strategies to deal with it were described in Section 7.

In order to enhance detection of gaming behaviour, the known concrete manifestations of gaming can be described as styles with the available attributes and fed into level I clustering.

Concluding, it must be stated that security is a highly relevant issue in all phases of adaptive systems development and application process and in order to keep, for example, a recommender system reliable, several issues must be carefully dealt with. Not only must the algorithms applied be tested regarding their robustness against different kinds of attacks, but also must the location and the way of data storage, and the communication between the system and its users, be carefully planned and implemented in order to make an adaptive system as secure as possible.

8.2 Privacy, Anonymity and Pseudonymity

Personalized systems usually keep and process information about their users that falls under the scope of "personal data", according to the *EU Data Protection Directive* [EC, 1995], [Kobsa and Schreck, 2003]. [Kobsa, 2007] states that the first discussion of the tension between personalization and privacy, published by [Kobsa, 1990], did not have much impact, a trend that did not change for almost a decade. A rapid change, however, came in the late 1990s, due to the following reasons [Kobsa, 2007]:

- personalized systems moved to the web,
- more sources of user data became available,
- more powerful analyses of user data became possible, and
- restrictions imposed by privacy legislation arose.

Before this gradual revolution of personalized systems, user models were mostly bound to one stand-alone machine or a local network. Furthermore, back in the beginnings of personalized systems, user data almost exclusively consisted of data explicitly entered by the user, whereas now, the most important conclusions are drawn based on complex analyses of implicit usage data monitored by the system. These developments led to the introduction of stricter privacy laws not only concerning commercial websites but also experimental research on user modeling [Kobsa, 2007].

8.2.1 Definitions and Description

Privacy is a term that can be defined and described in multiple ways as summarized by [Kobsa and Schreck, 2003]. [Warren and Brandeis, 1890] define privacy as the "right of the individual to be let alone". [Westin, 1970] defines privacy as the right of people "to determine for themselves when, how, and to what extent information about them is communicated to others". Another definition is provided by [Posner, 1984], where privacy entails "giving people property rights in information about themselves and letting them sell those rights freely".

Regarding *anonymity*, we can distinguish between different types, according to [Gavish and Gerdes, 1998], who introduce the following classification: environmental anonymity, being determined by external factors like the number and diversity of users, content-based anonymity, meaning that users cannot be identified by data processed about them, and procedural anonymity, which is determined by the communication protocol and underlying communication layers. These types are further discussed by [Kobsa and Schreck, 2003] who claim that in order to protect users' privacy through anonymity, all three must be present in personalized systems.

In addition to these *types of anonymity*, the *degree of anonymity* can be determined by, for instance, using the "levels" introduced by [Flinn and Maurer, 1995] who distinguish between six different forms of user identification:

- *Super-identification (level 5)* means that the system must be able to uniquely identify every user and to unambiguously associate activities with the respective users. This kind of identification requires not only a valid user account, i.e., a user name and a password, but must ensure that a person's identity is evident. Either must the respective information be provided by the user directly, or by a third party organization which can guarantee the authenticity of such information. This kind of identification can be equated with zero anonymity.

- *Usual identification (level 4)* means that a user must log in to a system with a name and a password. This is the most common form of identification as it is used by most multi-user systems at the moment.

- *Latent identification (level 3)* means that a user is known as a person to the system but may create a set of pseudonyms that are mutually disjoint. This leads to the system being able to identify a user, whereas distinct users cannot directly identify others.

- *Pen-name identification (level 2)* means that a user is known to the system by a user name and needs a password to log in. However, also here, multiple pseudonyms can be used. The user does not have to be properly identifiable as a person.

- *Anonymous identification (level 1)* means that a user is identified by the system but without a name which makes the user unaddressable. The system may keep the user's log history in order to, for instance, analyze the user's behaviour and, based on that, tailor its own responses.

- *No identification (level 0)*

What kind of identification should be implemented, depends on the system's function and objectives. Most adaptive systems will, in general, require at least a way to establish a link between user activities and a user in order to offer long-term personalization.

As already explained, usage data is needed for two different purposes: first, to tailor the information provided by the system to the users individually, based on this user's activity history, and second, to be integrated in the system's general body of knowledge about its users that becomes then the basis for CF. Therefore full anonymity is almost impossible in such settings.

8.2.2 Potential Threats

[Teltzrow and Kobsa, 2004] state that adaptive systems are often useful in domains where users extensively and repeatedly use a system but may be less appropriate for infrequent users with short sessions because there the system is not able to collect enough data.

However, Teltzrow and Kobsa also point out that the collection of data causes privacy concerns. Specifically, they mention a study reported by [Culnan and Milne, 2001] showing that privacy concerns are the most important barrier for those customers who refuse to shop online, an observation that is also supported by what is described by [Pavlou, 2003].

In general, several studies have shown that people, due to privacy concerns, tend to withhold information about themselves that would be needed in order to find the optimum degree of personalization for them [Wang and Kobsa, 2007], [Kobsa, 2007], [Teltzrow and Kobsa, 2004]. Therefore, adaptive features face the challenge of finding the right balance between privacy and personalization.

Teltzrow and Kobsa present results from 30 different user privacy surveys and analyze how the results of these surveys could potentially impact different types of personalization. They distinguish between the following types of privacy aspects:

- (a) privacy of user data in general,
- (b) privacy in a commercial context,
- (c) tracking of user sessions and use of cookies,
- (d) e-mail privacy, and
- (e) privacy and personalization

The results in category (a) show that there is, in general, a significant concern about the use of personal information (for example, one survey reveals that 27% of the users would never provide personal information to a web site [Fox and Rainie, 2000]).

Regarding category (b), the results indicate that privacy may play an even more important role in commercial systems than it does in adaptive ones generally. For example, a huge part of the participants (37%, [Forrester Research, 2001]) stated they would buy more if they would not be concerned about their privacy. Results like this are reinforced by the fact that most participants stated that they want to be asked before their personal information is used.

In category (c), the results suggest that users are most uncomfortable being tracked across multiple web sites (over 90%) and that more than half of them are also worried about being tracked in general [Harris Interactive, 2000]. However, the percentage of users who generally accept cookies is relatively high too [Personalization Consortium, 2000].

Regarding category (d), results suggest that although the percentage of people who state they have been sent offensive e-mail is not too high (28%, [Fox and Rainie, 2000]), the majority of users complains about irrelevant or unsolicited e-mail ([Ipos-Reid and Columbus Group, 2001], [Cyber Dialogue, 2001]). Although these results are only marginally relevant within the scope of adaptive systems as described here because e-mail and content / navigation adaptation can be handled, to a great extent, independent from each other, they help to draw a clearer picture about acceptance of personalization on the web.

Category (e) is the most interesting one within the scope of this thesis because the studies there directly addressed the question of privacy in personalized contexts.

The results show different trends; first, more than half of the users (59%) stated they see personalization as a good thing in general, as opposed to 37% who do not [Harris Interactive, 2000]. Second, it is considered useful if a system remembers basic information that is needed more than once (e.g., an e-mail address), and third, users are sceptical about the system's intentions concerning collected data.

A different, more recent representative user survey on tailored advertising and behavioural tracking [Turow et al., 2009] reveals highly interesting results. Asked for their opinion about behavioural tracking of people on the web, 69% of the participants stated that there should be a law giving people the right to know everything a website knows about them. Even more (92%) were of the opinion that there should be a law requiring the deletion of all stored data about a user if requested by the user. 70% suggested a fine for companies that purchase or illegally use someone's information.

Additionally, almost half of the participants of that survey agree or strongly agree with the statement that "consumers have lost all control over how personal information is collected and used by companies".

The information provided by the summary of these surveys ([Teltzrow and Kobsa, 2004] and [Turow et al., 2009]) is highly relevant for further design and implementation of adaptive systems. It shows that whether or not a user trusts a system in the sense of not fearing privacy to be compromised, often correlates to how and to what extent users can actively influence how their data is dealt with, stored, and made public. However, it must be considered additionally that often there are discrepancies between users' general attitudes towards privacy and their actual behaviour [van de Garde-Perik et al., 2008].

8.2.3 Coping Strategies

Generally, privacy can be approached in different ways. [Pfitzmann and Köhntopp, 2001] identify the following aspects: anonymity, unlinkability, unobservability, and pseudonymity.

- *Privacy* is commonly considered to be strongly related to *anonymity.* Anonymity, based on [ISO 99, 1999], can be defined as the state of being not identifiable within a set of subjects[1], i.e. a user's activities cannot be linked to the respective user profile and therefore also not with the user's identity. The set of subjects is also named the "anonymity set" and consists of all subjects that might cause an action.

- *Unlinkability* of items (for instance, users' activities) means that they are no more or less related than they are related concerning a-priori knowledge [ISO 99, 1999].
- *Unobservability* means the state of an item being indistinguishable from any other item [ISO 99, 1999], i.e. a specific subject of interest cannot be distinguished from "random noise".
- *Pseudonymity* implies the use of a pseudonym as identifier. A pseudonym, in general, is an identifier of a sending or receiving object, i.e. a name other than the actual one, thus pseudonyms can also be understood as a kind of mapping between objects and identifiers. The use of a pseudonym does not automatically imply anonymity (see a more detailed discourse by [Kobsa and Schreck, 2003]). Whether pseudonyms ultimately lead to anonymity, is dependent on how the mechanism of pseudonymity is implemented. In general, pseudonyms do not only refer to single persons. [Kobsa and Schreck, 2003], for instance, introduce different types of pseudonyms: role pseudonyms, relationship pseudonyms, and role-relationship pseudonyms.

How and to what extent, personalized systems protect their users' privacy, can differ drastically. The following sections discuss different ways of approaching the privacy issue. These ways are not only ways to ensure that a user's personal privacy sphere is not intruded, but also to foster the user's trust in the system.

Privacy Laws and Regulations

Besides different approaches of protecting their users' privacy provided by the systems themselves, there are privacy policies partly independent from the systems that predetermine specific aspects to be abided by. These privacy prescriptions can be split up into

- privacy laws (mostly imposed by countries or unions of countries like the EU)
- privacy regulations imposed by industry sectors, and
- privacy regulations imposed by individual companies.

Privacy laws and regulations do not only affect the systems in that personalized features must be designed to keep them, they might well also affect the user's behaviour, if they are sufficiently well communicated.

Over 40 countries have instated their own privacy laws. The EU, for example, additionally has privacy laws that include constraints and regulations for data practices in all sectors of

[1] the set of possible subjects is dependent on the knowledge of the observer (e.g., an attacker), thus, anonymity is not an absolute state but is relative with respect to the observer

their economy. Other countries, including the U.S.A. have several different sector-specific regulations instead [Wang, 2010].

The following paragraphs provide some examples for privacy-related regulations [EC, 2002]:

> "Where the provision of a value added service requires that traffic or location data are forwarded from an electronic communications service provider to a provider of value added services, the subscribers or users to whom the data are related should also be fully informed of this forwarding before giving their consent for the processing of the data."
>
> "Systems for the provision of electronic communications networks and services should be designed to limit the amount of personal data necessary to a strict minimum. Any activities related to the provision of the electronic communications service that go beyond the transmission of a communication and the billing thereof should be based on aggregated, traffic data that cannot be related to subscribers or users. Where such activities cannot be based on aggregated data, they should be considered as value added services for which the consent of the subscriber is required."
>
> "Where location data other than traffic data, relating to users or subscribers of public communications networks or publicly available electronic communications services, can be processed, such data may only be processed when they are made anonymous, or with the consent of the users or subscribers to the extent and for the duration necessary for the provision of a value added service. The service provider must inform the users or subscribers, prior to obtaining their consent, of the type of location data other than traffic data which will be processed, of the purposes and duration of the processing and whether the data will be transmitted to a third party for the purpose of providing the value added service. Users or subscribers shall be given the possibility to withdraw their consent for the processing of location data other than traffic data at any time."

From this examples it becomes obvious that it is of crucial importance to not only communicate to the user what kind of data is used for what purposes (which will be discussed in more details later in this section) but also to ask for the user's consent.

A way to communicate privacy regulations to users would be the introduction of a standardized format used and supported by all related (adaptive) systems. Such a format could facilitate increased awareness for privacy among the users because they would have to become familiar with privacy regulations and their implications only once. However, the introduction of such a format can only have impact if it is commonly accepted and used.

Although such a standardized (and, in this case, machine-readable) format was already provided by the Platform for Privacy Preferences (P3P) [Cranor et al., 2006] many years ago, it has been practically not in use until now. For instance, [Electronic Privacy Information

Center, Junkbusters, 2000] claimed the protocol to be "complex and confusing" and to "make it more difficult for Internet users to protect their privacy" already over a decade ago, naming, among others, the following reasons:

- P3P, instead of ensuring the observance of Fair Information Practices, builds on the notice and choice privacy approach and fails to establish privacy standards.
- P3P excludes good websites lacking P3P code.
- P3P lacks any means to enforce privacy policies.
- P3P did not impress jurisdictions that have considered its use for the implementation of legal rules for privacy (e.g., the EU explicitly rejected to integrate P3P in its privacy protection framework).

More recently, [Leon et al., 2010] found errors in about a third of the websites using P3P policies they analyzed and further found thousands of the invalid policies to be identical. They identified two causes for these errors: potentially misleading practices by web administrators, and accidental mistakes. Leon et al. reason that many sites could misinterpret the policies which subsequently leads to misleading users and making privacy protection facilities ineffective.

Summing up, different analyses found the basic idea behind P3P to be good but criticise the concrete implementation.

In general, as revealed by the survey reported by [Turow et al., 2009], many users do not have the impression that existing privacy laws and regulations provide a sufficient level of privacy protection (only 54% of the participants stated they regard the privacy level as provided by existing laws reasonable). Therefore, an adaptive system must not rely on privacy laws only but has to take care of further measures, as described in the following sections.

Tailoring the Degree of Identification

Depending on the way in which personalization should be offered, different degrees of identification are possible. For instance, if personalization should be provided for the current session only, and using the information provided in this session is sufficient, anonymous identification may be used.

In other environments where, for instance, adaptive learning support should be offered, the system needs to monitor and analyze users' long-term behaviour in order to draw reasonable conclusions. If the recommendations and adaptations made do not seem reasonable to the user, the trust in the system will wane. Specifically, in personalized systems, there are two potential hazards regarding user trust; first, users may lose trust regarding their privacy and the integrity of their data, and second, a user may lose trust in the quality of personalization. Both are substantial issues within the scope of adaptive systems research.

However, another viewpoint about privacy, anonymity and pseudonymity on the web is provided by [Kobsa and Schreck, 2003] who do not only explain different types and levels of user privacy but also discuss the risks of anonymity and pseudonymity, like, for example, reduced suppression of criminal behaviour, adoption of fake identities or missing credit for contributions (see [Kobsa and Schreck, 2003], based on [Neumann, 1996] and [Gavish and Gerdes, 1998]). Still, Kobsa and Schreck encourage pseudonymous access to user adaptive systems. They argue that it

- is demanded by the users,
- hides the relation between users and their related data from the applications, and
- potentially fosters more frank interaction with the system.

Involving the User

[Kobsa, 2001] states that due to strong variations in users' preferred privacy preferences, it is not possible to provide general specifications for adaptive systems that meet all possible privacy requirements. He further argues that it is thus necessary to tailor privacy to each user, taking into account a user's preferences. The privacy issue is a very important one in the context of adaptive systems that must be carefully addressed, for example by measures like

- communicating to the users what data is collected,
- communicating to the users what this data is needed for,
- communicating to the users what kind of added value personalization means for them,
- making the user model more transparent to the users by offering ways for the users to influence their own user model, for instance, by letting them delete specific information, and
- offering a non-personalized version of the system if technically possible with reasonable efforts [Kobsa, 2001].

in order to increase acceptance of adaptive systems on the web.

Another way of better involving the user can be found in the decentralization of user models (see, for example, [Lum, 2003]) as already introduced in Section 8.1.3. Decentralization cannot only be considered a security aspect in the sense of data integrity, it can also be approached from the privacy perspective for the following reasons:

- If a user's model is stored on the own machine, the user gets a better feeling of "being in charge" of what happens with it.
- A user can decide (and actively take the measures to do so) who the model should be shared with.

- A user can, at any time, change or delete the own model if being not comfortable with it any more.

[Wang and Kobsa, 2007] discuss a different user modeling approach that dynamically selects personalization methods at run-time according to the user's individual privacy concerns. The idea is based on different personalization methods being encapsulated in different components, so that at run-time only those components compliant with the respective privacy constraints can become operational. Another idea in the area of involving the user in adaptive systems is described by [Wang and Kobsa, 2010] who provide an infrastructure that

- provides support for system designers to graphically express privacy constraints,
- allows users to set their personal privacy preferences, and
- enables run-time enforcement of privacy constraints.

The first feature mainly aims at providing a structured overview about the system components and dependencies between them on the one hand, and about the privacy constraints, the dependencies between them and their impact on the system components on the other hand. The second feature aims at demonstrating for the users the impacts their privacy settings have on the behaviour of the system.

Integrating Trust

Trust in recommender systems, and specifically in correspondence with CF techniques that recommender systems are usually based on, is discussed in detail by [O'Donovan and Smyth, 2005]. They argue that in order to obtain reliable predictions on the basis of CF, not only profile-profile similarity should be a criterion for partner selection (with partner here meaning "recommendation partner", i.e. a user whose profile can serve as a basis for recommendations), but also the trustworthiness of a partner.

O'Donovan and Smyth further reason that a recommendation partner with evident similarities with a specific user does not necessarily have to be a reliable predictor and suggest selection of "trustworthy" partners in the sense that those have a history of making reliable recommendations.

8.2.4 Implications for the Proposed Approach

This section discusses the relevance of the presented privacy threats and strategies for adaptive e-learning systems in general and for the proposed approach in particular.

Effects on Adaptive E-Learning Systems

Privacy is a crucial issue in the context of adaptive systems in general because these systems rely on the users to participate and to reveal "honest", authentic behaviour. Honest in this context means that users should behave the same way they would have if the system had not tracked them. This is, however, only the case if users trust the system regarding the administration and transmission of their data.

This also, or even particularly, applies for adaptive e-learning systems. Although e-learning systems usually do not require users to provide some kinds of information that are, for example, needed in the context of e-commerce, like banking information or a full address, there are different kinds of sensitive information users entrust to the system in this context, for instance:

- personal learning achievements,
- knowledge on specific topics and general knowledge,
- interest in specific topics,
- preferred way of learning or problem-solving, or
- speed of comprehension.

From this information, one could try to even draw conclusions about a user's complete learning profile including learning difficulties. Therefore, e-learning data has to be considered highly sensitive – a user who is not sure whether personal data could fall into the hands of for example, colleagues, trainers or employers, would strongly hesitate to interact with the system.

Thus, the privacy issue must be carefully addressed in the context of adaptive e-learning systems. Important measures to be taken can be listed as follows:

- The lowest possible level of identification that allows to link a user to activities should be chosen. Possibilities range from *anonymous identification* to *super identification*, depending on the respective system and the system's purpose. *Super identification* could be necessary in a system, provided by an educational institution, that includes assessment and grading. *Anonymous identification* can be sufficient within open online e-learning systems that are designed to be used by everyone without any obligation to pass tests or show progress.
- The users should be informed about what kind of data is collected and what it is needed for.
- The users should be given the opportunity to decide whether they want to "use" personalization features or not. Ideally, they should be provided a non-adaptive version of the system if they prefer.

- The user must be asked for consent regarding the collection, analysis and processing of personal data.
- The own user model should be made transparent to the user.
- The user should be given the opportunity to remove parts of the information stored.
- A user should be given the opportunity to decide whether personalization affects only the user individually or also potential collaboration partners. This implies that a system should, in the best case, be able to distinguish between individual learner support and collaboration support so that one can, independently from the other one, be activated or deactivated.

Regarding the last measure, we have to also consider that if adaptation affects collaboration (e.g., by a system offering group synthesis recommendations), this could reveal information about the individual user to the respective collaboration partner (i.e., lateral exposure of user model data). If the basic ideas behind grouping are transparent to the users, conclusions could be drawn about, for example, another user's learning characteristics.

Effects on the Proposed Approach

Regarding the effects of privacy issues on the proposed approach, we have to consider the following facts:

- The approach suggests individual learner support and additionally also collaboration support based on learners' activities within an ITS.
- The approach relies on user activity data collected during the user's interaction with the platform.
- The ITS used in this case is not designed to assess learners' performance or to grade it, but rather as an environment for training.
- The ITS in its current form does not allow users to cooperate – users are not aware of each other.

From these facts we can conclude that generally, the privacy concerns regarding the collection and analysis of usage data that apply for e-learning also apply for the approach. Thus, the measures to strengthen user trust as described for adaptive e-learning in general should be taken in this particular case also.

However, the environment itself (in this case, the ITS) brings along a few characteristics like the purpose behind it and its accessibility. In this particular case, the ITS is hosted by a university that might want to limit access in order to exclusively accept students that are registered at the university. If this were the case, user profiles would have to be linkable to users' real identities, or at least to their profiles on another university-internal platform.

The ITS itself, however, can be well implemented in open learning settings also, where there is no necessity of linking users' profiles to their real identity. There, a comparatively low level of identification could be chosen and pseudonymous access could be implemented without loss of personalization quality.

Furthermore, the last fact determines that only individual users' activity sequences can be logged by the system. As the resulting information is intended to be used for collaboration support also, the users should be given the opportunity to agree or disagree to the disclosure of the information.

This means that users should be able to decide whether their user model data is used as a basis for grouping, etc. However, we should in general consider providing a non-adaptive version of the system in addition to an adaptive one. In the case of the ITS discussed here, this would be implementable relatively easily.

The fact that users can decide themselves whether they want to use personalized features or not would potentially increase users' trust in the system and thus also their behaviour.

However, in systems that allow users to decide themselves whether information about them should be used as basis for adaptations or not (as suggested here), we have to consider an additional challenge. Regarding adaptive collaboration support, the system's tasks might involve grouping users. If a part of the users have prohibited user model information to become the basis for adaptation, the system must decide how to consider this regarding the groups it recommends – should these users be grouped with the ones who allowed adaptation or not? In the first case, algorithms taking into account such disparities must be found.

8.3 Summary

This chapter discussed security issues on different levels (i.e., the level of system security and the level of users' privacy). It considered not only concerns implied by adaptive systems in general but specifically addressed the domain of e-learning and the user data analysis and interpretation methods as introduced in Chapters 4, 5 and 6, but also the interventions suggested in Chapter 7.

Chapter 9

Discussion and Future Work

This chapter discusses the presented approach, compares it to other, related ones, and provides an outlook on future activities along this line of work. A brief preliminary version of the chapter also appeared as a part of [Köck and Paramythis, 2011].

9.1 Summary

In this thesis, a novel approach to adaptive learning support was described. Although the approach itself concentrates on the data analysis and interpretation phase of the adaptation cycle (see Chapters 5 and 6), a full picture of its application in adaptive e-learning systems was provided (see, for example, Chapters 1, 7 and 8). The overall aim behind what is described in this thesis is to enhance personalization in e-learning.

The analysis and interpretation of user data were discussed at different levels, starting with individual user activities monitored to form a basis for the prediction of users' future interest in specific topics, forms of content, ect. (see Chapter 4). Experiments indicated that this approach works well for the concrete aim it was designed for.

Thus, the results gained by this kind of analysis can become the foundation for content-based adaptive learning support, however, it does not take into account (a) activity-based behavioural information about the learner and (b) the connections that may exist between individual user activities. The loss of an information dimension, bearing the danger of creating inaccurate user models, is a problem that frequently arises in adaptive systems that treat user activities as independent from each other.

Therefore, the presented approach additionally particularly addressed the aspects of interrelation and sequential connectedness of activities within a user's interaction history with the system. Several different ways of modeling sequential data were described, one of which was selected to become the basis for the subsequent steps in the process.

The selected modeling approach is based on DMMs that depict a users' activities within a specific task, i.e., each DMM models one particular user's solving sequence for one particular problem (that can consist of several steps). A lot information can be read out of a specific user / problem DMM, like for instance:

- a user's attitude towards the use of help,
- a user's suspensions regarding the submission of wrong answers, or
- a user's general behaviour regarding problem-solving.

However, the DMMs are not primarily created to be interpreted by human experts but to be further processed by an unsupervised learning mechanism. The results are then prepared for interpretation and analysis of the sequential user activities. As the available information about user behaviour can differ in its level of granularity, the approach was designed to handle these different circumstances.

To establish a way to compare and verify the results for different settings, kinds of data, students and academic terms, and to show the approach's practicability, a variety of experiments was run that, on different clustering levels, demonstrated the detection of

1. predefined problem-solving styles (level I),
2. problem-solving styles along predefined learning dimensions (level II), and
3. (new) learning dimensions (level III) that again can comprise multiple problem-solving styles.

On level I, the well-known problem-solving style *Trial and Error* was identified based on students' activity sequences.

On level II, the predefined dimension of *Help-Seeking* behaviour served as a basis for the detection of different problem-solving styles within this dimension. The process succeeded in two ways: it successfully clustered for the requested dimension and it additionally identified concrete styles within it. On this level, the results of the experiments also confirmed different models of help-seeking behaviour described in the recent literature (cf. [Aleven et al., 2006] and [Baker et al., 2006]).

On level III, the main goal was to automatically detect learning dimensions (as used in a predefined way on level II) by a system-driven clustering approach. Not only was the approach able to identify dimensions autonomously, it also confirmed the predefined concrete problem-solving styles and dimensions that were selected on levels I and II.

The information gained by the different levels of clustering is then used as a basis for different kinds of adaptive support that aims at enhancing the learning process of both individual users and user groups.

9.2 Comparison

This section compares the presented approach to the most relevant efforts described before in Chapter 2.

Regarding the base data used for the experiments, [Romero and Ventura, 2007] and [Romero et al., 2008] provide examples for similar approaches. However, what is described there differs drastically from what is the core of this thesis regarding the further process and the adaptation goals. Romero and Ventura utilize classification algorithms, i.e., supervised learning, for the prediction of students' final grades, whereas here clustering, i.e., an unsupervised learning approach, is implemented.

Furthermore, the approach discussed here not only aims at detecting predefined possible outputs and measuring the probabilities for these outputs to occur, but, more importantly, at the automatic identification of significant aspects in user behaviour without falling back to fully predefined options. The way data is analyzed and further processed, reflects the difference between the two approaches.

As opposed to what is described here, where the system operates on activity sequences directly in order not to lose a dimension of information (relations and dependencies within the sequences) that is essential for pattern detection, Romero and Ventura first aggregate the sequences so that the classification process is provided an abstracted representation of the data.

Comparing the approach described here to the one introduced by [Beal et al., 2006], which does not only consider activity data but also students' self-reported motivation profiles and teachers' ratings, a main difference lies in the amount of human effort during the monitoring process; here, human effort in this phase of the process is negligible. However, the nature of the data is quite similar – both approaches include correct and incorrect attempts and help requests monitored by an ITS (Wayang Outpost [University of Massachusetts Amherst, 2010] in their case).

Yet, the further process and main objectives of the two approaches, differ. Beal et al. aim at the classification of students regarding the constellation of beliefs that they bring to the learning scenario, and to show that multiple data sources can be used integratedly in order to reach this aim, whereas here, as already explained before, the main goal is to provide ways of (semi)-automatic detection of different aspects of learning behaviour in order to subsequently adaptively support the process.

[Amershi and Conati, 2009] describe a similar approach regarding the point in time when human intervention becomes necessary – their approach as well as the one here, delays this necessity until the end of the process. This means, no intervention is needed in both cases until when behavioural patterns have already been automatically detected. There are, however, also several important differences: the system described by Amershi and Conati does not provide a clear notion of “correct” and “incorrect” behaviour, therefore, there is no immediate feedback and help from the system based on the correctness of the answers.

The approach of Amershi and Conati uses one feature vector per student that represents an aggregated version of this student’s activities, whereas here, every problem-solving sequence of a student becomes one feature vector, which results in a much higher number of vectors and more fine-grained information. However, the two approaches again resemble regarding the long-term goals (individual adaptations and guidance based on the knowledge retrieved from the models) but are applied on different levels in different environments: exploratory systems vs. ITS.

The clustering-based approach discussed by [Anaya and Boticario, 2009b] seems to be similar to the one described in this thesis at first glance. However, some significant differences can be identified after a more detailed view. For instance, the approach of Anaya and Boticario requires a considerable amount of human intervention and effort which is not always realistic to expect in real-world settings.

Furthermore, their system, similar to the one of Amershi and Conati, uses aggregated data, which results in the loss of sequential information. Finally, when comparing the approach presented here to the one of Anaya and Boticario, we can recognize different ultimate goals. Here, statistical information is used to optionally supplement with sequential data, aiming at the detection of specific types of behaviour. In their approach, the revelation of relations between statistical indicators and collaboration behaviour is the main goal.

[Beal et al., 2007] present an HMM-based pattern detection approach that again differs from the one described here in its modeling aims. Here, the relevant activity sequences consist of observable actions only, clearly suggesting a certain state configuration, which resulted in the selection of DMMs instead of HMMs for modeling the sequences. However, the clustering and prediction results of Beal et al. are highly relevant regarding the valuation of the general applicability and capabilities of the approach described here because they indicate Markov-based models to be very well suitable for the modeling and the analysis of sequential student activity data.

Finally, [Li and Yoo, 2006] use models based on Bayesian Markov Chains that are structurally comparable to the ones described here. The ultimate aim there is to support adaptive tutoring, thus, another parallel can be drawn to the approach introduced here. However, the two approaches differ regarding the modeling process and the interpretation of the outcome. Li and Yoo, based on predefined learning types, limit the number of possible models to exactly three.

Furthermore, the granularity of the outcome is limited and restricted to a very specific kind of information. As opposed to this idea, here the models are dynamically created for the students’ problem-solving sequences individually. Thus, a more fine-grained analysis is possible, which again potentially results in higher information gain.

9.3 Discussion

This section concentrates on potential challenges and implications of the proposed approach and discusses the case study described in Chapter 6. Particularly, the focus lies on the requirements involved by the transfer of the approach to different settings and application domains.

9.3.1 Study Setup and Selected Data

Regarding the nature of the base data employed in the study reported in this thesis we have to consider that

1. the problems that had to be solved by the students were relatively uniform, thus
2. students showed relatively homogeneous problem-solving strategies across different series of problems.

Particularly, all problems were from the broader area of physics, i.e., the same knowledge domain. However, if there would have been greater variability regarding the nature of the problems, this might have resulted in different strategies or categories of strategies for addressing different (categories of) kinds of problems.

This involves the risk of receiving higher distribution of student behaviours among the clusters identified by the data analysis process, which, subsequently, could make the identification of comprehensive behaviour models for individuals more difficult. In this case, adjustments of the proposed approach would become necessary, like, for example,

- segmentation of the analysis to mirror the categorization of problems or problem domains, or
- comparison of student entropies to problem entropies to determine whether an observed significantly high distribution is caused by students intentionally applying different strategies on the same kind of problems, or by problem variability.

A strategy to deal with increased variability regarding problems or problem categories, would be the explicit segmentation of the user model to allow for distinction between behavioural patterns that were applied for different settings. Alternatively one could also, in a second clustering phase, adjust the weights in the optimization formula in a way that lowers the relevance of the student entropy, so that the clusters are formed based on the other factors and accounting for instable student behaviour.

A system that is able to segment the user model would also be able to organize different sub-models, each representing the behaviour, strategies or patterns employed for one specific kind of problem. A version of the approach adjusted to deal with different kinds or categories of problems would in turn also impact subsequent adaptive behaviour. However, adaptive system interventions could still be associated with (families of) patterns, however, on a per-problem-type basis.

9.3.2 Sequence Modeling

This section discusses the selection of a Markov-based modeling approach for the purposes discussed in this thesis and for potential consideration of other scenarios and domains, as well as the identification of "episodes" in user activities, i.e., related events.

Model Selection

As shown in previous chapters, the proposed way of modeling activity sequences with DMMs is well applicable when all related activities and resulting states are observable. If the proposed approach should be applied to learning domains other than problem-solving, or even entirely different domains, several different adjustments of the Markov models might become necessary. In this case, HMMs might be the better choice, as the literature provides evidence that HMMs are well applicable in such settings.

However, before a different or adjusted kind of Markov model could be selected for a changed modeling task, further comparative work would be necessary in order to establish the relative merits of each type of model for the respective purpose. As a first step it may be advisable that researchers concentrate on the analysis of the intrinsic characteristics of the modeled activities and states.

After the selection of a particular, suitable kind of Markov model, the next step would be the decision for states (or activities, in case states are not observable) that should be represented in the model. First, the introduction of alternative aggregation levels regarding individual users' activities might be advantageous in different ways and lead to additional findings.

Second, for future work, it might be also interesting to model the activities of groups which would be a less straightforward task regarding the adjustment of the models, which could be approached, by, for example,

1. treating a group of users as one single entity, thus creating a group model including the activities of all group members, or
2. keeping modeling a group's activities from the perspective of the individual group members, but introducing into the model also "external" activities to which a member's own activities may be a response.

In the first approach, the model states would represent the collective status of the group, whereas in the second approach the individual learner would stay in the center of the model which, however, would integrate related activities of others.

A selection of a model for the representation and analysis of group work should, similar to what was stated for the analysis of individual user behaviour, be based on the analysis (here: clustering) goals. The two approaches just mentioned can serve as a first categorization of themes around which several modeling variations are possible.

Identification of Episodes

Another important decision to be made when applying the approach to a different setting, is how “episodes”, i.e., semantically related sets of activities, are delineated and how they can be distinguished from other sets of activities. In the case of the proposed approach, the boundaries between different episodes could be identified relatively easily by the given association with a problem.

However, if such information is not available, one has to find alternative ways to identify episodes. The first, most obvious criterion is the temporal aspect. Yet, whether the temporal aspect is a reliable indicator for dependencies, highly depends on the nature of activities and the tools employed.

As already described in Chapter 5, the nature of temporal relations between activities can differ drastically for different tools, which was, in Section 5.2, approached by introducing different time slots for different tools. However, the definition of a time slot for a tool cannot be static, but has to include additional information that could, for instance, be based on the analysis of the respective content.

A different possibility would be to introduce additional structures for joint work, like topics under which a chat or forum discussion takes place. In order to establish reliable episode boundaries, it might be necessary to apply multiple criteria.

Another aspect to consider regarding the proposed approach and related future work is the number of possible activity categories, which is, in the present scenario, limited, resulting in models of relatively low complexity. In spite of the limited number of activities, activity categories respectively, it can be argued that their semantics were heavily dependent on the sequences in which they were performed (for instance, how often a student requested a specific type of hint, how often a student tried to submit an answer in the meantime, whether the student changed the preferred type of hint at a specific point in time, etc.).

Regarding the line of work reported in this thesis, the main emphasis lies on the development of techniques that allow for analysis and subsequent detection of patterns in learner’s activities. The presented case study demonstrates that substantial, non-obvious results can be derived from the activity sequences *despite* the limited variability of activities / activity categories.

Thus it can be expected that, provided with more types of activities /activity categories, the approach would be able to offer even more diverse insights. Additionally, given more or richer sets of activities, these activities could be also aggregated to result in a set of types similar to the one utilized for the case study. Aggregation in this case could, for example, be done by grouping all synchronous activities of learners within a group.

However, a future step regarding the validation of the proposed approach provided more diverse activities, would further ascertain its more general applicability.

9.3.3 Metrics and Clustering Process

This section discusses the metrics used by the approach and their application in the clustering process in different scenarios and domains.

Metrics

The application of the approach to different e-learning- or other domains would potentially also entail different adjustments regarding the metrics that dynamically guide the clustering process. However, although the indices presented in Section 6.2, Average Student Entropies, Average Problem Entropies, Average Variance, and Average Expected Prediction Error, are domain-specific, they are of a sufficiently general nature to be used as a starting point.

The Average Expected Prediction Error, for example, is related to the “success” of an activity sequence and can be replaced by any measure or combination of measures that captures the semantics of desirable and undesirable effects of activity sequences in the target domain.

The Average Problem Entropies, to provide another example, also seem to be largely domain-specific because they represent the context within the activities are performed and are thus directly related to the “episodes” discussed before. The position of the “context” could well be filled by an information other than the problem as in this case though.

It must be considered, however, that the Average Expected Prediction Error and the Average Problem Entropies are in a relation of mutual influence, as one is an indicator of success that is bounded by the activities contained in the other one’s instances.

A last decision that needs to be made in this context is the empirical establishment of the weights in Equation 6.10 to match the researcher’s objectives in deriving the appropriate number of clusters for different clustering goals.

After the establishment / adjustment of the metrics, their application on the different levels introduced by the clustering process, is the next step, which requires preceding determination of the data sets that are fed into the process.

Application in the Clustering Process

The first clustering level (behavioural pattern-oriented clustering) involves the selection of the features that represent relevant states in the models. Relevant in this context means that these states result from activities that are part of the respective pattern that should be identified.

The second clustering level (dimension-driven clustering) is potentially more challenging because it requires the inclusion of all behaviour that may be related to a specific dimension. Thus, this part of the process is well implementable for learning dimensions that are well defined in the literature but much more difficult if no, only partial, erroneous or non-behavioural-related definitions are available.

Regarding the third clustering level ("open discovery" clustering), it is most challenging to establish an upper limit for the number of feature combinations that will be used in the clustering process. In the study reported here, this upper limit was determined empirically, using the results of the previous levels as a basis.

However, this approach may not be applicable under all circumstances, for instance, when the previous levels of clustering should not be run there would not be a possibility to determine the limit. In such cases, one could use an initial default value for the upper limit, for instance, a reasonable percentage of the features in the data set.

Alternatively, meta-information about the features and the relations between them could be introduced. This kind of meta-information could become the basis for decisions on what attributes should be used in what combinations (considering, for example, that semantically related attributes should not be used disjointly).

Another possibility would be to integrate a domain-independent dimension-reduction method (like, for example, PCA) in the process. This would aim at the identification of a smaller number of primary features providing a sufficient characterization of the data set.

Yet, for the scenario and purposes discussed here, such methods are only restrictedly appropriate because they typically reduce dimensionality by combining features. The resulting "new" features however, may have lost their original behavioural semantics, which subsequently complicates the interpretation of clustering results based on them.

9.3.4 Potential Extensions

The discussion above points towards a related potential limitation of the proposed approach. Considering complex behavioural models, a qualitative analysis of the results of the third level of clustering may be challenging, especially when new e-learning domains should be explored. In such cases, the identification of dimensions and related patterns might be difficult for the human observer.

This challenge could be approached by introducing additional meta-information for the features and relations, as already discussed before. This additional information could help to semantically interpret the candidate dimensions and patterns suggested by the system.

However, as already stated before, this extension might be a rather demanding one regarding its implementation. Alternatively, one could consider integrating visualized information to make the system's propositions better interpretable and understandable by human observers. Such visualizations should be able to capture the semantic relations underlying the system's propositions and facilitate judgement of their significance.

This however, presumes the existence of tools that are capable of creating visualizations that illustrate dimensions and patterns. Such tools could of course be of great assistance also if the complexity of the behavioural models analyzes is not that high – as in the study reported in this thesis.

9.4 Outlook

This section provides a concise overview on potential future activities along the line of work reported in this thesis. Generally, potential future work can follow two different directions:

1. the application of the approach on collaborative learning data, and
2. the application of the approach on data produced via different user interfaces and on different devices.

9.4.1 Collaborative Learning

As reported in this thesis, the proposed approach was applied to individual learners' activity data in order to gain information about their learning / problem-solving behaviour that can subsequently become the basis for adaptive support. "Adaptive support" in this context does not only include supporting the individual learner but also supporting collaboration based on information about the individual.

As already explained in Chapters 5 and 6, information about an individual learner's behaviour and preferred learning style can be a highly reliable source for inferences about group constellations and grouping of learners in general because the learner's behaviour is not or only marginally influenced by external factors.

External factors in this context can be, for instance, the personal relations (like friendship or competition) between users that are hidden to the system. These factors could elicit user behaviour that cannot be correctly interpreted by the system and thus potentially bias the user profile.

However, the analysis of collaboration behaviour could also reveal information that is not capturable by the exclusive analysis of individual user activities. Thus, a combination of both can potentially constitute an improvement to the current approach.

The approach in general is potentially well applicable on collaboration activity data also. Whether the resulting information is as valuable for collaboration support as expected, will have to be shown by an additional case study. For that purpose, data produced in an environment like Sakai (Data Set I as described in Section 3.2.1) can be used.

As already stated, the main objectives would slightly differ from what was described here. Here, the aim was to extract from the activity sequences of individual users information needed to (a) adaptively support the learning process of this user, and to (b) provide a basis for collaboration establishment support. The aim for the proposed further case study would be to analyze the behaviour of given groups of collaborating learners in order to adaptively support the learning process of this particular group and of groups that behave in a similar way.

The provision of this kind of support would then render the process applicable for all kinds of adaptive learning support introduced in Section 1.1.2 which would in turn demonstrate even more convincingly that the approach depicts a holistic concept including all stages of the adaptation cycle, from data acquisition to adaptive system interventions and is independent from the learning setting and environment.

On a higher level, it could additionally be shown that the approach is applicable to domains other than learning or other sub-domains within the scope of learning. This would, in both cases, involve adjustments regarding the metrics and their application in the clustering process, as discussed earlier in Section 9.3.

An e-learning domain potentially interesting to analyze would be, for example, knowledge acquisition. The "success" in this case would be, compared to the problem-solving domain, similarly easy to measure.

An example for a domain other than e-learning that could be analyzed by an adapted version of the proposed approach would be the area of e-advertising, where it would be interesting to analyze the influence of (tailored) advertisements on user's behaviour within a self-contained environment like an e-commerce platform.

9.4.2 Different Interfaces and Devices

Along a different direction, future potential for the proposed approach lies in its adoption for the analysis of data monitored during the interaction via different user interfaces and furthermore also different devices. The current trend towards touch and multi-touch devices would probably justify the efforts that would have to be put into the necessary adjustments.

An interface as provided, for example, by devices like the Microsoft Surface [Microsoft, 2011] introduces prerequisites for both individual users' problem-solving processes and for collaboration of multiple users that are significantly different to those provided by a standard web-based interface. Thus, future research efforts could involve answering the following questions:

- Does the environment (i.e., interface including specific ways of interaction) influence the problem-solving strategies of individuals?
- Does the environment influence the problem-solving strategies within collaborating groups?
- Does the environment provide a basis for entirely different problem-solving strategies of individuals and groups in general?
- Does the success of individual users' or groups' problem-solving strategies depend on prerequisites provided by the environment?
- Do the different interaction metaphors and facilities make necessary different kinds of adaptive support for both individual users and groups?

A simple web-based interface as used by most e-learning scenarios, including the one described in this thesis, provides a predefined, limited number of ways of interacting with the system, like a mouse click, scrolling, or a keyboard entry. In this case, it is easy to link these low-level events (like, for instance, a mouse click) to semantically characterized high-level activities (like, for instance, "help request").

One of the most important aspects regarding an adoption of the proposed approach to different interfaces / devices is that those allow for different kinds of user interactions, like, for example [Villamor et al., 2011]:

- flicking (i.e., brushing the surface with a fingertip),
- pinching (i.e., touching the surface with two fingers and bringing them closer together),
- spreading (i.e., touching the surface with two fingers and moving them apart), or
- rotating (i.e., touching the surface with two fingers and moving them in a clockwise or counterclockwise direction).

In addition to the interactions contained in this list, any possible gesture could be integrated as a way to communicate with the system. The latter, however, refers to interaction with a mouse also. Furthermore, many devices integrate additional ways of interaction, like the Microsoft Surface that allows interaction with real objects.

Whether a system itself makes use of these novel possibilities, is an implementation-internal aspect depending also on the purpose of the system. For instance, if a system should be exploratory like the one used in the case study reported by [Amershi and Conati, 2009], interactions as "natural" as possible might be beneficial, whereas in systems like traditional

ITSs where the purpose is to impart a very specific, well predefined portion of knowledge, this aspect might be subsidiary.

The paradigm-shift in the area of learning (see Section 1.1) might suggest a development towards increased influence of exploratory, self-explanatory environments, which in turn could soon reach adaptive systems research also.

However, the “new” user interaction possibilities lead, during an interaction process, to the production of a huge amount of low-level events that cannot be fed into the process as easily as the ones used here. The utilization of the proposed approach for this kind of data would thus necessitate the introduction of an additional preliminary step abstracting from the low-level events high-level activities, which would be a potentially rather complex task.

The results of such an analysis could on the one hand potentially show how interaction data with a standard web interface is comparable to interaction data with a technologically more advanced interface. On the other hand, the analysis could consider and potentially be able to identify not only domain-specific interaction characteristics (for example, in the domain of problem-solving), but also different interaction paradigms and patterns which can significantly contribute to research in the area of human-computer interaction, linking it to the fields of e-learning and adaptive systems.

Bibliography

[Advanced Distributed Learning Initiative, 2002] Advanced Distributed Learning Initiative (2002). Sharable Content Object Reference Model (SCORM^{TM}) Version 1.2.

[Aho et al., 1974] Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*. Addison-Wesley.

[Aiken et al., 1998] Aiken, R., Leng, P., Mühlbacher, J., Schauer, H., and Shave, M. (1998). Interactive Seminars Using the Web: An International Experience. In *Teleteaching, Distance Learning, Training and Education, Proceedings of the XV. IFIP World Computer Congress*, pages 869–875.

[Aleven, 2001] Aleven, V. (2001). Helping Students to Become Better Help Seekers: Towards Supporting Metacognition in a Cognitive Tutor. Paper Presented at German-USA Early Career Research Exchange Program: Research on Learning Technologies and Technology-Supported Education.

[Aleven et al., 2006] Aleven, V., McLaren, B., Roll, I., and Koedinger, K. (2006). Toward Meta-Cognitive Tutoring: A Model of Help Seeking With a Cognitive Tutor. *International Journal of Artificial Intelligence and Education*, 16:101–128.

[Aleven et al., 2003] Aleven, V., Stahl, E., Schworm, S., Fischer, F., and Wallace, R. (2003). Help Seeking and Help Design in Interactive Learning Environments. *Review of Educational Research*, 73(3):277–320.

[Alfonseca et al., 2006] Alfonseca, E., Carro, R. M., Martín, E., Ortigosa, A., and Paredes, P. (2006). The Impact of Learning Styles on Student Grouping for Collaborative Learning: A Case Study. *User Modeling and User-Adapted Interaction*, 16.

[Ally, 2004] Ally, M. (2004). *Theory and Practice of Online Learning*, chapter Foundations of Educational Theory for Online Learning, pages 3–32. Athabasca University.

[Amazon.com, 2010] Amazon.com (2010). Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more. http://www.amazon.com/, last accessed: September 26th, 2010.

[Amershi and Conati, 2006] Amershi, S. and Conati, C. (2006). *Intelligent Tutoring Systems*, volume 4053/2006 of *Lecture Notes in Computer Science*, chapter Automatic Recognition of Learner Groups in Exploratory Learning Environments.

[Amershi and Conati, 2009] Amershi, S. and Conati, C. (2009). Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments. 1(1).

[Anaya and Boticario, 2009a] Anaya, A. R. and Boticario, J. G. (2009a). A Data Mining Approach to Reveal Representative Collaboration Indicators in Open Collaboration Frameworks. In *Proceedings of the 2nd International Conference on Educational Data Mining (EDM09)*.

[Anaya and Boticario, 2009b] Anaya, A. R. and Boticario, J. G. (2009b). Clustering Learners according to Their Collaboration. In *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design*.

[Anaya and Boticario, 2011] Anaya, A. R. and Boticario, J. G. (2011). Content-Free Collaborative Learning Modeling Using Data Mining. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems*, 21(1).

[Anderson, 2002] Anderson, C. R. (2002). *A Machine Learning Approach to Web Personalization*. PhD thesis, University of Washington.

[Anderson and Messick, 1974] Anderson, S. and Messick, S. (1974). Social Competency in Young Children. *Development Psychology*, 10:282–293.

[Anderson, Terry, 2004] Anderson, Terry (2004). *Theory and Practice of Online Learning*, chapter Toward a Theory of Online Learning, pages 33–60. Athabasca University.

[Arroyo et al., 2007] Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S., and Woolf, B. P. (2007). Repairing Disengagement With Non-Invasive Interventions. In *Proceedings of International Conference of Artificial Intelligence in Education (AIED07)*, pages 195–202.

[Association of American Colleges, 1981] Association of American Colleges (1981). Computer-Based Learning.

[ATutor, 2010] ATutor (2010). ATutor Learning Content Management System. http://www.atutor.ca, last accessed: October 19th, 2010.

[Ayodele et al., 2010] Ayodele, T., Zhou, S., and Khusainov, R. (2010). Email Classification Using Back Propagation Technique. *International Journal of Intelligent Computing Research*, 1(1).

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press.

[Bailey, 1990] Bailey, K. (1990). Social Entropy Theory: An Overview. *Systemic Practice and Action Research*, 3(4):365–382.

[Baker, 2010] Baker, R. S. (2010). *International Encyclopedia of Education*, chapter Data Mining for Education. Elsevier, 3 edition.

[Baker et al., 2009] Baker, R. S., Corbett, A. T., and Koedinger, K. R. (2009). Educational Software Features that Encourage and Discourage "Gaming the System". In *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, pages 475–482.

[Baker et al., 2006] Baker, R. S., Corbett, A. T., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A. Z., Naim, M., Raspat, J., Baker, D. J., and Beck, J. E. (2006). *Intelligent Tutoring Systems*, volume 4953/2006 of *Lecture Notes in Computer Science*, chapter Adapting to When Students Game an Intelligent Tutoring System, pages 392–401.

[Baker et al., 2008] Baker, R. S., Corbett, A. T., Roll, I., and Koedinger, K. R. (2008). Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction*, 18(3):287–314.

[Baker and Yacef, 2009] Baker, R. S. and Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining*, 1(1).

[Bakiri and Dietterich, 2002] Bakiri, G. and Dietterich, T. G. (2002). *Data Mining Techniques in Speech Synthesis*, chapter Achieving High-Accuracy Text-to-Speech with Machine Learning. Chapman and Hall.

[Ballone and Czerniak, 2001] Ballone, L. M. and Czerniak, C. M. (2001). Teachers' Beliefs About Accommodating Students' Learning Styles In Science Classes. *Electronic Journal of Science Education*, 6(2).

[Beal et al., 2007] Beal, C., Mitra, S., and Cohen, P. (2007). Modeling Learning Patterns of Students With a Tutoring System Using Hidden Markov Models. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED)*, pages 238–245.

[Beal et al., 2006] Beal, C., Qu, L., and Lee, H. (2006). Classifying learner engagement through integration of multiple data sources. In *Proceedings of the 21st International Conference on Artificial Intelligence*, pages 151–156.

[Ben-Ari, 1998] Ben-Ari, M. (1998). Constructivism in Computer Science Education. *ACM SIGCSE Bulletin*, 30(1):257–261.

[Bengio and Frasconi, 1996] Bengio, Y. and Frasconi, P. (1996). Input-Output HMMs for Sequence Processing. *IEEE Transactions on Neural Networks*, 7(5).

[Bengoetxea, 2002] Bengoetxea, E. (2002). *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, École Nationale Supérieure des Télécommunications (Paris).

[Bienenstock and von der Malsburg, 1987] Bienenstock, E. and von der Malsburg, C. (1987). A Neural Network for Invariant Pattern Recognition. *Europhysics Letters*, 4(1):121–126.

[Black, 2004] Black, P. E. (2004). "Euclidean Distance" in Dictionary of Algorithms and Data Structures, U.S. National Institute of Standards and Technology. Accessed February 2010, available from http://www.itl.nist.gov/div897/sqg/dads/HTML/euclidndstnc.html.

[Blackboard Inc., 2010] Blackboard Inc. (2010). Blackboard International. http://www.blackboard.com, last accessed: October 19th, 2010.

[Bonk and Reynolds, 1997] Bonk, C. J. and Reynolds, T. H. (1997). *Web-Based Instruction*, chapter Learner-Centered Web Instruction for Higher-Order Thinking, Teamwork, and Apprenticeship, pages 167–178. Englewood Cliffs, NY: Educational Technology Publications.

[Borek et al., 2009] Borek, A., McLaren, B. M., Karabinos, M., and Yaron, D. (2009). *Learning in the Synergy of Multiple Disciplines*, chapter How Much Assistance Is Helpful to Students in Discovery Learning? Lecture Notes in Computer Science.

[Bottou et al., 1997] Bottou, L., Bengio, Y., and Le Cun, Y. (1997). Global Training of Document Processing Systems Using Graph Transformer Networks. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

[Bottou and Le Cun, 2005] Bottou, L. and Le Cun, Y. (2005). Graph Transformer Networks for Image Recognition. *Bulletin of the International Statistical Institute (ISI).*

[Boyle and Encarnacion, 1994] Boyle, C. and Encarnacion, A. O. (1994). Metadoc: An Adaptive Hypertext Reading System. *International Journal of User Modeling and User-Adapted Interaction*, 4(1):1–19.

[Breiman, 1996] Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24(2):123–140.

[Brown et al., 2007] Brown, E., Fisher, T., and Brailsford, T. (2007). Real Users, Real Results: Examining the Limitations of Learning Styles Within AEH. In *Proceedings of the 18th Conference on Hypertext and Hypermedia*, pages 57–66.

[Brusilovsky, 1992] Brusilovsky, P. (1992). Intelligent Tutor, Environment and Manual for Introductory Programming. *Innovations in Education & Training International.*

[Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *International Journal of User Modeling and User-Adapted Interaction*, 6(2–3):87–129.

[Brusilovsky, 2001] Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110.

[Brusilovsky, 2004] Brusilovsky, P. (2004). KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers&Posters.*

[Brusilovsky, 2007] Brusilovsky, P. (2007). Adaptive Navigation Support. In *The Adaptive Web*, Lecture Notes in Computer Science, pages 263–290. Springer-Verlag Berlin Heidelberg.

[Brusilovsky et al., 2004] Brusilovsky, P., Chavan, G., and Farzan, R. (2004). Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In *Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 24–33.

[Brusilovsky and Maybury, 2002] Brusilovsky, P. and Maybury, M. T. (2002). From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM*, 45(5).

[Brusilovsky and Millán, 2007] Brusilovsky, P. and Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In *The Adaptive Web*, Lecture Notes in Computer Science, pages 3–53. Springer-Verlag Berlin Heidelberg.

[Brusilovsky and Rizzo, 2002] Brusilovsky, P. and Rizzo, R. (2002). Map-Based Horizontal Navigation in Educational Hypertext. *Journal of Digital Information*, 3(1).

[Burke et al., 2005a] Burke, R., Mobasher, B., Bhaumik, R., and Williams, C. (2005a). Collaborative Recommendation Vulnerability to Focused Bias Injection Attacks. In *Proceedings of the Workshop on Privacy and Security Aspects of Data Mining*.

[Burke et al., 2006a] Burke, R., Mobasher, B., Williams, C., and Bhaumik, R. (2006a). Classification Features for Attack Detection in Collaborative Recommender Systems. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'06)*.

[Burke et al., 2006b] Burke, R., Mobasher, B., Williams, C., and Bhaumik, R. (2006b). Detecting Profile Injection Attacks in Collaborative Recommender Systems. In *Proceedings of the Joint Conference on E-Commerce Technology and Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2006)*.

[Burke et al., 2005b] Burke, R., Mobasher, B., Zabicki, R., and Bhaumik, R. (2005b). Identifying Attack Models for Secure Recommendation. In *Proceedings of Beyond Personalization: Workshop on the Next Generation of Recommender Systems*.

[Butler and Pinto-Zipp, 2006] Butler, T. J. and Pinto-Zipp, G. (2006). Students' Learning Styles and Their Preferences for Online Instructional Methods. *Journal of Educational Technology Systems*, 34(2).

[Cannell, 1999] Cannell, L. (1999). Review of [Distance Education] Literature. Distance Education Association of Theological Schools. Unpublished Paper.

[Carro et al., 2003a] Carro, R. M., Ortigosa, A., Martín, E., and Schlichter, J. (2003a). *Groupware: Design, Implementation, and Use*, volume 2806/2003 of *Lecture Notes in Computer Science*, chapter Dynamic Generation of Adaptive Web-Based Collaborative Courses.

[Carro et al., 2003b] Carro, R. M., Ortigosa, A., and Schlichter, J. (2003b). *Web Engineering*, volume 2722/2003 of *Lecture Notes in Computer Science*, chapter Adaptive Collaborative Web-Based Courses.

[Cassidy, 2004] Cassidy, S. (2004). Learning Styles: An Overview of Theories, Models and Measures. *Educational Psychology*, 24(4).

[CDNOW, 2010] CDNOW (2010). CDNOW: Never miss a beat. http://www.cdnow.com/, last accessed: November 20th, 2010.

[Chirita et al., 2005] Chirita, P.-A., Nejdl, W., and Zamfir, C. (2005). Preventing Shilling Attacks in Online Recommender Systems. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, pages 67–74.

[Choi and Kang, 2008] Choi, H. and Kang, M. (2008). Analyzing Learner Behaviours, Conflicting and Facilitating Factors of Online Collaborative Learning using Activity System. In *Proceedings of the 2008 International Conference on Learning Sciences.*

[Clark, 1983] Clark, R. E. (1983). Reconsidering Research on Learning from Media. *Review of Educational Research*, 53(4):445–459.

[Clark, 1994] Clark, R. E. (1994). Media and Method. *Educational Technology Research & Development*, 42(3):7–10.

[Clark, 2001] Clark, R. E. (2001). *Learning from Media: Arguments, Analysis, and Evidence*, chapter A Summary of Disagreements with the "Mere Vehicles" Argument, pages 125–136. Greenwich, CT: Information Age Publishing Inc.

[Conejo et al., 2004] Conejo, R., Guzmán, E., Millán, E., Trella, M., Péred-De-La-Cruz, J. L., and Ríos, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education*, 14:1–33.

[Conte et al., 2004] Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty Years Of Graph Matching In Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298.

[Corral et al., 2007] Corral, G., Armengol, E., Fornells, A., and Golobardes, E. (2007). Data Security Analysis Using Unsupervised Learning and Explanations. In *Innovations in Hybrid Intelligent Systems*, Advances in Soft Computing, pages 112–119. Springer-Verlag Berlin Heidelberg.

[Cranor et al., 2006] Cranor, L., Dobbs, B., Egelman, S., Hogben, G., Humphrey, J., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J., Schunter, M., Stampley, D. A., and Wenning, R. (2006). The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. http://www.w3.org/TR/P3P11/, last accessed: September 1st, 2011.

[Culnan and Milne, 2001] Culnan, M. J. and Milne, G. R. (2001). The Culnan-Milne Survey on Consumers & Online Privacy Notices: Summary of Responses. In *Interagency Public Workshop: Get Noticed: Effective Financial Privacy Notices.*

[Cyber Dialogue, 2001] Cyber Dialogue (2001). UCO Software to Address Retailers $6.2 Billion Privacy Problem. Press Release.

[Darrow, 1944] Darrow, K. K. (1944). The Concept of Entropy. *American Journal of Physics*, 12(4).

[Das et al., 2007] Das, A. S., Datar, M., Garg, A., and Rajaram, S. (2007). Google News Personalization: Scalable Online Collaborative Filtering. In *Proceedings of the 16th International Conference on World Wide Web*, pages 271–280.

[De Bra and Calvi, 1998] De Bra, P. and Calvi, L. (1998). AHA! An Open Adaptive Hypermedia Architecture. *Hypermedia*, 4(1):115–139.

[De Bra et al., 1999] De Bra, P., Houben, G.-J., and Wu, H. (1999). AHAM: A Dexter-Based Reference Model for Adaptive Hypermedia. In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia: Returning to our Diverse Roots*, pages 147–156.

[De Bra et al., 2006] De Bra, P., Smits, D., and Stash, N. (2006). The Design of AHA! In *Proceedings of the 17th Conference on Hypertext and Hypermedia*, pages 133–134.

[De Carolis et al., 1993] De Carolis, N., Pizzutilo, S., and De Rosis, F. (1993). User Tailored Hypermedia Explanations. In *Proceedings of INTRTACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems*, pages 169–170.

[Dewar and Whittington, 2000] Dewar, T. and Whittington, D. (2000). Online Learners and Their Learning Strategies. *Journal of Educational Computing Research*, 23(4).

[Dietterich, 2002] Dietterich, T. G. (2002). *Structural, Syntactic, and Statistical Pattern Recognition*, chapter Machine Learning for Sequential Data: A Review, pages 227–246. Lecture Notes in Computer Science.

[Dillenbourg, 1999] Dillenbourg, P. (1999). *Collaborative Learning: Cognitive and Computational Approaches*, chapter What Do You Mean by 'Collaborative' Learning'?, pages 1–19. Oxford: Elsevier.

[Dillenbourg et al.,] Dillenbourg, P., Blaye, A., and O'Malley, C. The Evolution of Research on Collaborative Learning. In Spada, E. and Reiman, P., editors, *Learning in Humans and Machine: Towards an Interdisciplinary Learning Science.*

[Divotkey et al., 2002] Divotkey, R., Mühlbacher, J., Reisinger, S., and Remplbauer, D. (2002). The WeLearn Distance Teaching Framework. In *Open and Distance Learning in Europe and Beyond Rethinking International Co-operation; Proceedings of the 2002 EDEN Conference*, pages 308–314.

[Du et al., 1999] Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi Tesselations: Applications and Algorithms. *Society for Industrial and Applied Mathematics Review*, 41(4):637–676.

[Dunn, 1973] Dunn, J. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57.

[EC, 1995] EC (1995). Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data. *Official Journal of the European Communities of 23 November 1995.*

[EC, 2002] EC (2002). Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector (Directive on Privacy and Electronic Communications). *Official Journal L 201.*

[Eck, 2007] Eck, D. (2007). Autotagging Music Using Supervised Machine Learning. In *Proceedings of the International Conference on Music Information Retrieval.*

[Eirinaki and Vazirgiannis, 2003] Eirinaki, M. and Vazirgiannis, M. (2003). Web Mining for Personalization. *ACM Transactions on Internet Technology*, 3(1):1–27.

[Electronic Privacy Information Center, Junkbusters, 2000] Electronic Privacy Information Center, Junkbusters (2000). Pretty Poor Privacy: An Assessment of P3P and Internet Privacy. http://epic.org/reports/prettypoorprivacy.html, last accessed: September 28th, 2011.

[Encarnaçao, 1995] Encarnaçao, J. L. (1995). Adaptivity in Graphical User Interfaces: An Experimental Framework. *Computers & Graphics*, 19(6):873–884.

[Endsley, 1995] Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64.

[Ezzaidi and Rouat, 2006] Ezzaidi, H. and Rouat, J. (2006). Automatic Musical Genre Classification Using Divergence and Average Information Measures. *Engineering and Technology*, 15.

[Faber, 1994] Faber, V. (1994). Clustering and the Continuous k-Means Algorithm. *Los Alamos Science*, 22:138–144.

[Facebook, 2010] Facebook (2010). Facebook. http://www.facebook.com/, last accessed: September 26th, 2010.

[Fahy, 2004] Fahy, P. J. (2004). *Theory and Practice of Online Learning*, chapter Media Characteristics and Online Learning Technology, pages 137–174. Athabasca University.

[Farzan and Brusilovsky, 2005] Farzan, R. and Brusilovsky, P. (2005). Social Navigation Support Through Annotation-Based Group Modeling. In *Proceedings of the 10th International Conference on User Modeling*, pages 463–472.

[Fawcett and Provost, 1997] Fawcett, T. and Provost, F. (1997). Adaptive Fraud Detection. *Data Mining and Knowledge Discovery*, 1(3).

[Felder and Brent, 2005] Felder, R. M. and Brent, R. (2005). Understanding Student Differences. *Journal of Engineering Education.*

[Felder and Silverman, 1988] Felder, R. M. and Silverman, L. K. (1988). Learning and Teaching Styles in Engineering Education. *Journal of Engineering Education*, 78(7):674–681.

[Fleming, 1987] Fleming, M. L. (1987). *Instructional Technology Formations*, chapter Displays and Communication, pages 233–260. Hillsdale, NY:Erlbaum.

[Flinn and Maurer, 1995] Flinn, B. and Maurer, H. (1995). Levels of Anonymity. *Journal of Universal Computer Science*, 1(1):35–47.

[Food.com, 2010] Food.com (2010). Food.com. http://www.food.com/, last accessed: November 20th, 2010.

[Forgy, 1965] Forgy, E. (1965). Cluster Analysis of Multivariate Data: Efficiency Versus Interpretability of Classifications. *Biometrics*, 21:768–769.

[Forrester Research, 2001] Forrester Research (2001). Privacy Issues Inhibit Online Spending (Survey Summary).

[Fox and Rainie, 2000] Fox, S. and Rainie, L. (2000). Trust and Privacy Online: Why Americans Want to Rewrite the Rules. Pew Internet & American Life Project.

[Freund and Schapire, 1996] Freund, Y. and Schapire, R. E. (1996). Experiments With a New Boosting Algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Series of Books in the Mathematical Sciences. W.H.Freeman & Co Ltd.

[Gavish and Gerdes, 1998] Gavish, B. and Gerdes, J. H. (1998). Anonymous Mechanisms in Group Decision Support Systems Communication. *Decision Support Systems*, 23(4):297–328.

[Ghahramani, 2004] Ghahramani, Z. (2004). Unsupervised Learning. In *Advanced Lectures on Machine Learning*, Lecture Notes in Computer Science, pages 72–112. Springer.

[Goldman and Zhou, 2000] Goldman, S. and Zhou, Y. (2000). Enhancing Supervised Learning with Unlabeled Data. In *Proceedings of the 17th International Conference on Machine Learning*, pages 327–334. Morgan Kaufmann.

[Google News, 2010] Google News (2010). Google News. http://news.google.com, last accessed: November 28th, 2010.

[Goren-Bar et al., 2000] Goren-Bar, D., Kuflik, T., and Lev, D. (2000). Supervised Learning for Automatic Classification of Documents Using Self-Organizing Maps. In *Proceedings of the DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries 2000*.

[Graham, 2005] Graham, C. R. (2005). *Handbook of Blended Learning*, chapter Blended Learning Systems: Definition, Current Trends, and Future Directions, pages 3–21. Pfeiffer Publishing.

[Gray, 2009] Gray, R. M. (2009). *Entropy and Information Theory*. Springer New York.

[Haddad et al., 2005] Haddad, W. M., Chellaboina, V., and Nersesov, S. G. (2005). *Thermodynamics: A Dynamical Systems Approach*. Princeton Series in Applied Mathematics. Princeton University Press.

[Hall et al., 2009] Hall, M., Eibe, F., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1).

[Hämäläinen et al., 2004] Hämäläinen, W., Suhonen, J., Sutinen, E., and Toivonen, H. (2004). Data Mining in Personalizing Distance Education Courses. In *Proceedings of the 21st ICDE World Conference on Open Learning and Distance Education*, pages 18–21.

[Harmon and Jones, 1999] Harmon, S. W. and Jones, M. G. (1999). The Five Levels of Web Use in Education: Factors to Consider in Planning Online Courses. *Educational Technology*.

[Harris Interactive, 2000] Harris Interactive (2000). A Survey of Consumer Privacy Attitudes and Behaviours.

[Hathaway and Bezdek, 1988] Hathaway, Richard, J. and Bezdek, J. C. (1988). Recent Convergence Results for the Fuzzy c-Means Clustering Algorithms. *Journal of Classification*, 5:237–247.

[Hauger, 2008] Hauger, D. (2008). Fine-Grained User Models by Means of Asynchronous Web Technologies. In *ABIS 2008 - Adaptivity and User Modeling in Interactive Systems*, pages 17–19.

[Hauger, 2009a] Hauger, D. (2009a). Enhancing User Models by Client-Side User Monitoring. In *UMAP 2009: Doctoral Consortium*.

[Hauger, 2009b] Hauger, D. (2009b). Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems. In *Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009)*.

[Hauger and Köck, 2007] Hauger, D. and Köck, M. (2007). State of the Art of Adaptivity in E-Learning Platforms. In *Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems*.

[Hauger et al., 2010] Hauger, D., Paramythis, A., and Weibelzahl, S. (2010). Your Browser is Watching You: Dynamically Deducing User Gaze from Interaction Data. In *UMAP 2010 - Adjunct Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*, pages 10–12.

[Hauger and van Velsen, 2009] Hauger, D. and van Velsen, L. (2009). Analyzing Client-Side Interactions to Determine Reading Behaviour. In *ABIS 09 - Adaptivity and User Modeling in Interactive Systems*.

[Heeter, 1999] Heeter, C. (1999). Technology Enhanced Learning. White Paper for the Internet 2 Sociotechnical Summit.

[Heisele, 2003] Heisele, B. (2003). Visual Object Recognition with Supervised Learning. *IEEE Intelligent Systems*, 18(3).

[Henze, 2000] Henze, N. (2000). *Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources*. PhD thesis, University of Hannover.

[Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of thr 22nd ACM Conference on Research and Development in Information Retrieval*.

[Herlocker et al., 2000] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining Collaborative Filtering Recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Coperative Work*, pages 241–250.

[Hlaoui and Wang, 2002] Hlaoui, A. and Wang, S. (2002). A New Algorithm for Inexact Graph Matching. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02)*, volume 4, Washington, DC, USA. IEEE Computer Society.

[Höök, 1997] Höök, K. (1997). Evaluating the Utitlity and Usability of an Adaptive Hypermedia System. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 179–186.

[Hopcroft and Wong, 1974] Hopcroft, J. E. and Wong, J. (1974). Linear Time Algorithm for Isomorphism of Planar Graphs. In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing*, pages 172–184.

[Howard and Huckvale, 2005] Howard, I. S. and Huckvale, M. A. (2005). Training a Vocal Tract Synthesizer to Imitate Speech Using Distal Supervised Learning. In *Proceedings of the 10th International Conference on Speech and Computer 2005*.

[Hughes, 2004] Hughes, J. A. (2004). *Theory and Practice of Online Learning*, chapter Supporting the Online Learner, pages 367–384. Athabasca University.

[IEEE 1484.12.1-2002, 2002] IEEE 1484.12.1-2002 (2002). Draft Standard for Learning Object Metadata. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, last accessed: June 2nd, 2011.

[IEEE Learning Technology Standardization Committee, 2001] IEEE Learning Technology Standardization Committee (2001). LTSC PAPI. http://www.cen-ltso.net/main.aspx?put=230, last accessed: June 2nd, 2011.

[imc Advanced Learning Solutions, 2010] imc Advanced Learning Solutions (2010). CLIX. http://www.im-c.de/de/produkte/learning-management-system, last accessed: October 19th, 2010.

[IMS Global Learning Consortium, 2001] IMS Global Learning Consortium (2001). IMS Learner Information Package Specification. http://www.imsglobal.org/profiles/, last accessed: June 2nd, 2011.

[IMS Global Learning Consortium, 2003a] IMS Global Learning Consortium (2003a). IMS Content Packaging Specification. http://www.imsglobal.org/content/packaging/, last accessed: June 2nd, 2011.

[IMS Global Learning Consortium, 2003b] IMS Global Learning Consortium (2003b). IMS Question & Test Interoperability Specification. http://www.imsglobal.org/question/, last accessed: June 2nd, 2011.

[IMS Global Learning Consortium, 2003c] IMS Global Learning Consortium (2003c). IMS Simple Sequencing Specification. http://www.imsglobal.org/simplesequencing/, last accessed: June 2nd, 2011.

[IMS Global Learning Consortium, 2011] IMS Global Learning Consortium (2011). Learning Design Specification. http://www.imsglobal.org/learningdesign/, last accessed: March 20th, 2011.

[Ipos-Reid and Columbus Group, 2001] Ipos-Reid and Columbus Group (2001). Privacy Policies Critical to Online Consumer Trust. Canadian Inter@ctive Reid Report.

[ISO 99, 1999] ISO 99 (1999). IS 15408.

[Jain et al., 1999] Jain, A., Murty, M., and Flynn, P. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3).

[Jarvis, 2005] Jarvis, M. (2005). *The Psychology of Effective Learning and Teaching*. Nelson Thornes.

[Jeong and Biswas, 2008] Jeong, H. and Biswas, G. (2008). Mining Student Behaviour Models in Learning-by-Teaching Environments. In *Proceedings of the 1st International Conference on Educational Data Mining*, pages 127–136.

[Jin and Mobasher, 2003] Jin, X. and Mobasher, B. (2003). Using Semantic Similarity to Enhance Item-Based Collaborative Filtering. In *Proceedings of the 2nd IASTED International Conference on Information and Knowledge Sharing*, pages 176–181.

[Jung et al., 2005] Jung, S. Y., Hong, J.-H., and Kim, T.-S. (2005). A Statistical Model for User Preference. *IEEE Transactions on Knowledge and Data Engineering*, 17(6).

[Kanninen, 2008] Kanninen, E. (2008). Learning Styles and E-Learning. Master's thesis, Tampere University of Technology.

[Kanungo et al., 2002] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7).

[Kay and Kummerfeld, 1994] Kay, J. and Kummerfeld, R. J. (1994). An Individualised Course for the C Progamming Language. In *Proceedings of the Second International WWW Conference '94 Mosaic and the Web.*

[Knutov et al., 2009] Knutov, E., De Bra, P., and Pechenizkiy, M. (2009). AH 12 Years Later: A Comprehensive Survey of Adaptive Hypermedia and Multimedia. *New Review of Hypermedia and Multimedia*, 15(1):5–38.

[Kobsa, 1990] Kobsa, A. (1990). User Modeling in Dialog Systems: Potentials and Hazards. *AI & Society*, 4(3):214–240.

[Kobsa, 2001] Kobsa, A. (2001). Tailoring Privacy to Users' Needs. In *Proceedings of the 8th International Conference on User Modeling*, pages 303–313. Invited Keynote.

[Kobsa, 2007] Kobsa, A. (2007). *The Adaptive Web*, chapter Privacy-Enhanced Web Personalization, pages 628–670. Lecture Notes in Computer Science. Springer Berlin/Heidelberg.

[Kobsa et al., 1994] Kobsa, A., Müller, D., and Nill, A. (1994). KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-MS. In *Proceedings of the 4th International Conference on User Modeling*, pages 99–105.

[Kobsa and Schreck, 2003] Kobsa, A. and Schreck, J. (2003). Privacy Through Pseudonymity in User-Adaptive Systems. *ACM Transactions on Internet Technology (TOIT)*, 3(2):149–183.

[Köck, 2009] Köck, M. (2009). Towards Intelligent Adaptive E-Learning Systems – Machine Learning for Learner Activity Classification. In *ABIS 09 – Adaptivity and User Modeling in Interactive Systems.*

[Köck and Paramythis, 2010] Köck, M. and Paramythis, A. (2010). Towards Adaptive Learning Support on the Basis of Behavioural Patterns in Learning Activity Sequences. In *Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS 2010)*, pages 100–107.

[Köck and Paramythis, 2011] Köck, M. and Paramythis, A. (2011). Activity Sequence Modelling and Dynamic Clustering for Personalized E-Learning. *User Modeling and User-Adapted Interaction*, 21(1):51–97.

[Koedinger et al., 2010] Koedinger, K., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., and Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining.*

[Koedinger et al., 2008] Koedinger, K., Cunningham, K., Skogsholm, A., and Leber, B. (2008). An Open Repository and Analysis Tool for Fine-Grained, Longitudinal Learner Data. In *Educational Data Mining 2008: 1st International Conference on Educational Data Mining*.

[Koedinger and Aleven, 2007] Koedinger, K. R. and Aleven, V. (2007). Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. *Educational Psychology Review*, 19(3).

[Kohonen, 2001] Kohonen, T. (2001). *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg New York.

[Kolb, 1984] Kolb, D. (1984). *Experiential Learning*. Prentice Hall.

[König, 2010] König (2010). Towards Improved Support for Adaptive Collaboration Scripting in IMS LD. In *Proceedings of the 5th European Conference on Technology Enhanced Learning 2010 (EC-TEL)*, pages 197–212.

[König and Paramythis, 2010a] König, F. and Paramythis, A. (2010a). Closing the Circle: IMS LD Extensions for Advanced Adaptive Collaboration Support. In *Proceedings of the 2nd International Workshop on Adaptive Systems for Collaborative Learning in the Context of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010)*, pages 421–426.

[König and Paramythis, 2010b] König, F. and Paramythis, A. (2010b). Collaboration Contexts, Services, Events and Actions: Four Steps Closer to Adaptive Collaboration Support in IMS LD. In *Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010)*, pages 145–152.

[König et al., 2009] König, F., van Velsen, L., and Paramythis, A. (2009). Finding *My* Needle in the Haystack: Effective Personalized Re-Ranking of Search Results in Prospector. In *Proceedings of the 10th International Conference on E-Commerce and Web Technologies (EC-Web 2009)*, pages 312–323.

[Kozma, 1991] Kozma, R. B. (1991). Learning with Media. *Review of Educational Research*, 61(2):179–211.

[Kozma, 2001] Kozma, R. B. (2001). *Learning from Media: Arguments, Analysis, and Evidence*, chapter Counterpoint Theory of "Learning with Media", pages 137–178. Greenwich, CT: Information Age Publishing Inc.

[Lackinger and Mühlbacher, 1984] Lackinger, H. and Mühlbacher, J. (1984). Lernen mit Bildschirmtext: dezentral und zeitlich ungebunden. In *Tagungsband "Mikroelektronik für den Menschen"*, page 731ff.

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In

Proceedings of the Eighteenth International Conference on Machine Learning, pages 282–289.

[Lam and Riedl, 2004] Lam, S. K. and Riedl, J. (2004). Shilling Recommender Systems for Fun and Profit. In *Proceedings of the 13th International Conference on World Wide Web*, pages 393–402.

[Lambert, 2002] Lambert, F. L. (2002). Disorder – A Cracked Crutch for Supporting Entropy Discussions. *Journal of Chemical Education*, 79(2).

[Last.fm, 2010] Last.fm (2010). Last.fm. http://www.lastfm.de/, last accessed: Septemner 26th, 2010.

[Lefrancois, 2006] Lefrancois, G. R. (2006). *Psychologie des Lernens*, volume 4. Springer, Heidelberg.

[Leon et al., 2010] Leon, P. G., Cranor, L. F., McDonald, A. M., and McGuire, R. (2010). Token Attempt: The Misrepresentation of Website Privacy Policies Through the Misuse of P3P Compact Policy Tokens (CMU-Cylab-10-014). Technical report, Carnegie Mellon University Research Showcase. http://repository.cmu.edu/cylab/73/, last accessed: September 28th, 2011.

[Li and Yoo, 2006] Li, C. and Yoo, J. (2006). Modeling Student Online Learning Using Clustering. In *ACM-SE 44: Proc. of the 44th Annual Southeast Regional Conference*, pages 186–191. ACM.

[Li and Zaïane, 2004] Li, J. and Zaïane, O. R. (2004). Combining Usage, Content, and Structure Data to Improve Web Site Recommendation. In *Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web)*, pages 305–315.

[Lichtenwalter et al., 2009] Lichtenwalter, R., Lichtenwalter, K., and Chawla, N. V. (2009). Applying Learning Algorithms to Music Generation. In *Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI 2009)*, pages 483–502.

[Lindsay,] Lindsay, B. G. *Mixture Models: Theory, Geometry and Applications., series= NSF-CBMS Regional Conference Series in Probability and Statistics, volume=5, year = 1995, publisher = Institute of Mathematical Statistics, Hayward.*

[Liu and Dean, 1999] Liu, Y. and Dean, G. (1999). Cognitive Styles and Distance Education. *Online Journal of Distance Learning Administration*, 2(3).

[Loidl-Reisinger, 2006] Loidl-Reisinger, S. (2006). Towards Pervasive Learning: WeLearn.Mobile - A CPS Package Viewer for Handhelds. *Journal of Network and Computer Applications*, 29(4):277–293.

[Loidl-Reisinger and Paramythis, 2003] Loidl-Reisinger, S. and Paramythis, A. (2003). Distance Education - A Battlefield for Standards. In *The Quality Dialogue. Integrating Quality*

Cultures in Flexible, Distance and eLearning; Proceedings of the 2003 EDEN Annual Conference, pages 89–94.

[.LRN, 2010] .LRN (2010). .LRN. www.dotlrn.org.

[.LRN, 2010] .LRN (2010). .LRN Learn, Research, Network. http://www.dotlrn.org/, last accessed: October 19th, 2010.

[Lum, 2003] Lum, A. W. K. (2003). Scrutable User Models in Decentralized Adaptive Systems. In *Proceedings of the 9th International Conference on User Modelling*, pages 426–428.

[Lum, 2007] Lum, A. W. K. (2007). *Light-Weight Ontologies for Scrutable User Modelling*. PhD thesis, University of Sydney.

[Mandel et al., 2006] Mandel, M. I., Poliner, G. E., and Ellis, D. P. (2006). Support Vector Machine Active Learning for Music Retrieval. *Multimedia Systems*, 12(1):3–13.

[Martínez and Kak, 2001] Martínez, A. M. and Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Knowledge and Data Engineering*, 23(2).

[McBurney et al., 2008] McBurney, S., Papadopoulou, E., Taylor, N., and Williams, H. (2008). Adapting Pervasive Environments through Machine Learning and Dynamic Personalization. In *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications*, pages 395–402.

[McCallum et al., 2000] McCallum, A., Freitag, D., and Pereira, F. C. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598.

[McGreal and Michael, 2004] McGreal, R. and Michael, E. (2004). *Theory and Practice of Online Learning*, chapter Technologies of Online Learning (e-Learning), pages 115–137. Athabasca University.

[McKay and Fujinaga, 2009] McKay, C. and Fujinaga, I. (2009). jMIR: Tools for Automatic Music Classification. In *Proceedings of the International Computer Music Conference*.

[Microsoft, 2011] Microsoft (2011). Microsoft Surface. http://www.microsoft.com/surface, last accessed: September 7th, 2011.

[Mobasher, 2005] Mobasher, B. (2005). *Practical Handbook of Internet Computing*, chapter Web Usage Mining and Personalization. CRC Press.

[Mobasher et al., 2007a] Mobasher, B., Burke, R., and Bhaumik, R. (2007a). Attacks and Remedies in Collaborative Recommendation. *IEEE Intelligent Systems*, 22(3):56–63.

[Mobasher et al., 2005] Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2005). Effective Attack Models for Shilling Item-Based Collaborative Filtering Systems. In *Proceedings of the 2005 WebKDD Workshop, held in conjunction with ACM SIGKDD'2005*.

[Mobasher et al., 2007b] Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2007b). Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness. *ACM Transactions on Internet Technology*, 7(4).

[Mobasher et al., 2006] Mobasher, B., Burke, R., Williams, C., and Bhaumik, R. (2006). Analysis and Detection of Segment-Focused Attacks Against Collabosative Recommendation. In *Proceedings of the 2005 WebKDD Workshop.*

[Mobasher et al., 2004] Mobasher, B., Jin, X., and Zhou, Y. (2004). Semantically Enhanced Collaborative Filtering on the Web. In *Web Mining: From Web to Semantic Web*, Lecture Notes in Computer Science, pages 57–76. Springer-Verlag Berlin Heidelberg.

[Moodle, 2010] Moodle (2010). Moodle.org: Open-Source Community-Based Tools for Learning. www.moodle.org.

[Morris and Trivedi, 2009] Morris, B. and Trivedi, M. (2009). Unsupervised Learning of Motion Patterns of Rear Surrounding Vehicles. *IEEE International Conference on Vehicular Eletronics and Safety (ICVES)*, pages 80–85.

[movielens, 2010] movielens (2010). movielens - movie recommendations. http://movielens.umn.edu/, last accessed: September 26th, 2010.

[Mühlbacher, 1998] Mühlbacher, J. (1998). Das Ende des Präsenzunterrichts? *Österreichische Zeitschrift für Berufsbildung*, pages 8–11.

[Mühlbacher et al., 2002] Mühlbacher, J., Mühlbacher, S. C., and Reisinger, S. (2002). Learning Arrangements and Settings for Distance Teaching / Coaching / Learning: Best Practice Report. In Hofer, C. and Chroust, G., editors, *IDIMT - 2002: 10th Interdisciplinary Information Management Talks*, pages 243–253. Universitätsverlag Rudolf Trauner.

[Mühlbacher and Putzinger, 2006] Mühlbacher, J. and Putzinger, A. (2006). *Die WeLearn Plattform.* Universitätsverlag Rudolf Trauner.

[Muldner et al., 2011] Muldner, K., Burleson, W., Van e Sande, B., and VanLehn, K. (2011). An Analysis of Students' Gaming Behaviors in an Intelligent Tutoring System: Predictors and Impacts. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems*, 21(1).

[Murray and VanLehn, 2005] Murray, R. C. and VanLehn, K. (2005). Effects of Dissuading Unnecessary Help Requests While Providing Proactive Help. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED05)*, pages 887–889.

[Nelson-Le Gall, 1981] Nelson-Le Gall, S. (1981). Help-Seeking: An Understudied Problem-Solving Skill in Children. *Development Review*, 1:224–246.

[Nelson-Le Gall, 1985] Nelson-Le Gall, S. (1985). Help Seeking Behaviour in Learning. *Review of Research in Education*, 12:55–90.

[Neumann, 1996] Neumann, P. G. (1996). Inside Risks: Risks of Anonymity. *ACM Communications*, 39(12):162.

[Newman, 1994] Newman, R. (1994). *Self-Regulation of Learning and Performance: Issues and Educational Applications*, chapter Adaptive Help Seeking: A Strategy of Self-Regulated Learning, pages 283–301. Hillsdale, NY:Erlbaum.

[O'Donovan and Smyth, 2005] O'Donovan, J. and Smyth, B. (2005). Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174.

[OECD, 1992] OECD (1992). Guidelines for the Security of Information Systems. Technical report, Organization for Economic Cooperation and Development (OECD).

[OLAT, 2011] OLAT (2011). OLAT – Your Open Source LMS. www.olat.org.

[O'Mahony et al., 2004] O'Mahony, M., Hurley, N., Kushmerick, N., and Silvestre, G. (2004). Collaborative Recommendation: A Robustness Analysis. *ACM Transactions on Internet Technology*, 4(4).

[OpenACS, 2010] OpenACS (2010). Open Archicetcure Community System. http://openacs.org/, last accessed: October 20th, 2010.

[Opitz and Maclin, 1999] Opitz, D. and Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11(1999):169–198.

[O'Reilly, 2005] O'Reilly, T. (2005). What Is Web 2.0. http://oreilly.com/web2/archive/what-is-web-20.html, last accessed: March 13th, 2011.

[Ortiz et al., 2011] Ortiz, J., García Olaya, A., and Borrajo, D. (2011). A Dynamic Sliding Window Approach for Activity Recognition. In *Proceedings of User Modeling, Adaptation and Personalization (UMAP2011)*, pages 219–230.

[Osguthorpe and Graham, 2003] Osguthorpe, R. T. and Graham, C. R. (2003). Blended Learning Systems: Definitions and Directions. *Quarterly Review of Distance Education*, 4(3):227–234.

[O'Sullivan et al., 2002] O'Sullivan, M. P., Wilson, D. C., and Smyth, B. (2002). Improving Case-Based Recommendation: A Collaborative Filtering Approach. In *Proceedings of the Sixth European Conference on Case Based Reasoning*, page 278 ff.

[Pandora, 2010] Pandora (2010). Pandora Internet Radio. http://www.pandora.com/, last accessed: September 26th, 2010.

[Paramythis, 2008] Paramythis, A. (2008). Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet? In *Proceedings of the Workshop on Adaptive Collaboration Support, in conjunction with the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'08)*, pages 17–29.

[Paramythis, 2009] Paramythis, A. (2009). *Adaptive Systems: Development, Evaluation and Evolution.* PhD thesis, Johannes Kepler University Linz.

[Paramythis and Cristea, 2008] Paramythis, A. and Cristea, A. (2008). Towards Adaptation Languages for Adaptive Collaborative Learning Support. In *Proceedings of the Workshop on 'Individual and Group Adaptation in Collaborative Learning Environments', held in conjunciton with the Third European Conference on Technology Enhanced Learning Maastricht (ECTEL 2008).*

[Paramythis et al., 2008] Paramythis, A., König, F., Schwendtner, C., and van Velsen, L. (2008). Using Thematic Ontologies for User- and Group-Based Adaptive Personalization in Web Searching. In *Proceedings of the 6th International Workshop on Adaptive Multimedia Retrieval.*

[Paramythis and Loidl-Reisinger, 2003] Paramythis, A. and Loidl-Reisinger, S. (2003). Adaptive Learning Environments and e-Learning Standards. In *Proceedings of the 2nd European Conference on e-Learning*, pages 369–379.

[Paramythis and Mühlbacher, 2008] Paramythis, A. and Mühlbacher, J. (2008). Towards New Approaches in Adaptive Support for Collaborative E-Learning. In *Proceedings of the 11th IASTED International Conference on Computers and Advanced Technology in Education (CATE 2008).*

[Paramythis and van Velsen, 2009] Paramythis, A. and van Velsen, L. (2009). Lessons Learned from the Evaluations of an Adaptive Meta-Search Engine. In *Proceedings of the „Users' Preferences Regarding Intelligent User Interfaces: Differences Among Users and Changes Over Time" Workshop, held in conjunction with the 2009 International Conference on Intelligent User Interfaces (IUI 2009).*

[Paris and Gespass, 2001] Paris, C. and Gespass, S. (2001). Examining the Mismatch Between Learner-Centered Teaching and Teacher-Centered Supervision. *Journal of Teacher Education*, 52(5):398–412.

[Pavlou, 2003] Pavlou, P. A. (2003). Consumer Acceptance of Electronic Commerce – Integrating Trust and Risk with the Technology Acceptance Model. *International Journal of Electronic Commerce*, 7(3):101–134.

[Pawlak, 2000] Pawlak, Z. (2000). Rough Sets, Decision Algorithms and Bayes' Theorem. *European Journal of Operational Research*, 136:181–189.

[Perera et al., 2009] Perera, D., Kay, J., Koprinska, I., Yacef, K., and Zaïne, O. R. (2009). Clustering and Sequential Pattern Mining of Online Collaborative Learning Data. *IEEE Transactions on Knowledge and Data Engineering.*

[Personalization Consortium, 2000] Personalization Consortium (2000). Personalization % Privacy Survey.

[Pfitzmann and Köhntopp, 2001] Pfitzmann, A. and Köhntopp, M. (2001). Anonymity, Unobservability, and Pseudonymity: A Proposal for Terminology. In *Anonymity 2000*, volume 2009 of *Lecture Notes in Computer Science*. Springer-Verlag.

[Piaget, 1954] Piaget, J. (1954). *The Construction of Reality in the Child*. New York: Basic Books.

[Popescu, 2009] Popescu, E. (2009). Learning Styles and Behavioral Differences in Web-based Learning Settings. In *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*.

[Posner, 1984] Posner, R. A. (1984). *Philosophical Dimensions of Privacy*, chapter An Economic Theory of Privacy, pages 333–345. Cambridge University Press.

[Putzinger, 2007a] Putzinger, A. (2007a). Towards Asynchronous Adaptive Hypermedia: An Unobtrusive Generic Help System. In *Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2007), held in the context of Lernen-Wissensentdeckung-Adaptivität 2007 (LWA 2007)*, pages 24–26.

[Putzinger, 2007b] Putzinger, A. (2007b). Upcoming Privacy Issues in Asynchronous Adaptive Hypermedia Systems. In Hoyer, C., Chroust, G., and Doucek, P., editors, *IDIMT - 2007: 15th Interdisciplinary Information Management Talks Linz*. Universitätsverlag Rudolf Trauner.

[Putzinger, 2008] Putzinger, A. (2008). *Asynchronitätsuntersuchungen in adaptiven Hypermedia Systemen*. PhD thesis, Johannes Kepler University Linz.

[Putzinger and Szedmina, 2006] Putzinger, A. and Szedmina, L. (2006). The Application of 'WeLearn' in Language and Teaching. In *Proceedings of the 4th Serbian-Hungarian Joint Symposium on Intelligent Systems (SISY 2006)*, pages 527–538.

[Quian and Sejnowski, 1988] Quian, N. and Sejnowski, T. J. (1988). Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology*.

[Quignard and Baker, 1999] Quignard, M. and Baker, M. (1999). Favouring Modellable Computer-Mediated Argumentative Dialogue in Collaborative Problem-Colving Situtations. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education*, pages 129–136.

[Rabiner, 1989] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, pages 257–286.

[Razzaq and Heffernan, 2009] Razzaq, L. M. and Heffernan, N. T. (2009). To Tutor or Not to Tutor: That is the Question. In *Proceedings of the 2009 Artificial Intelligence in Education Conference*, pages 457–464.

[Reay, 2001] Reay, J. (2001). Blended Learning – a Fusion for the Future. *Knowledge Management Review*, 4(3).

[Richmond and Cummings, 2005] Richmond, A. S. and Cummings, R. (2005). Implementing Kolb's Learning Styles into Online Distance Education. *International Journal of Technology in Teaching and Learning*, 1(1).

[Romero and Ventura, 2006] Romero, C. and Ventura, S., editors (2006). *Data Mining in E-Learning*, volume 4 of *Advances in Management Information.* WITPress.com.

[Romero and Ventura, 2007] Romero, C. and Ventura, S. (2007). Educational Data Mining: A Survey from 1995 to 2005. *Expert Systems with Applications*, 33:135–146.

[Romero et al., 2008] Romero, C., Ventura, S., Espejo, P. G., and Hervás, C. (2008). Data Mining Algorithms to Classify Students. In *Proceedings of the 1st International Conference on Educational Data Mining (EDM08).*

[Romero et al., 2007] Romero, C., Ventura, S., and García, E. (2007). Data Mining in Course Management Systems. *Computers & Education*, 51:368–384.

[Rooney, 2003] Rooney, J. (2003). Blended Learning Opportunities to Enhance Educational Programming and Meetings. *Association Management*, 55(5):26–32.

[Rossett, 2002] Rossett, A. (2002). *The ASTD E-Learning Handbook. Best Practices, Strategies, and Case Studies for an Emerging Field.* McGraw-Hill.

[Rummel and Krämer, 2010] Rummel, N. and Krämer, N. (2010). Computer-Supported Instructional Communication: A Multidisciplinary Account of Relevant Factors. *Educational Psychology Review.*

[Ryan et al., 2001] Ryan, A., Pintrich, P., and Midgley, C. (2001). Avoiding Seeking Help in the Classroom: Who and Why? *Educational Psychology Review*, 13:93–114.

[Sakai, 2010] Sakai (2010). http://www.sakaiproject.org.

[Salas et al., 2005] Salas, E., Sims, D. E., and Burke, C. (2005). Is There a "Big Fiveïn Teamwork? *Small Group Research*, 36(5):555–599.

[Salton, Gerard, 1968] Salton, Gerard (1968). *Automatic Information Organization and Retrieval.* McGraw-Hill.

[Sanfeliu and Fu, 1983] Sanfeliu, A. and Fu, K. (1983). A Distance Measure Between Attributed Relational Graphs for Pattern Analysis. *IEEE Transactions on SMC*, 13(3).

[Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295.

[Schaller et al., 2009] Schaller, D. T., Borun, M., Allison-Bunnell, S., and Chambers, M. (2009). One Size Does Not Fit All: Learning Style, Play, and On-line Interactives. In *Proceedings of Museums and the Web 2007 - the International Conference for Culture and Heritage On-Line.*

[Schramm, 1977] Schramm, W. (1977). *Big Media, Little Media: Tools and Technologies for Instruction.* Bevery Hills, CA: Sage.

[Schreck, 2001] Schreck, J. (2001). *Security and Privacy in User Modeling.* PhD thesis, Universität Essen.

[Schwendtner et al., 2006] Schwendtner, C., König, F., and Paramythis, A. (2006). Prospector: An Adaptive Front-End to the Google Search Engine. In *Proceedings of the 14th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS 2006), held in the context of Lernen-Wissensentdeckung-Adaptivität 2006 (LWA 2006)*, pages 56–61.

[Seipold and Pachler, 2011] Seipold, J. and Pachler, N. (2011). Evaluating Mobile Learning Practice. *MedienPädagogik – Zeitschrift für Theorie und Praxis der Medienbildung*, 19.

[Sejnowski and Rosenberg, 1987] Sejnowski, T. J. and Rosenberg, C. R. (1987). Parallel Networks That Learn to Pronounce English Text. *Journal of Complex Systems*, 1(1).

[Seymore et al., 1999] Seymore, K., McCallum, A., and Rosenfeld, R. (1999). Learning Hidden Markov Model Structure for Information Extraction. In *Proceedings of the AAAI 99 Workshop on Machine Learning for Information Extraction*, pages 37–42.

[Shannon, 1948] Shannon, C. E. (1948). A Mathematical Theory of Communication. In *Key Papers in the Development of Information Theory.*

[Shavlik and Dietterich, 1990] Shavlik, J. W. and Dietterich, T. G. (1990). *Readings in Machine Learning*, pages 1–11. Morgan, Michael B.

[Simon, 1983] Simon, H. A. (1983). *Machine Learning. An Artificial Intelligence Approach*, chapter Why Should Machines Learn?, pages 25–38. Morgan Kaufmann.

[Simon, 2000] Simon, S. J. (2000). The Relationship of Learning Style and Training Method to End-User Computer Satisfaction and Computer Use: A Structural Equation Model. *Information Technology, Learning, and Performance Journal*, 18(1).

[Singh and Reed, 2001] Singh, H. and Reed, C. (2001). A White Paper: Achieving Success with Blended Learning. Centra Software.

[Soller, 2007] Soller, A. (2007). Adaptive Support for Distributed Collaboration. In *The Adaptive Web.*

[Soller and Lesgold, 2007] Soller, A. and Lesgold, A. (2007). Modeling the Process of Collaborative Learning . In *The Role of Technology in CSCL.*

[Soller et al., 2005] Soller, A., Martínez, A., Jermann, P., and Muehlenbrock, M. (2005). From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education*, pages 261–290.

[Sonntag, 2003] Sonntag, M. (2003). Online Learning Platforms and E-Government. In Galindo, F. and Traunmüller, R., editors, *E-Government: Legal, Technical and Pedagogical Aspects*, pages 283–299. Seminario de Informatica y Derecho.

[Sonntag, 2006] Sonntag, M. (2006). Syntax and Semantics of Learning Object Metadata - The IEEE/IMS LOM and Beyond. In Koohang, A. and Harman, K., editors, *Learning Objects: Theory Praxis, Issues, and Trends*. IS Press.

[Sonntag and Paramythis, 2008] Sonntag, M. and Paramythis, A. (2008). Adaptive Feedback for Legal E-Learning. In *Proceedings of the Special Track 'Computer-Based Knowledge & Skill Assessment and Feedback in Learning Settings' at the International Conference on Computer-Aided Learning*.

[Srivastava et al., 2000] Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. (2000). Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2).

[Steeples and Jones, 2002] Steeples, C. and Jones, C., editors (2002). *Networked Learning: Perspectives and Issues*. Springer, London.

[Styer, 2000] Styer, D. F. (2000). Insight into Entropy. *American Journal of Physics*, 68(12).

[Su et al., 2011] Su, J.-M., Tseng, S.-S., Chen, C.-H., and Lin, H.-Y. (2011). A Personalized Learning Content Adaption Mechanism to Meet Diverse User Needs in Mobile Learning Environments. *International Journal on User Modeling and User-Adapted Interaction. Special Issue on Data Mining for Personalized Educational Systems*, 21(1).

[Suh and Lee, 2006] Suh, H.-J. and Lee, S.-W. (2006). Collaborative Learning Agent for Promoting Group Interaction. *ETRI Journal*, 28(4):461–474.

[Summers, 1997] Summers, R. (1997). *Secure Computing, Threats and Safeguards*. McGraw-Hill.

[Svensson, 2001] Svensson, M. (2001). E-Learning Standards and Technical Specifications. Technical report, Luvit AB.

[Teltzrow and Kobsa, 2004] Teltzrow, M. and Kobsa, A. (2004). *Designing Personalized User Experiences for eCommerce*, chapter Impacts of User Privacy Preferences on Personalized Systems, pages 1–18. Kluwer Academic Publishers.

[Terrell, 2005] Terrell, S. R. (2005). Supporting Different Learning Styles in an Online Learning Environment: Does it Really Matter in the Long Run? *Online Journal of Distance Learning Administration*, 8(2).

[Thorndike, 1903] Thorndike, E. L. (1903). *Educational Psychology*. Kessinger Publishing .

[Tribus and McIrvine, 1971] Tribus, M. and McIrvine, E. C. (1971). Energy and Information. *Scientific American*, 225(3):179–188.

[Tsai and Fu, 1979] Tsai, W. and Fu, K. (1979). Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis. *IEEE Transactions on SMC*, 9(12).

[Turow et al., 2009] Turow, J., King, J., Hoofnagle, C. J., Bleakley, A., and Hennessy, M. (2009). Americans Reject Tailored Advertising and Three Activities that Enable It. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1478214, last accessed: September 1st, 2011.

[Ullman, 1976] Ullman, J. (1976). An Algorithm for Subgraph Isomorphism. *Journal of the ACM*.

[University of Massachusetts Amherst, 2010] University of Massachusetts Amherst (2010). Wayang Outpost. www.wayangoutpost.com, last accessed: September 5th, 2011.

[Vail et al., 2007] Vail, D. L., Veloso, M. M., and Lafferty, J. D. (2007). Conditional Random Fields for Activity Recognition. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*.

[van de Garde-Perik et al., 2008] van de Garde-Perik, E., Markopoulos, P., de Ruyter, B., Eggen, B., and Ijsselsteijn, W. (2008). Investigating Privacy Attitudes and Behavior in Relation to Personalization. *Social Science Computer Review*, 26(1):20–43.

[van Velsen et al., 2009] van Velsen, L., König, F., and Paramythis, A. (2009). Assessing the Effectiveness and Usability of Personalized Internet Search through a Longitudinal Evaluation. In *Proceedings of the Sixth Workshop on User-Centred Design and Evaluation of Adaptive Systems, held in conjunction with the International Conference on User Modeling, Adaptation, and Personalization (UMAP2009)*, pages 44–53.

[VanLehn et al., 2005] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R. H., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *International Journal of Artificial Intelligence in Education*, 15(3).

[Varlamis and Apostolakis, 2006] Varlamis, I. and Apostolakis, I. (2006). The Present and Future of Standards for E-Learning Technologies. *Interdisciplinary Journal of Knowledge and Learning Objects*, 2.

[Varlamis et al., 2005] Varlamis, I., Apostolakis, I., and Karatza, M. (2005). A Framework for Monitoring the Unsupervised Educational Process and Adapting the Content and Activities to Students' Needs. In *Web Information Systems Engineering (WISE) Workshops 2005*, pages 124–133.

[Vesanto and Alhoniemi, 2000] Vesanto, J. and Alhoniemi, E. (2000). Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 11(3):586–600.

[Vialardi et al., 2009] Vialardi, C., Bravo, J., Shafti, L., and Ortigosa, A. (2009). Recommendations in Higher Education Using Data Mining Techniques. In *Proceedings of the 2nd International Conference on Educational Data Mining (EDM09)*, pages 190–199.

[Villamor et al., 2011] Villamor, C., Willis, D., and Wroblewski, L. (2011). Touch Gestures Reference Guide. http://www.lukew.com/touch/TouchGestureGuide.pdf, last accessed: September 7th, 2011.

[Walker et al., 2009] Walker, E., Rummel, N., and Koedinger, K. R. (2009). CTRL: A Research Framework for Providing Adaptive Collaborative Learning Support. *User Modeling and User-Adapted Interaction*, 19(5):387–431.

[Walonoski and Heffernan, 2006] Walonoski, J. A. and Heffernan, N. T. (2006). Detection and Analysis of Off-Task Gaming Behavior in Intelligent Tutoring Systems. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, pages 382–391.

[Wang, 2010] Wang, Y. (2010). *A Framework for Privacy-Enhanced Personalization.* PhD thesis, University of California, Irvine.

[Wang and Kobsa, 2007] Wang, Y. and Kobsa, A. (2007). Respecting Users' Individual Privacy Constraints in Web Personalization. In Conati, C., McCoy, K., and Paliouras, G., editors, *User Modeling 2007*, Lecture Notes in Computer Science, pages 157–166. Springer-Verlag Berlin Heidelberg.

[Wang and Kobsa, 2010] Wang, Y. and Kobsa, A. (2010). A Generic Privacy-Enhancing Personalization Infrastructure. In *Adjunct Proceedings of the 18th International Conference on User Modeling, Adaptation and Personalization*, pages 61–63.

[Warren and Brandeis, 1890] Warren, S. and Brandeis, L. (1890). The Right to Privacy. *Harvard Law Review*, 4(5):193–220.

[Weber et al., 2001] Weber, G., Kuhl, H.-C., and Weibelzahl, S. (2001). Developing Adaptive Internet Based Courses with the Authoring System NetCoach. In *Proceedings of the 3rd Workshop on Adaptive Hypermedia*, pages 226–238.

[Weimer, 2002] Weimer, M. (2002). *Learner-Centered Teaching. Five Key Changes to Practice.* Jossey-Bass, John Wiley & Sons.

[Weld, 2002] Weld, J. (2002). *Learning Science and Science of Learning*, chapter Learner-Centered Teaching. NSTA Press.

[Westin, 1970] Westin, A. F. (1970). *Privacy and Freedom.* The Bodley Head Ltd.

[Widmer, 2006] Widmer, G. (2006). Guest Editorial: Machine Learning in and for Music. *Machine Learning*, 65:343–346.

[Wolpert, 1992] Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, 5(2):241–259.

[Wright, 1970] Wright, P. (1970). Entropy and Disorder. *Contemporary Physics*, 11(6):581–588.

[Yoo et al., 2005] Yoo, J., Li, C., and Pettey, C. (2005). Adaptive Teaching Strategy for Online Learning. In *IUI '05: Proc. of the 10th Int. Conf. on Intelligent User Interfaces*. ACM.

[Ypma et al., 2007] Ypma, A., Özer, S., van der Werf, E., and de Vries, B. (2007). Bayesian Feature Selection for Hearing Aid Personalization. In *Proceedings of the 2007 IEEE Workshop on Machine Learning for Signal Processing*, pages 425–430.

[Zancanaro et al., 2007] Zancanaro, M., Kuflik, T., Boger, Z., Goren-Bar, D., and Goldwasser, D. (2007). Analyzing Museum Visitor's Behavior Patterns. In *User Modeling 2007*, Lecture Notes in Computer Science, pages 238–246. Springer-Verlag Berlin Heidelberg.

[Zhou and Xu, 2010] Zhou, M. and Xu, Y. (2010). Injecting Pedagogical Constraints into Sequential Learning Pattern Mining. In *Proceedings of the 10th IEEE International Conference on Advanced Learning Technologies*.

[Zhu, 2008] Zhu, X. (2008). Semi-Supervised Learning Literature Survey. Technical report, University of Wisconsin – Madison.

[Zimmerman et al., 2004] Zimmerman, J., Kurapati, K., Buczak, A. L., Schaffer, D., Gutta, S., and Martino, J. (2004). *Personalized Digital Television: Targeting Programs to Individual Viewers*, chapter TV Personalization System: Design of a TV Show Recommender Engine and Interface, pages 27–52. Kluwer.

Appendix A: Software

Part of the software developed within the scope of this thesis and the ASCOLLA project, the *ActivitySequenceAnalyzer*, has been made available for download under http://ascolla.org/results/software/activity-analysis.dot.

Description

The *ActivitySequenceAnalyzer* tool covers the overall process depicted in Figure 6.1 except for the “Identification of Constraints for Creation of Data Set” part of the third level of clustering. Thus, it starts from the raw data (Data Set II as described in Section 3.2.2) and creates DMM-based models of the students’ activities. In a next step, the models are serialized and processed by a clustering algorithm. The clustering process aims at the automated detection of different problem-solving dimensions and styles within student behaviour. The process can be run with different numbers of clusters and determines the most promising cluster setting by the computation of different quality metrics. The result file provides information about these metrics and the concrete content of the different clusters in order to have it further analyzed by a human expert.

Data

Raw data processable by *ActivitySequenceAnalyzer* is supposed to be in the form as exported from PSLC DataShop [Koedinger et al., 2010] and contains content as can be found in the test file pslc_testdata.txt included in the download. The first line presents the types of information that are expected in the data. Data must strictly follow this scheme, different parts of information must not be switched, a tab must be used to separate the parts from each other. A line break denotes a new data instance.

The tool determines related sequences in the raw data that belong to the same student / problem combination and converts the related data to Markov models. The models are later extended by additional statistical information like the number of attempts a student needed so solve a task. The resulting files of this step are in ARFF format and contain both attributes retrieved from the Markov models and attributes resulting from the additional statistical information. The clustering process then runs on this data and produces an output file which contains the final results for this clustering run. The attributes derived from the data and stored in the models are predefined. However, the user can chose to eliminate attributes for the clustering which is done via specific settings in the configuration file.

How to Use

The ActivitySequenceAnalyzer.zip file contains a build directory which again contains an executable: asa.bat. A configuration file is needed to start the process. This file must either be placed in the same directory as the .bat file and be named analysis.config or be provided as a parameter when the .bat is started. A sample configuration file is provided at http://ascolla.org/results/software/activity-analysis.dot.

Appendix B: Curriculum Vitae

Personal Information

Name:	Mirjam Augstein (maiden name Köck)
Date/Place of Birth:	April 29th, 1983, Vienna, Austria
Address:	Seeweg 6, 4040 Linz, Austria
eMail:	mirjam.augstein@fh-hagenberg.at

Education

Present	**PhD Candidate in Computer Science** Johannes Kepler University, Linz, Austria
July 2006	**Diploma in Engineering for Computer-Based Learning (CBL)** Thesis title: „Computer-Supported Cooperative Project Management with Particular Emphasis on its Application in Education" University of Applied Sciences, Hagenberg, Austria

Affiliations

September 2010 - present	**Assistant Professor for "Collaborative Systems"** Degree Programme Communication and Knowledge Media Upper Austria University of Applied Sciences, School of Informatics/Communications/Media, Hagenberg, Austria
October 2006 - August 2010	**Research Staff** FIM – Institute for Information Processing and Microprocessor Technology Johannes Kepler University, Linz, Austria
October 2007- July 2010	**Academic Staff (Part-Time)** Degree Programmes Communication and Knowledge Media and Engineering for Computer-Based Learning Upper Austria University of Applied Sciences, School of Informatics/Communications/Media, Hagenberg, Austria

Teaching

- **TEL 3: Telekooperation**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2010W, 2011W
- **MTP1: Medientechnik und –praxis**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2010W, 2011W
- **WHM1: Web und Hypermedia**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2011W
- **ISY 4: Verteilte Informationssysteme**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2008S, 2009S, 2010S, 2011S

- **AUT 5: Multimediadesign und –authoring**, University of Applied Sciences, Degree Programme Engineering for Computer-Based Learning, Hagenberg, Austria
  2007W
- **AUT 2: Multimediadesign und –authoring**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2009S, 2010S, 2011S
- **PRO 1: Einführung in die Programmierung**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2011W
- **SP1/2: Studienprojekt 1/2**, University of Applied Sciences, Degree Programme Communication and Knowledge Media, Hagenberg, Austria
  2008S, 2010W, 2011S, 2011W

Participation in Research and Development Projects

- ATLab, "Assistive Technology Lab, Providing ePad Access to People with Disabilities" (FFG eCall No 2360344), 2011-2014
- ASCOLLA, "Adaptive Support for Collaborative E-Learning" (FWF P20260-N15), 2008–2010
- ALS, "Adaptive Learning Spaces" (229714-CP-1-2006-1-MINERVA-M), 2006–2009

Publications

M. Köck, A. Paramythis - Activity Sequence Modeling and Dynamic Clustering for Personalized E-Learning – International Journal of User Modeling and User-Adapted Interaction, Vol. 21, No. 1, 2011, pp 51-97

M. Köck, A. Paramythis - Towards Adaptive Learning Support on the Basis of Behavioural Patterns in Learning Activity Sequences – Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS 2010), Thessaloniki, Greece, 2010, pp 100-107

M. Köck - Towards Intelligent Adaptive E-Learning Systems - Machine Learning for Learner Activity Classification – Proceedings of the 17th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS2009), Darmstadt, Germany, 2009, pp 26-31

M. Köck - Computational Intelligence for Communication and Cooperation Guidance in Adaptive E-Learning Systems – Proceedings of the 16th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS2008), Würzburg, Germany, 2008, pp 32-34

D. Hauger, M.Köck - State of the Art of Adaptivity in E-Learning Platforms – Proceedings of the 15th Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS2007), Halle/Saale, Germany, 2007, pp 355-360

M.Köck - Computer-Supported Cooperative Project Management with Particular Emphasis on its Application in Education – Diploma Thesis, Upper Austria University of Applied Sciences, Hagenberg, Austria, 2006

Other Scientific Activities

Program Chair

- ABIS09, 17th Workshop on Adaptivity and User Modeling in Interactive Systems

Member of Program Committee and Reviewer

- EDM11, 4th International Conference on Educational Data Mining (PC/Reviewer)
- INCOS10, 2nd International Conference on Intelligent Networking and Collaborative Systems (Reviewer)
- ABIS09, 17th Workshop on Adaptivity and User Modeling in Interactive Systems (PC/Reviewer)

Charter Member

- International Educational Data Mining Society

Appendix C: Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Des weiteren versichere ich, dass ich diese Dissertation weder im In- noch im Ausland in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Linz, 5. November, 2011 Mirjam Augstein