Department of Computer Science
Series of Publications A
Report A-2003-5

Considering Individual Differences in Computer-Supported Special and Elementary Education

Jaakko Kurhila

University of Helsinki
Finland

Department of Computer Science
Series of Publications A
Report A-2003-5

Considering Individual Differences in Computer-Supported Special and Elementary Education

Jaakko Kurhila

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XII, University Main Building, on June 2nd, 2003, at 12 o'clock noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O.Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 44441

ISSN 1238-8645
ISBN 952-10-1201-3 (paperback)
ISBN 952-10-1202-1 (PDF)
Computing Reviews (1998) Classification: K.3.1, K4.2
Helsinki 2003
Helsinki University Printing House

Considering Individual Differences in Computer-Supported Special and Elementary Education

Jaakko Kurhila

Department of Computer Science
P.O. Box 26, FIN-00014 University of Helsinki, Finland
jaakko.kurhila@cs.helsinki.fi
http://www.cs.helsinki.fi/u/kurhila/

PhD Thesis, Series of Publications A, Report A-2003-5
Helsinki, May 2003, 135 pages
ISSN 1238-8645
ISBN 952-10-1201-3 (paperback)
ISBN 952-10-1202-1 (PDF)

Abstract

Special education, particularly the education of disabled children, suffers from the lack of computer-science-oriented research and moderate computer expertise in educational software used in classrooms. Special education provides a particularly challenging research area to computers and education, since every learner is unique.

Successful learning systems for special education can benefit from three distinctive properties: adaptation to the individual learning process, domain independence in the learning content with ease-of-authoring, and support for special needs. The thesis presents a model, called a learning space model, as a basis for a learning system that tries to address these three issues. The model is based on structuring the learning material in a n-dimensional vector space. The author of the material can specify the dimensions used.

The primary target group for the learning space model is children with deficiencies in mental programming. When simplified, mental programming means the ability to compose a problem solving strategy, fluency in solving various tasks, and the ability to uphold attention and motivation. Although deficiencies in mental programming are most severe with brain damage or occur often with developmental disabilities, it is clear that these deficiencies are present to some extent in every one of us.

Therefore, the learning space model is taken out into two classrooms and tested empirically. The first test is in a special education setting, but the transfer to non-disabled education is tested in elementary education. The findings from these two case studies imply that the model operates as expected if the learning material is authored carefully.

Lastly, the properties of the model are inspected formally to understand the limitations, challenges and potential of the model better.

Computing Reviews (1998) Categories and Subject Descriptors:
K.3.1 Computers Uses in Education: Computer-assisted instruction (CAI)
K.4.2 Social Issues: Assistive technologies for persons with disabilities

General Terms:
Human Factors

Additional Key Words and Phrases:
Individualization, computer-aided learning environments, intelligent tutoring systems, adaptive educational hypermedia, special education, mental programming

Acknowledgements

I am deeple grateful to both of my advisors, professors Erkki Sutinen and Matti Mäkelä. Distance and "wrong" affiliation did not make Erkki forget the responsibility to guide me through this dissertation. Countless inspiring discussions from the long roads of Indiana to the home of Erkki and Päivi were irreplaceable. Matti Mäkelä is an example of a true scholar; his broadmindedness encouraged me to pursue the work outside the core of computer science. I am also thankful to professors Henry Tirri and Jukka Paakki who helped me take the final steps on this learning path. The feedback from professor Peter Brusilovsky and Dr Michael Joy to the manuscript was most helpful.

It is often said that a dissertation would not have seen the daylight without the help of other people. This time, it is especially true. Many are those who made this dissertation; sometimes I felt like a by-stander trying to keep up. Five student groups and a total of twenty-eight students worked on the topic during the early years of this thesis as well as experts at the Department of Teacher Education, the Department of Computer Science and the Ruskeasuo School. The contributions of the students, especially MSc Pekka Uronen, were crucial during the early stages, when the only guideline was a vaguely-defined need from the Ruskeasuo School. At the later stages, EdLic Matti Lattu innovated and organised issues related to the evaluation. In addition, I want to express my gratitude to neuropsychologist Tuula Eriksson, Dr Anu Pietilä, professor Jose Maria Turull Torres, Ms Hanna Varjola, MA Leena Paasu and MSc Tei Laine. It was a pleasure to work with you.

There are many others who ought to be acknowledged: those special learners at the Ruskeasuo School, along with special teachers Pirjo Turunen, Maarit Karinen, and Jukka Laine; teachers Pia Häkli and Johanna Sollo and their enthusiastic classes of young learners at the Porolahti Comprehensive School; neuropsychologist Pekka Räsänen at the University of Jyväskylä; Matti Luukkainen, Pasi Porkka, Jan Lindström, Kjell Lemström for very meaningful discussions outside the thesis; my current research companions Miikka Miettinen and Petri Nokelainen; Marina Kurtén for her help in correcting the grammar; and all the fine people at the department I have been privileged to work with.

I am indebted to my family and close relatives, in particular my parents Pallas and Pekka and my grandmother Kaarina, for their support. The sincerest gratitude, however, goes to my long-time muse Jenni Kivelä.

Part of the research has been financially supported by the Helsinki Graduate School in Computer Science and Engineering (HeCSE), Helsingin Sanomat and the University of Helsinki. The support is gratefully appreciated.

Contents

1 Introduction **1**
1.1 Motivation for the research 1
1.2 Research issues 3
1.3 Structure of the thesis 5

2 Terminology related to the thesis **7**
2.1 Hypertext, hypermedia and hyperspace 7
2.2 Learning with computers 8
2.3 Adaptation in computer-supported education 9
2.4 Disabilities related to the thesis 10

3 Review of educational software **13**
3.1 Motivation 13
3.2 Educational software paradigms 15
3.3 Desired properties in learning systems 19
3.3.1 Adaptation to individual learning processes 19
3.3.2 Openness in learning content 23
3.3.3 Support for special needs 28
3.4 Classification of desirable properties in learning systems . . 30

4 Learning space model **35**
4.1 Definition of learning space 35
4.2 Characteristics of the learning space from the authoring point-of-view 42
4.3 Discussion of the schema 44
4.4 Learning space model in relation to tutoring systems 45
4.5 Description of AHMED 49

5 Learning materials for the learning space model **55**
5.1 Types of learning material suitable for learning spaces . . . 55
5.1.1 Frame-based computer-assisted instruction 55

5.1.2 Educational hypermedia 56
5.1.3 Adaptive educational hypermedia 57
5.1.4 Learning through games 61
5.1.5 An example: building a simulator using learning seeds 62
5.2 Characteristics of learning material in the learning space model 64
5.3 Learning material to support brain deficits 65
5.3.1 Luria's model of working brain 65
5.3.2 Attracting lost attention 67
5.3.3 Offering appropriate modalities 69
5.3.4 Supporting the mental programming 69
5.4 Testing the learners' cognitive abilities 75

6 Empirical studies 77
6.1 About the studies 77
6.2 Test space for number word sequence skills 79
6.2.1 Study setting 79
6.2.2 Test results 85
6.2.3 About the results 87
6.3 Learning space for the addition algorithm 89
6.3.1 Study setting 89
6.3.2 Test results 94
6.3.3 About the results 98
6.4 Evaluating the learning space model with potential material authors 98
6.4.1 Study setting 98
6.4.2 Test results 101
6.4.3 About the results 104

7 Learning spaces as vector spaces 105
7.1 Aspects of the concept of the learning space 105
7.1.1 Basic definitions 106
7.1.2 Formalization of a learning space as a vector space . 107
7.1.3 Learning spaces as metric spaces 109
7.2 Partial order for the dimensions 110
7.3 Discussion 111

8 Conclusion 113

Chapter 1

Introduction

1.1 Motivation for the research

Computer-aided learning has been subjected to intense research for decades. Therefore, it is remarkable that special education, particularly the education of disabled children, still suffers from the lack of computer-science-oriented research and moderate computer expertise in existing educational software. In fact, there have been only a few serious efforts to exploit the state-of-the-art computer science methods and techniques to advance special education (see Klaus et al. (1996), Edwards et al. (1998) or Miesenberger et al. (2002) for the *lack* of examples). Basic learning programs are available, but helpful and widely applicable learning environments do not exist. Special education provides a challenging research area to computer-aided learning, since every learner is, in the broadest sense, unique.

When targeted to special needs, educational software is usually for visually or hearing impaired people (see e.g. Buaud et al. (2002) and Archambault & Burger (2002)). There is also software for motorically impaired, but cognitive impairments (e.g. people with learning difficulties) are rarely addressed. The most attention has been paid to the design of alternative user interfaces, following the tradition of assistive technology. On the other hand, the *inside* of educational software, i.e., the pedagogically sound content and its technologically advanced implementation, is often forgotten (Eriksson et al. 1997).

Reasons for the *status quo* are obvious: visual, motor and hearing impairments are more common and easier to address. They are clear and straightforward to isolate, and the design characteristics of the software are largely those concerning the user interface. Cognitive and learning disabilities are more vague. There is no consensus of reasons for or even the manifestation of various cognitive disabilities. To make things even more complex, very often people with learning disabilities have multiple impairments. Therefore, educational software and computer-aided learning environments for the cognitive-impaired persons must include support for the visually, motor and hearing impaired. In some cases, one can have the possibility to use external screen readers and one-switch scanning-input devices, but this is not standard.

Can one use those numerous adaptive learning environments that have been researched and produced for non-disabled education in special education? In general, the answer is no. The reasons are three-fold, related to the user interface, the learning content, and the processing of the learning content (i.e. the storyboard of the program). In addition to the need of extra-ordinary input and output (Edwards 1995), the content of a learning program or the topic of education in a learning environment is usually special. There is no need for a Lisp course in the special education curriculum but the need for educating children to handle everyday life is essential.

Moreover, not only the topic must differ from standard educational software. The style of learning cannot be similar to that for non-disabled education. As an example, let us assume a person cannot formulate a problem-solving strategy for a simple task. In that case, the learning environment should partition the task into subtasks, so that the learner is led to the final goal step-by-step. These kinds of requirements result in novel solutions: the emphasis is no more on the quantity of information, but the way it is processed and presented to the learner. In order to deal with cognitive disabilities, the software solutions related to the user interface, the content and its processing must cooperate with each other and human cognition for a consistent learning experience. The challenges are deeper than finding another way for human-computer interaction.

There is only little empirical evidence that adaptation in computer-aided education can actually enhance learning compared to "static" tutoring systems or educational hypermedia (Brusilovsky & Eklund 1998*b*). However, this argument is without any relevance when computer-aided education is addressed to learners with special needs. Information technology can be their only means of communication and self-expression. Therefore, information technology is also essential in order to facilitate learning. Admittedly, the question whether computer-aided learning programs enhance learning in general, is completely different and cannot be answered with similar simplicity, but research results contain evidence of such phenomena (see e.g. Kiswarday (1996) and Moreno et al. (2002)).

Another important point concerning computer-augmented communication is that it is possible to activate more than one sensory channel simultaneously. Multimodal, interactive software makes it possible for a disabled learner to experience things in reciprocity, which he or she may have been totally unaware of so far. Computer-augmented communication expands a disabled student's sphere of life and takes him or her into a virtual society where it is possible to experience things without being farther burdened with the handicap.

1.2 Research issues

Learner at the center of the design. How does a computer help a learner with special needs? The computer can compensate for missing observation, expression and motor coordination. Clearly identifiable defects in motor co-ordination and sense perception, or various social and emotional benefits and disadvantages can be clarified while designing the educational software. Computerized solutions can open up the interactive process and thus support the start of a learning process.

Because of its varying requirements, special education provides computer-supported instruction with a particular challenge. Up till now, these challenges have been met with computerized one-purpose teaching tools. Instead of a technically-driven design which directs the passive student straight to the desired goal, new trends in education emphasize the activity of the learner. The computer should give him tools to explore, experiment, and evaluate — to construct his knowledge of the world. The shortest path to the learning objective might not be the most desirable one (Hübschner & Puntambekar 2002). Solutions for special education are not the same as they are for higher education, whether it is a question of disabled or elementary learners.

To improve a learning process for disabled children, the thesis proposes a model for structuring the learning material in a way that provides a personal learning experience for every individual, suitable for special as well as elementary education. The adaptation method raised from the model and the representation of the learning material are important issues to be discussed. Since the adaptation technique is not similar to traditional intelligent tutoring techniques, the expressive power of the learning environment, the authoring of the learning materials and the evaluation of the learning results are also under investigation.

Role of the teacher. In computer-supported special education, the role of the teacher needs to be re-thought. Though certain parts of the syllabus could be almost totally dealt with through computers, the students still need teachers as advisors and tutors. Therefore, a learning event in special education differs from a typical computer-aided learning event: often, there is a person present. Computer-supported special education does not need to replace the teacher or other helping hands, but free those experts to use their time and effort to more meaningful tasks.

The fact that the learning process in computer-supported special education does not take place without a human teacher sets new requirements for educational software. In addition to enabling the learner to work on the topic, the software should also analyze the learning process for the teacher, in order to support the teaching process. Moreover, since the teacher usually has more than one student, the software should be able to co-operate with the human teacher. Software supporting the learners does not suffice: the teacher can also benefit from computer assistance.

Research contributions. The research contribution is a *learning environment* consisting of three parts, presented in Fig. 1.1. The approach is to share an open learning space among its makers (authors), browsers (learners), and explorers (evaluators), without lapsing into a restrictive learning tunnel of the old days' behavioristic drills. We postulate that the learning space will give its users freedom to progress on a meaningful learning path, instead of being bound to the virtual infinity of meaningless options.

When defined more restrictively, the contribution is the *learning space model* that serves as a basis for systems to be used in special education, both for disabled children and in elementary education. The model is studied in a real-life setting by two implemented systems The design for the learning space model has evolved from the particular needs in a special school.

One of the issues in the learning space model is the capability to cater to different learning contents. In addition, the model should be simple and usable for interested non-experts, such as special teachers, to operate. The authoring process should not require computer science expertise.

The underlying idea behind the learning space model. The idea behind the learning space model is that we can try to guide the learner through certain parts of the task, but should not execute the task for the learner. The model serves as an adapter to a vast hyperspace; an interface between the human learner and the learning material. The system carries out the strategies incorporated into the learning material but remains invisible to the user.

The key issue is *support*. The learner needs support that can be provided in a computerized learning environment. However, the purpose is not

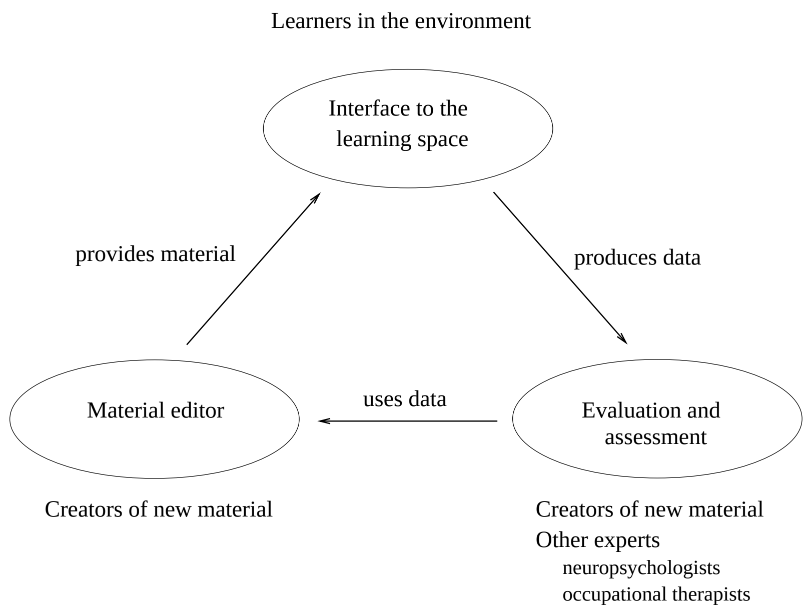

Figure 1.1: The learning environment for the thesis.

to act on behalf of the user and execute tasks but to help the learner in conceptualizing the learning task. Because of the openness for various materials and different instructional approaches or theories, the task conceptualization can be achieved, for example, by different types of metacognitive support in tasks.

1.3 Structure of the thesis

In the next chapter, we formulate the definitions and additional terminology that the thesis is built upon. Since the thesis is multi-disciplinary and tries – among other things – to bridge the gap between different disciplines, the concepts also cover other areas than the area of computer science.

The third chapter presents an interlude to related research in the area by reviewing and classifying educational software. The purpose of the classification is to motivate the need for an open model described in the thesis.

The main contribution of the research begins in the fourth chapter. The chapter defines the expressive model for structuring any learning material. The model allows hosting different domains and different users, and can be effective in different situations by providing adaptation (i.e. individual learning experiences). In addition, the model does not restrict the use of various learning theories or approaches. The part of the thesis describing the model is mainly based on a seminal paper about the model (Kurhila & Sutinen 2000*b*), updated and refined for the purpose[1]. Later in the chapter, we try to illuminate the operation of the learning space model by reflecting it against the operational principles in traditional tutoring systems, mainly to those well-established in cognitive science. The section derives from the issues discussed in Kurhila & Laine (2000).

The fifth chapter deepens the discussion about the features of the learning space model and describes the different content the domain independence enables. First, we review how the traditional computer-aided instruction can be incorporated into the learning space model, before proceeding into more advanced types, such as educational hypermedia, rudimentary adaptive educational hypermedia, computer games and simulations. The last two sections discuss learning material supporting specific deficiencies and evaluational material for neuropsychological assessment. Apart from the two last sections in the fifth chapter, the text is mainly based on Kurhila & Sutinen (1999). The last section discussing the support for mental programming is an enhanced version of Kurhila & Sutinen (2000*a*).

The sixth chapter describes the empirical evaluations of the learning space model. Two studies were conducted to test the model from the learners' point-of-view, one in the context of special education and one in the context of elementary education. The learning space model presented in the thesis is more powerful than the empirical evaluations suggest. The evaluations carried out were deliberately simple since large-scale evaluations were out of reach due to the heavy workload included in learning material design and implementation as well as organizing the field-trials. The two studies were originally presented in Kurhila & Varjola (2002) and Kurhila et al. (2002), respectively. The third study presented in the chapter concentrates on examining the model from the authors' point-of-view, illustrating the importance of ease of authoring in adaptive systems for learning. The third study is from Kurhila (2003).

Formalizing a scientific endeavour can open up possibilities otherwise missed. The seventh chapter makes a tentative step towards formalizing the learning space model and discusses some properties of the model. The chapter suggests some lines of work that could be followed. The text in the chapter builds on the work presented in Kurhila et al. (2001).

The last chapter summarizes the main points and discusses some of the questions raised throughout the thesis. In addition, some issues for future work are pointed out.

[1]Origins of the work are in Kurhila & Sutinen (1998) and Kurhila et al. (1998).

Chapter 2

Terminology related to the thesis

2.1 Hypertext, hypermedia and hyperspace

Today, hypertext and hypermedia are common concepts. The idea of hypertext dates back to the time before computers, and it is credited to Vannevar Bush (Bush 1945). A typical definition of *hypertext* is that hypertext consists of *nodes* and *links* between the nodes. Nodes are normally concepts, and links present relationships between the concepts. The concepts in nodes are presented in a textual form. If the nodes contain graphics, video, audio or any other non-textual format, it is normal to refer to the collection of nodes and links as *hypermedia* (Smith & Weiss 1988).

The links in hypertext or hypermedia can be bidirectional or restricted to one direction. The links can also be *typed*, for example, as specification links, elaboration links, membership links or others. In addition, the links can be referential for cross-referencing or hierarchical.

Hyperspace, on the other hand, refers to the nodes and their interconnections (links) as a structure. The use of the term hyperspace is often interchangeable with the term hypermedia. There is a clear distinction in the emphasis, though: hypermedia refers to the content of the hypermedia environment, and hyperspace refers to the nodes and links regardless of the node content.

Hyperspace can also be a space defined by more than three dimensions. Mathematically, a space may be defined by any number of dimensions, and the position of objects within that space may be located, much as we might locate an object in 3-space on the basis of axes of length, width, and height.

2.2 Learning with computers

Learning in general. In the thesis, the concept of *learning* should be understood as it is in standard dictionaries (Allen 1994). Learning is gaining knowledge or understanding of something by study, instruction, or experience. It should be noted that learning is not to be mistaken for memorizing, although in some cases learning includes memorization.

A *learner* is a person subjected to learning. A person can be a learner even if actual learning does not take place. In this thesis, a learner is also a person taking part in the learning environment. However, the term learner always refers to a person supposed to be learning when using the system or participating in the learning environment, whereas a *user* refers to the person modifying or providing content to the learning environment. The user is as important as the learner, since creating the learning material is an essential part of computer-assisted education.

A *learning event* stands for a session, during which learning takes place. The learning event is a linear event, with a starting point and an ending point, thus referring to a certain time interval. The learning event can also be called a learning *session* or, in some cases, a learning *experience*.

The *learning environment* is constructed – in this thesis – from the physical place where there are learners, teachers, equipment and anything related to the learning event. In some cases, where the context is obvious, the learning environment refers to the collection of computers with corresponding software and input as well as output devices. In the literature, the term learning environment can also refer to only one piece of educational software. To name an example, Anderson (1995) describes reflections about their ten years of research on intelligent tutoring systems, which they call *tutors*. They report a recent conceptual change in terminology: "We now conceive of a tutor as a learning environment in which helpful information can be provided and useful problems can be selected." This does not contradict the usage of the term in this thesis, because a piece of software can be used in several computers simultaneously, thus forming a learning environment as we see it. However, a piece of educational software is referred commonly as a *learning system* in current research literature, used also in this thesis.

Computer-aided learning. The question of *computer-aided learning* against *computer-aided education* and *computer-aided instruction* is widely discussed in the literature (Alessi & Trollip 1985, Steinberg 1991, Boyle 1997). In addition, all of these concepts are normally abbreviated with the traditional CAI, even though computer-aided instruction refers to a different emphasis in the learning process. Instruction refers to instructional methods and thus implies behavioristic instruction or education. There is a difference between education, instruction and learning, but in this thesis, they are interchangeable. The same applies to the term *computer-supported learning*. In the scope of this thesis, there is no difference between "supported" and "aided".

2.3 Adaptation in computer-supported education

Learning systems can have built-in adaptation mechanisms. A system, which can be adapted by the user before or during the action of the system, is called *adaptable*. In practice, adaptable environments are adapted by parameters, often called *user preferences*. The parameters are used in determining various variables, such as font size and color, sound volume etc.

When referring to the autonomous adaptation of a system, it is common to use the term *adaptive*. These systems adapt autonomously according to the user's operations in the environment. In the scope of this thesis, an adaptive learning system refers to a program, which uses only deterministic and purposeful adaptation. This rules out e.g. computer games, such as simulators, where the state of the simulated world changes according to random functions.

A learning system can be both adaptable and adaptive at the same time. However, often in such systems, the individual properties are either adaptive or adaptable; it is unlikely for a single property to be both adaptive and adaptable.

Adaptive and adaptable learning systems have slightly different uses. Adaptation provides a changing environment according to the actions taken during a learning session. In an adaptable system, after the parameter adjustment, the session is fixed. However, parameters can be altered during the learning session, so that the learning event resembles the event achieved by an adaptive system. The essential difference is that with an adaptable system, the user is in control and is supposed to have enough metacognitive skills to decide how to adjust the system, whereas in adaptive systems, the user model a system builds takes the responsibility of being the basis for the alteration of the environment.

In many cases, systems are both adaptive and adaptable. In fact, it has been proposed that a system needs to have both aspects of adaptation, simply because it is remarkably difficult for a machine to interpret the user, especially before enough input has been received from the user (Höök 2000).

The standard way has been to provide some parameters a user can alter, such as fonts and colors, and the adaptation during the learning sessions is autonomous (see for example ELM-ART (Brusilovsky et al. 1996*a*)).

2.4 Disabilities related to the thesis

Overview of terminology. A standard way to classify disablements is to divide them into disabilities, impairments and handicaps. According to the United Nations declaration (Anonymous 1975), "the term 'disabled person' means any person unable to ensure by himself or herself, wholly or partly, the necessities of a normal individual and/or social life, as a result of a deficiency, either congenital or not, in his or her physical or mental capabilities".

As Edwards (1995) points out, this definition applies to all people. His refinement to the definition above is that "some people have impairments of their faculties which severely affect their ability to take part in everyday life, and those people are usually referred to as being *disabled*". This is the view also applied in this thesis.

An *impairment* is a deficiency or abnormality in the physical or mental condition which manifests itself in structure or in action. An impairment is not related to birth or to development. It can be innate or acquired. Impairment can be related to e.g. hearing, learning, seeing, physical or motor action, or cognition.

In the thesis, the term *special education* refers to the education of children with disablements. The opposite of special education is *regular education*. If there is a need to emphasize that the learners are not disabled, a term *non-disabled education* is used.

Motor impairments. The reasons and manifestations of motor impairments vary from mild impairments to severe. In a case where the motor impairment is mild, the user can e.g. use larger buttons for input. However, in the scope of this thesis, the interest also lies elsewhere. Computers should offer a meaningful environment when a user can only elicit minor movements with, for example, the head. In such cases there is a need for *single-switch input* and *scanning of choices*.

Single-switch input refers to an input device, which can be used – together with scanning (see below) – as an input method for a person with restricted mobility, in a case where a person cannot use more than one switch. In this case, one switch does not refer to one switch at a time, but truly one switch. In the following text, the term one-switch input is sometimes used instead of single-switch input.

Scanning is a method used in input for persons, who can only use one switch. In scanning, selectable options are highlighted in turn, and the user can make a choice when the desired option is highlighted. In a typical situation, the scanning time of the options can be reduced if the options are divided into rows and columns, and the desired row is chosen first, and the desired column next.

Scanning is also used if a person can use an input device with two switches: one switch to scan the choices in a fixed order, and another switch to select the desired choice. Since this method does not offer any radical improvement to the usability, persons who could use two switches (e.g. a person who can nod his head to the left and right) still use only one switch with scanning, because of the physical strain every choice causes.

Deficiencies in mental programming. The main user group for the thesis is children with deficiencies in *mental programming*. The definition of mental programming is not agreed on globally. The following definition, adapted from Vilkki (1995), is used in this thesis. Mental programming is "the subjective optimization of subgoals for the achievement of the overall goal with available skills". To put it slightly differently, "[mental] programming can be seen as a process that activates, adapts, and modifies previously established plans in unexpected situations during the course of action." However, as Vilkki (1995) points out, mental programming is the optimization of *conscious* subgoals, so mental programming is always a conscious activity.

A decisive property in mental programming is the "interactive search for subgoals and operations (behavioral routines) which are subjectively optimal for the achievement of the overall goal" (Vilkki 1995). A *goal* is a conscious subjective representation of a state or outcome to be achieved (Luria 1973). *Operations* are habitual means to accomplish actions under variable but specific conditions (Vilkki 1995). An *action* is "usually a series of operations planned or programmed for a specific purpose and situation" (Vilkki 1995).

According to Vilkki (1995), the division of the task into optimal subgoals *succeeds*, if two complementary aspects succeed. First, the selected set of subgoals should lead to the final goal as efficiently as possible. Secondly, the subject should be able to reach the selected set of subgoals with his or her operational resources (i.e. operations).

Mental programming *fails*, if one of three conditions occur (Vilkki 1995). First, if the subject does not find a set of subgoals that leads to the completion of the final task. Secondly, if the selected set of subgoals cannot be reached by the operational resources of the subject. Thirdly, if the selected set of subgoals is not optimal, i.e., a more efficient set of subgoals exists.

Deficiencies in mental programming are caused by frontal-lobe lesions (Luria 1973, Korkman 1988). Typical to these lesions, other than mental programming disorders, are also emotional indifference, lack of initiative, and poor social judgment.

Although it is not a part of mental programming, there is also evidence that the *feeling of knowing* is impaired after frontal lobe lesions. By the feeling of knowing, Vilkki (1995) refers to an ability to accurately evaluate the success or a failure of a action.

As well as mental programming, motivation is also an important factor when considering patients with frontal lobe lesion. Normally, if the completion of a task seems to be possible but requires more than a simple routine operation, a subject is more likely to be motivated. And, if the achievement of a goal seems impossible, a subject feels emotional rather than motivated. Therefore, the motivation to achieve a goal or accomplish a task depends on relatively stable motives and values and on the subjective probability to achieve the goal with the means and skills available (Atkinson 1964). The subjective probability is best if it is near 0.50. With frontal lobe lesion, this matching of subgoals with available operational resources is disturbed. This can be explained if mental programming is seen as an intermediate process between performance and motivation. As Vilkki (1995) puts it, "the subjective optimization of subgoals integrates motivation and skills (operational resources) to purposeful activity."

For frontal lobe lesion patients, the triggering mechanism of the ability to generate autonomic responses is also altered. Damasio et al. (1991) describes this with an example. In a test situation, the testees was shown neutral pictures and pictures with a strong implied meaning (social disaster, mutilation, or nudity). The testees did not react differently to different pictures. However, the testees did react to pictures with strong implied meaning, if the testees had to comment the pictures verbally. This was in support of the hypothesis that the triggering mechanism to generate autonomic responses was not destroyed but altered (Damasio et al. 1991).

Deficits in mental programming occur very often with developmental disabilities, so the number of potential users is much more than one would think. In addition, the frontal lobe lesions causing deficits in mental programming can be innate or acquired later in life, thus increasing the amount even more.

Chapter 3

Review of educational software

3.1 Motivation

There is no purpose in only reviewing learning systems for special education, since the vast majority of existing solutions are nearly trivial from a computer science point-of-view. Therefore, we concentrate on investigating educational software in general. The aim of this chapter is to outline the properties of a desirable learning system so that the system would be usable in special education.

It should be noted that educational software has been classified in the past (see e.g. Heller (1991) and Squires & McDougall (1994)), but classifications of software for special education, with an emphasis on computer science, do not exist. Since the emphasis is on computer science, we investigate what kind of solutions computer science can bring to educational software, and not judge software if it is made to support e.g. instructivist rather than constructivist learning theories.

Overview and examples. Much educational software, targeted to some specific disablement, exists. The most often addressed special needs are visual and hearing impairments. Also, assistive technology and software for blind persons are common (although the approach taken and the style of operation of these systems varies remarkably). However, mental disabilities, such as learning difficulties or aphasia, are rarely addressed.

Unless we consider slight impairments concerning e.g. hearing, disabled users pose demands on educational software that rule out most of the standard educational software. The software produced for regular education simply cannot be used in versatile environments found in special education.

In the field of assistive technology, most of the computer science oriented research for special needs concerns hardware. Hardware solutions consist of specially designed input or output devices (see e.g. Ross & Blasch (2000)). If the scope of the research is software, the solutions are often enhancements in the interface design (see e.g. Smith et al. (2000)).

Examples of learning systems designed for special education are usually simple from a computer science viewpoint. One of the reasons might be that these systems are not designed in collaboration with the computer scientists. Even software engineering is possibly done by an amateur, such as a teacher interested in programming. The two disciplines, computer science and special education, have rarely met.

Some ideas in learning systems for special education express nothing short of brilliant innovations, but the innovative ideas have not been in the area of computer science. An example of typical (but not brilliant) educational software for a disabled audience in general is a computerized version of a traditional memory game, where a learner has to find matching pairs. The program itself does not offer any new aspects to the age-old game. Some could even say that transferring such a simple game to a computer brings an extra cognitive load for the learner. However, we should keep in mind that the user group may not be able to play the memory game with any other means than a computer.

Dimensions of the classification. Adaptation has proven to be helpful in learning systems when addressed to regular education. For a review on the topic, see Brusilovsky & Eklund (1998*b*); more recent findings include Conati & VanLehn (2000), Hammerton (2002) and VanLehn et al. (2002), although zero effects have also been reported (see e.g. Ainsworth & Grinshaw (2002)). The case with adaptation is likely be the same with special education. In fact, adaptation to individuals is much more crucial in special education, since every learner is unique, and the variation between the learners can be huge, not only in the area of factual knowledge but in other dimensions (motorical, seeing and hearing) as well.

Openness in learning content is another key issue in special education software. Since special education classes are small, the markets are significantly smaller than for normal educational software. That is why there is a need for flexibility in the learning content, so that the special teacher can incorporate new material from different domains, according to individual curricula and different needs.

Support for special needs is essential, if a learning system is to serve a wide special education population. For example, motorical impairments are not rare, and the most universal way to tackle the limitations in input is to use single-switch input with scanning. Most of the advanced learning systems do no have support for single-switch input implemented. Moreover, there are several other types of special needs as well. We can speculate with the possibility of altering the systems so that they support special education. For example, single-switch input does not require much computer science contribution to be implemented (in fact, it is a question of rather trivial software engineering), but the pedagogical solutions and way of interaction should also be designed to support single-switch input.

Therefore, gathering the evidence from both research literature and actual field workers, we can conclude that, to be successful in special education, educational software needs these three properties:

- adaptation to individual learning processes
- openness in learning content, and
- support for special needs

The result. The examination of these three properties form the core of the classification. It should be noted that the intersection of the systems having the first two properties (adaptation to individual learning processes and openness in learning content) and the systems designed particularly for disabled users, is empty. Therefore, it is evident that there is a great deal to do in the field of computer science for the benefit of special education.

3.2 Educational software paradigms

Since it is not possible to classify all of the educational software for the purpose of this review, we will settle on the representative examples within each paradigm of computer-aided learning. The paradigms presented are not well-established, and there is a certain amount of overlap. Because of our purpose, we have omitted some steps in the continuum of developing learning systems that could be regarded as paradigms (e.g. Interactive Learning Environments, ILEs) since they are of no interest in this thesis. Another point to make is that the systems presented are biased in favour of academic research, since business-driven research and development has not been extensively reported. However, many of the academic systems have been commercialized recently. The paradigms included in this classification, in chronological order, are:

Traditional computer-aided instruction, CAI: Traditional CAI systems are non-adaptive with a fixed content. The first examples of this kind date back to the 1960's, but still today the most commercial learning systems employ this paradigm. It should be noted that although the system is not adaptive, the learning sessions can still be somewhat different for various users, since the learner can have different choices to make within the system and receive feedback accordingly. This instructional philosophy is often referred to as *learner-controlled instruction*. Also, most of the special education software falls into this category.

Adaptable learning systems: Many systems have the property of being adapted for individual users. Since the need to adapt the learning system for different types of users is evident in special education, the adaptable properties are often found in high-quality commercial special education software. This slight change in educational software paradigms is nothing but rather trivial software engineering, therefore not interesting in this thesis. It is, however, important for the users especially in the context of special education.

Intelligent Tutoring Systems, ITS: Since the beginning of the 1970's, the evolution of incorporating artificial intelligence into educational software saw daylight. One of the first systems of this approach was Scholar (Carbonell 1970, cited in Wenger 1987). Scholar made a well-controlled paradigm change from frame-oriented CAI to adaptive systems (called *information--structure-oriented CAI* by Carbonell). The Scholar system was operating in the field of South American geography. The system picked dialogue topics rather randomly, but the responses from the system were different according to the learner's input. Although ITS have been developed extensively after Carbonell's seminal work, the direction of the research was to bias the systems towards more refined learner modelling and teaching strategies. The systems were heavily domain dependent, although the more recent systems could have domain-independent parts in their architecture (see FITS (Nwana 1993*b*, Nwana 1993*a*) for an example of such system). Other examples of traditional intelligent tutoring systems include ACT-tutors such as Lisp Tutor (Anderson & Reiser 1985), and its descendants Geometry Tutor (Anderson et al. 1986) and

Algebra Tutor (Koedinger et al. 1997). Before the strong prevalence of graphical user interfaces with direct manipulation of objects, the systems from the "old-school" were mainly text-based, often supporting ways of dialogue. Therefore, natural language processing was an important research topic related to ITS research.

Adaptive Educational Hypermedia, AEH: After the dawn of hypertext, the area saw the rise of adaptive educational hypermedia systems, although most of the systems still today use only forms of hypertext. The explosive popularity of World-Wide Web, the area of Web-based AEH has dominated the adaptive learning system research. Most systems adapt the presentation of hypertext and/or support navigating by adaptively annotating (or hiding) links. The adaptation is based on user modelling, often adapted from the ITS systems. Well-documented examples are AHM (da Silva et al. 1998), HyperTutor (Perez et al. 1995) and ISIS-Tutor (Brusilovsky & Pesin 1995). Some systems are *hybrids*, incorporating properties found in both ITS and AEH. Examples include ELM-ART (Brusilovsky et al. 1996*a*, Weber & Brusilovsky 2001), where the user has the same kind of problem-based learning possibilities as in ELM-ART's predecessor ELM-PE (Weber & Möllenberg 1994). Naturally, most of the systems stretch the AEH paradigm to *distance education* using the Web. Examples include AHA[1] (de Bra & Calvi 1998), DCG (Vassileva 1997), AST (Specht et al. 1997) and AIMS (Aroyo & Dicheva 2001).

ITS shells and ITS authoring tools: This paradigm shift started in fact before the shift from ITS to AEH, and it concerns both ITS and AEH. To reduce the costs and improve effectiveness, a concept of *ITS shell* was formulated. ITS shells are generalized frameworks for building ITS, whereas ITS authoring tools are ITS shells with a user-interface for non-programmers to formalize and visualize the knowledge (Murray 1999). The goal of the ITS authoring system is not modest, and it has proven remarkably difficult to provide domain-independent authoring tools, which support pedagogically strong and meaningful adaptations, and still do not lack usability and ease-of-use. Murray (1996*b*) points out that there are decision tradeoffs in

[1] Technically, AHA is not adaptive educational hypermedia but an adaptive hypermedia system designed to support other forms of hypermedia use as well.

ITS authoring tools: complete domain-independence in an authoring tool means a more shallow tutor, and so does too much ease-of-use. The systems include e.g. Eon (Murray 1996*a*), Coca/REDEEM (Major & Reichgelt 1991, Major et al. 1997), Elint (Vassileva 1990), Calat (Nakabayashi et al. 1998) and InterBook (Brusilovsky 1998). Calat and InterBook are authoring systems for Web-based adaptive educational hypermedia, thus crossing paradigm boundaries. Of course, there are also hypermedia-based learning systems without adaptation and systems to build non-adaptive hypermedia learning systems but these are relatively uninteresting in the thesis since the contributions in them are often outside the area of computer science.

Agent-based learning environments, ABLE: Agent-based learning environments can be viewed as the most recent paradigm in computer-assisted learning research. Although the research around agents is just taking its form, there have been several serious attempts to employ agents as essential players in a learning system. One of the first steps to this new paradigm was the Learning Companion System (LCS) architecture (Chan & Baskin 1990) and its instantiation Integration-Kid (Chan 1991), although strictly speaking, it could be considered a traditional intelligent tutoring system. As in Integration-Kid, agent-based systems often deploy simulated *learning companions* as agents. This is the case for example in EduAgents (Hietala & Niemirepo 1996, Hietala & Niemirepo 1998). Other ways to include agents have been using them as helpers, which take a visual form (see for example Adele (Rickel & Johnson 1997, Shaw et al. 1999) for a project where agents are helpers-on-demand in a virtual reality environment for case-based medical education and training). Agents are also used in supporting collaborative learning by facilitating communication and collaboration (Ayala & Yano 1996, Greer et al. 2001), or modelling learners (Paiva 1996). In many cases, agents *per se* do not add anything to the environment, but considering learning environment participants as agents has caused a shift from teacher-oriented tutoring to more supportive learner-centered education. Contradictory to the last statement, some researchers have employed agents only as an architectural solution to reduce the cost of building an adaptive system (Cheikes 1995).

3.3 Desired properties in learning systems

After presenting the paradigms of educational software, we are ready to discuss the desired properties in a learning system. Following the terminology previously used in this chapter, we use the term *adaptive learning system* when referring to any or all of the following: intelligent tutoring systems, adaptive educational hypermedia, ITS shells, ITS authoring tools and agent-based learning environments.

3.3.1 Adaptation to individual learning processes

The first aspect in the classification is the *adaptation to individual learning processes*. The division is made by judging whether the system is non-adaptive, adaptable, adaptive, or both. In this particular case, we are not interested in the technique used to provide the adaptation, so we do not examine whether the adaptivity is achieved by software agents or by ordinary intelligent tutoring system techniques. By agents, we mean both agents that appear visually on screen, and the architectures that can be constructed to support agents and/or agent-based programming.

The term adaptation is by definition more closely related to adaptive educational hypermedia than to intelligent tutoring systems. Conceptually, intelligent tutoring systems are adaptive but in many cases they try to adapt the *learner* to the system, not the system to the learner.

Whereas in ITS the model of interaction has often been a text-based (socratic) dialogue, in AEH the emphasis is often on allowing more explorative learning. Therefore, in AEH, there is a need for additional adaptation properties. Brusilovsky (1996) has classified the properties that can be adapted in adaptive hypermedia (see also Brusilovsky (2001)). The first is adaptive presentation of the contents, and the second is adaptive navigation support. Adaptive presentation of contents usually means showing additional explanations or hiding unwanted parts of a presentation from the user. These parts are unwanted because the user is not assumed to have prerequisites of a concept. Adaptive navigation support stands for adaptive sorting, annotation or hiding of links, but sometimes also direct guidance or navigation map can be adapted.

Objectives. In education, every learner is unique and has personal preferences and methods for learning, as well as different ways of constructing knowledge and process information. To support individual learning processes, a computer-aided learning system should provide individual support for every learner. In special education, whether it is for disabled children or elementary education, the demand for this personalization is even more obvious. The individual differences between learners in a special school are far greater than in regular schools. Special needs range from motorical, visual, aural or cognitive demands to every combination of these. They all have a specific effect on the learning event.

In special education, a natural way to divide the responsibilities between adaptive and adaptable properties, is to use adaptable qualities in modifying the input and output. Adjustment of colors and font sizes is needed for learners with low vision. Motorical impairments can produce a need for extra-ordinary input devices, or, in less severe cases, rule out only certain devices, such as standard mouses. Mental disabilities, such as learning difficulties, can then be addressed in an adaptive manner, autonomously.

Challenges. The challenges in preparing an adaptive system are multiple. Adaptivity *per se* is a difficult issue. A system should adapt *correctly* to the user's actions or lack of actions. Even though a learning session may not qualify as so mission-critical that every decision the system makes has to be correct, the learning session should not frustrate the learner by drawing misconclusions about the learner.

Therefore, building and maintaining an accurate learner model is one of the challenges. Learner modelling is a complex issue, and work on learner modelling has many forms and therefore differences in opinion. However, most learner models are built with an *overlay model* (see e.g. Wenger (1987)). An overlay is a method in which the learner's knowledge about the subject is presented as an overlay of the domain knowledge. The domain knowledge is usually represented as a collection of concepts linked together. The overlay contains a – usually binary – value about the estimation of the learner's knowledge level of the concept (Brusilovsky 1996).

Other forms of learner modelling exist. One popular method is to use a simple *stereotype model* (Brusilovsky 1996). In a stereotype model, possible learner profiles are distinguished to several "stereotype" users, for example a beginner, an intermediate, and an expert. Stereotype models are less expressive, but they are easy to maintain and compute.

Also, the dependence on the *model of teaching* is a challenge in adaptive learning systems. Often, the model of teaching is fixed to some instructional theory and cannot be altered. This is not necessarily the case in ITS authoring tools or some adaptive educational hypermedia systems, but total freedom of incorporating different pedagogical views is a goal yet to be achieved.

One of the challenges is to maintain the usability of the system, when advanced adaptation mechanisms are incorporated. By this, we mean both the usability for a *user* in updating the system or the system's learning content, and the usability for a *learner* in a learning session. To name an example, if the learner model is not accurate enough, or the modelling demands answering too many questions before the system starts, we can say the usability of the system is low.

One major question is that in an optimal situation a system should be both adaptive and domain-independent. The challenge in this situation is to maintain the adaptation if the content is domain-independent. Not too many systems are capable of this. In every case, some form of metaknowledge about the learning content has to be provided by the content author to maintain the adaptation.

Many adaptive learning systems have a fixed content; the content is neither modifiable nor extensible. The hours needed to build an intelligent tutor or any kind of adaptive system is huge, and the construction has usually started from scratch. Different remedies have been proposed. One of the best ways is to use authoring tools to provide intelligent tutoring (see Murray (1999) for a complete survey), or shift to more modular ITS shell components (for example, see Vassileva (1990) and Vassileva (1992) for an architectural solution for a domain-independent ITS shell).

Another possible problem with adaptive systems is the *computational complexity*. In a standard case, the modelling of a learner is very much imperfect, so that the computational load is not going to be overwhelming. The overlay and stereotype models used do not pose large demands on the system, but the fact that computational complexity is an issue, has restricted the research and biased it towards making the models less complex and thus more imperfect. However, the usability in learning systems can be seen as a far more important issue than perfecting the learner model; it is necessary that the response times are short.

The effects of adaptation in a learning system are not supported by enough *empirical evidence*. This point is made by Brusilovsky & Eklund (1998*b*), to respond to the critique towards ITS research (examples can be found from Rosenberg (1987) among others). It turned out that the empirical evaluation is often either not valid, since the test groups are too small, or irrelevant, since the evaluation revolves around uninteresting issues such as counting the navigation steps (Brusilovsky & Eklund 1998*b*). Although several empirical tests have been carried out with various adaptive systems, there is still a need to validate the research results more thoroughly.

Examples. As a representative example of adaptive systems, we consider ELM-PE (Weber & Möllenberg 1994) and its descendant ELM-ART (Brusilovsky et al. 1996*a*, Weber & Brusilovsky 2001). ELM-PE is an intelligent learning system supporting example-based programming and analysis of a learner's solutions for problems. It is based on modelling the learner in terms of a collection of episodes, hence the title Episodic Learner Model (Weber 1996). In short, these episodes can be viewed as cases, as in case-based reasoning (Weber et al. 1993).

ELM-PE is designed to support novices in learning the programming language Lisp by problem solving. It has features to give immediate feedback, to reduce the working memory load, to support learner activity, to support example-based learning and to avoid unnecessary mistakes. ELM-PE is a complete programming domain, where a learner can learn Lisp by programming. The point is to offer help on-demand, and only in critical situations a system takes an active role. The basis of the adaptive helping in ELM-PE is a knowledge base consisting of the knowledge about problem solving in Lisp. This is represented as a network of concepts, plans and rules, and the learner modelling with an overlay model. This leads to different kinds of support: finding errors in the code, completing the coding exercises, or assessing if the learner's solution is correct.

ELM-ART is a Web-based intelligent tutoring system in the field of Lisp programming. ELM-ART is largely based on ELM-PE (Weber & Möllenberg 1994). The main distinction is that ELM-ART is to be used in distance learning. It provides both course materials and problem solving support on-line.

ELM-ART provides presentations of new concepts, test, examples, and problems in hypermedia form. To function adaptively, ELM-ART has a certain knowledge about the material it contains, so that it can support learners in navigating the course material (Brusilovsky et al. 1996*a*).

Whereas ELM-PE is a system with an open programming environment with help on-demand, ELM-ART is an "intelligent textbook" where the course material is Web-based hypertext, thus entailing a need for adaptation in some additional properties (Brusilovsky et al. 1996*a*). As the learning material in ELM-ART is provided as freely-browsable hypertext, the system uses two adaptive hypermedia techniques to support the student navigating through the course: adaptive annotation of links and adaptive sorting of links (Brusilovsky et al. 1996*a*). ELM-ART has three instances of adaptation: adaptive navigation support, prerequisite-based help, and intelligent problem solving support. Adaptive navigation support is based on the overlay model of the learner.

An example of prerequisite-based help is a student entering a page which is not yet ready to be learned (Brusilovsky et al. 1996*a*). Then, the system warns the learner that this material has unlearned prerequisites and shows additional links to textbook and manual pages where the unlearned prerequisite concepts are presented. When the student has problems with understanding some explanation or example, or solving a problem, he or she can request help using a help button and, as an answer to the help request, the system will show the links to all the pages where the prerequisite knowledge is presented.

Both ELM-PE and ELM-ART are systems that support example-based programming. They encourage the students to re-use the code of previously analyzed examples when solving a new problem. In ELM-ART, the learner can send a Lisp expression for evaluation or send a problem solution for analysis. An important feature of ELM-ART is that the system can predict the learner's method of solving a particular problem and find the most relevant example from the learner's profile (Brusilovsky et al. 1996*a*).

An interesting issue concerning ELM-PE and ELM-ART is that, because ELM-ART is transferred to the Web, there was a need to discard some properties from ELM-PE to enable the transfer (Brusilovsky et al. 1996*a*). This represents the paradigm shift from ITS to AEH: the learner modelling and user-adapted interaction became more shallow, but the systems became more domain-independent and were transferred onto the Web, thus enabling distance learning and a much wider audience.

Another example of classic adaptive tutoring systems is the Lisp Tutor (Anderson & Reiser 1985), and its descendants the Geometry Tutor (Anderson et al. 1986) and the Algebra Tutor (Koedinger et al. 1997). The tutors were created to support the development of ACT theory (Anderson 1993) experimentally. The ACT tutors are traditional in the sense that they try to keep the learner in an optimal solution path, although the latest versions allow some degree of freedom in learning (Anderson 1995). The ACT tutors are remarkably well-known, and they are among the few ITSs that are actually evaluated outside the research laboratories. They represent classic ITS research also in the sense that they are domain-dependent. The content domain is mathematics although tools for authoring the content have been built (Ritter et al. 1998) thus making them less domain-dependent.

3.3.2 Openness in learning content

The second dimension of our classification is *openness in learning content*. Because computer-supported special education has suffered from the lack of usable systems in several content domains, it would be beneficial to have a learning system, in which the educational content is not tightly-coupled into a domain but generic so that it could serve various kinds of education. The need for domain-independent systems gives rise to another problem. To ensure that domain experts are able and willing to contribute to the learning materials, means for authoring the material should be simple yet expressive at the same time.

Objectives. Especially intelligent tutoring systems have suffered from strong dependence on content domains. Even slight alterations to learning contents are often impossible. However, within the ITS research community, there has been a strong tendency to overcome this problem.

Domain-independence has already been acknowledged as one of the objectives in a computer-based learning system. Vassileva (1990) formulates it as "[an intelligent tutoring system] must be easily adaptable to work in various domains, without forcing the teacher to study programming". By domain-independence in learning content, we can overcome the restrictions of re-usability, thus saving the resources, time and effort to produce usable systems across the curriculum. This need is even more clear in the field of special education, where the resources are often more limited.

The solutions to tackle the problem for dependence on a content domain are, in fact, the same as when proposing easy-to-do adaptive systems. The solutions range from slight changes in ITS architecture (Vassileva 1990) to making more reusable modules and simple but versatile authoring tools (Murray 1999).

One form of partial domain-independence is to allow the content author to modify the learning content by switching some parts of the contents, or, more usually, adding new items to the contents. In any case, the *form* of the content is well-defined, and the new material should fit to this form. This is a question of rather trivial software engineering, and it has been done in several examples of educational software. In some cases, the alterations a user can make can enhance the usability.

Domain-independence gained popularity when the shift from ITS to adaptive educational hypermedia became reality. The educational trend towards learning by exploring or learning by doing in open learning systems was the thing the community needed. Then, domain-independence was a *must*, and it was not questioned. However, another problem appeared: how to maintain the individual adaptation, and still enable complete domain-independence?

Challenges. In adaptive systems which allow authoring novel material, the *ease of authoring* is a desirable property. Authoring an intelligent tutoring system is not as trivial as authoring Web pages. Depending on the model behind the learning material, one has to tell the system something about the material. In a standard case, the material has to be *indexed* or *scripted* in a certain way to enable adaptation during learning sessions. Particularly interesting descriptive examples of using indices as a basis for large, meaningful hypermedia systems exist (Schank 1990, Schank & Osgood 1992, Schank et al. 1993, Osgood 1994, Jona 1995, Bell 1996). The problem is to lure teachers or other personnel to create additional learning contents to the system, so the task of providing the metaknowledge needed has to be made very simple. However, utmost simplicity is not likely to succeed in ITS authoring, as Murray (1999) points out.

Another challenge in domain-independent systems is the *expressive power* of a learning system. Here, expressive power refers to the types of learning content the system can present. If the system is completely domain-independent, there can still be restrictions on what kind of material can be presented. As an example, many Web-based systems offer only the functionality of standard HTML. Of course, by using e.g. Java applets on Web-pages one can have enhanced interaction, but using Java contradicts the ease of authoring, since submitting adaptation information between the Java applet and the rest of the system becomes complicated.

It is a common conception that providing a sound pedagogical model of delivering the content (i.e. the teaching model), a system cannot be completely domain-independent but suitable only for a *class of domains* (Dooley et al. 1995, Murray 1996*b*). Even if the system itself is domain-independent, the system can be too complex to use because often systems are based primarily on theoretical concerns or artificial intelligence techniques (Murray 1996*a*).

The domain model is not the only thing that should be left open. The instructional model (i.e. teaching model) should also be independent of the rest of the system. One of the major reasons for the lack of success in ITS shells is that they are based on a specific instructional approach (Murray 1996*a*), and therefore, the tutoring systems built with these shells have also suffered from the fixed instructional model. One of the remedies brought by some ITS authoring tools is the independence from a fixed teaching model. Examples exist (van Marcke 1992, Cheikes 1995, Major 1995), but the easy-to-use systems have been scarce (Murray 1996*b*).

Examples. An example of a completely open system is InterBook (Brusilovsky et al. 1996*b*). The system can hold any domain, and the domain knowledge has to be assigned with metadata. As is often the case with open systems, InterBook is partly an authoring system and partly a system for the learner (i.e., an authoring and a delivery tool). InterBook is based on experiences with ELM-ART, so the adaptation is somewhat similar in these systems (Brusilovsky et al. 1996*b*). However, certain tradeoffs between the adaptation and the domain-independence have been made.

Adaptive learning systems created with InterBook are called "electronic textbooks" (Brusilovsky et al. 1996*b*). These adaptive textbooks use knowledge about its domain represented in the form of a domain model and about its users represented in the form of individual user models. The domain model serves as a basis for structuring the content of an adaptive electronic textbook (Brusilovsky et al. 1996*b*).

Another central part in an electronic textbook created with InterBook is the glossary. According to the approach taken in InterBook, the glossary is considered as an externalized domain network. Brusilovsky et al. (1996*b*) describe the domain network as follows:

> "Each node of the domain network is represented by a node of the hyperspace, while the links between domain network nodes constitute main paths between hyperspace nodes. The structure of the glossary resembles the pedagogical structure of the domain knowledge and, vice versa, each glossary entry corresponds to one of the domain concepts. The links between domain model concepts constitute navigation paths between glossary entries. Thus, the structure of the manual resembles the pedagogic structure of the domain knowledge. In addition to providing a description of a concept, each glossary entry provides links to all book sections which introduce the concept."

To make the textbook more adaptive and to connect it to the glossary, the system has to know what each unit of the textbook is about. This is done by indexing textbook units with domain model concepts. For each unit, a list of concepts related with this unit is provided.

Indexing is a relatively simple but powerful mechanism, because it provides the system with knowledge about the contents of its pages: the system knows which concepts are presented on each page and which concepts have to be learned before starting to learn each page (Brusilovsky et al. 1996*b*). It opens the way for several adaptation techniques.

InterBook supports sequential and hierarchical links between sections (Brusilovsky et al. 1996*b*). It generates the table of contents where all entries are actual links. In addition, it generates links between the glossary and the textbook. Links are provided from each textbook unit to the corresponding glossary pages for each involved concept. On the other hand, from each glossary page describing a concept the system provides links to all textbook units which can be used to learn this concept.

To support the learner navigating through the course, the system uses adaptive annotation of links (Brusilovsky et al. 1996*b*). Using the learner model, the system can distinguish several educational states for each page of material: the contents of the page can be known to the user, ready to be learned, or not ready to be learned (the latter example means that some prerequisite knowledge has not yet been learned). This is similar to ELM-ART, as well as the method for prerequisite-based help.

The main point in InterBook is that it is one of the truly domain-independent adaptive learning systems. The adaptation techniques are not the most powerful ones, but the trade-off between domain independence and adaptation is reasonable. The work with InterBook has continued (see (Brusilovsky et al. 1998) for the architectural description, and (Brusilovsky & Eklund 1998*a*) for experimental evaluation). Another example of more or less domain-independent systems is Co-Operative Classroom Assistant, COCA (Major & Reichgelt 1991), to be used in authoring a tutoring system. The user can select the domain content, teaching strategy and meta-teaching strategies. It uses a teaching process model where the next topic, the content detail and the teaching action are determined by the meta strategy. This meta-strategy is a set of rules based on the student history, enabling flexible adaptation. COCA is a text-based system, and its descendant REDEEM (Major et al. 1997, Ainsworth et al. 1999) is more versatile in that aspect, employing graphical user interface.

At the Institute for Learning Sciences, the learning systems are based on story-telling (ASK) model (Schank & Osgood 1992). An example system is Engines for Education (Schank & Cleary 1994). The ASK model is based on an assumption that the best way to learn is to listen to stories of experts and ask questions. The model is actually a hypermedia design methodology, so the systems based on the model are domain-independent: every domain can be decomposed and indexed according to the design methodology. Although the systems are based on asking questions, the questions are pre-defined and presented as multiple choice.

Other examples of domain-independent systems are CALAT and Eon. CALAT offers adaptive tutoring on the Web with adaptation techniques which are not particularly powerful, but it includes teaching strategy customization (Nakabayashi et al. 1998). Eon (Murray 1996*b*) is a meta-authoring tool for creating ITS. It has means for authoring the interaction, modelling the domain, authoring the teaching model, and authoring the student model. The Eon has been used in building tutors for different domains (Murray 1996*b*). Examples include tutors for foreign language learning and chemistry.

3.3.3 Support for special needs

When considering special education at large, the question of supporting *access* to the software is of utmost importance. There is often a need for some extra-ordinary input and output. These are discussed thoroughly by Edwards (1995). The possibility of using a single-switch input with automatic scanning of choices would be a solution for almost every motorical impairment; less radical limitations for input are sometimes sufficient. Therefore, the optimal solution would be that the method of input is an adaptable property of the system.

Moreover, there are other types of disabilities to support as well. Sensory disabilities can be supported by offering various modalities. Supporting mental deficits is also important yet difficult, since mental disabilities come in various forms.

Challenges. It should technically be relatively straightforward to implement single-switch scanning to many of the existing learning systems. However, some of the systems that could or even should be used in a special education context cannot be used with single-switch scanning of choices. There are two standard situations. First, if the system relies on typing in text, it is often too tedious for a learner with a single switch to use such a facility. Another case where a single switch is not feasible enough, is direct manipulation by pointing and clicking in a graphical interface.

Since it is not always a necessity to have a single-switch input even with the learners with mental programming deficiencies, the environment where every action is a form of multiple choice can be non-optimal, not in theory but in practice. Although in theory the expressive power is the same as single-switch multiple choice, the usability issues may not be as feasible as in cases with direct text input or pointing-and-clicking. Let us consider collaborative problem solving. The three standard ways people usually collaborate in a computer-based environment are by voice, by typing text messages, or by using a common "whiteboard". None of these methods can be used with single-switch input.

The obvious challenge of the single-switch is that it complicates the use of a system by slowing the pace of the process. However, it is more of a challenge, both educational and technical, to find out how to support and intensify a constructive learning process in such a context. Despite the restrictions of the input, the learner needs as flexible and rich a learning environment as possible, helping him or her to enjoy and move on in the landscape of constructing knowledge. The learning environment should incorporate the characteristics of unlimited personal self-expression and available technical aids into a functional combination.

At the first glance, it may seem that the use of a single switch restricts the freedom of the learner. In theory, every computerized environment allows its user a limited number of choices at one time. Typing on a keyboard is a series of multiple choices, and a screen can recognize a mouse click at any of its hundreds of thousands of pixels, a form of very wide multiple choice. Technically, the challenge is to offer the learner, bound to a very limited number of choices, an environment which enlarges his or her choices in terms of time.

Considering sensory disabilities, learning systems rely only on visual and auditory information. The challenge is that the functionality should not diminish and the usability should not suffer when a user can rely only on some sensory channels.

Mental disabilities are the most challenging disabilities from the learning systems perspective. Exact reasons for many mental disabilities are not necessarily known, and there is no consensus on how to tackle them. Mental disabilities vary hugely, and they appear combined in multiple disabilities. Mental disabilities can also correct over time. Therefore, it could be a property to be adapted in a learning system. One possible way to help deficits in mental programming is to support problem solving in a specific way: by partitioning the task into smaller sub-tasks, to lead the learner to a next logical step (as in Nakano et al. (2002)) or by metacognitive scaffolding (as in Hammerton (2002)).

Examples. There are various examples of systems with single-switch input with scanning. Most of them include the possibility of using another input device (e.g. a mouse for pointing to the desired choice). They also include the possibility of changing the waiting time at each step. In some relevant cases where the number of possible choices is large, such as the alphabet, the scan-lines are ordered hierarchically, so that the first click selects a row, and after that a proper column is chosen. This enhances the usability of a system by reducing the waiting time but it doubles the clicks needed to make a choice. For sensory disabilities, different modalities offer possible solution as well as enlargening text (e.g. screen magnifier systems) or screen readers. Visual output is far more heavily used in learning systems, so transforming visual output to auditory output is not often required. However, after the learning systems started to be based on multimedia, even the auditorial information can be crucial in the system.

An example of a system for mental disabilities is the TAPA (Training with Animated Pedagogical Agents) system (Tebarth et al. 2000, Mohamad et al. 2002). TAPA is a system to train meta-memory strategies by life-like agents in the Web. The system has four main training components for training specific memory strategies, for gaining knowledge about meta-memory, to improve perception of performance/self-efficacy, and to train attention and concentration. These training components are embedded into a story-line where the user has to help a gorilla and a magician to rescue their friend from the evil Zorgo-Skeleton (Mohamad et al. 2002).

TAPA includes a way to receive motivational information from the user. A special panel is designed, called fun-o-meter, for the user to input his or her current level of motivation. The agent tries to verify or falsify the information with additional questions from the user (Mohamad et al. 2002).

There are actually very few examples of software made for merely special education. However, the software for regular education can be used for special education if certain prerequisites are in order. These prerequisites differ according to the disablement. To name an example, severe motorical impairments demand single-switch input; dysphasia poses very different demands. The same applies *vice versa*, special education software can also normally be used when teaching according to the regular curriculum.

3.4 Classification of desirable properties in learning systems

As we have mentioned before, a learning system for special education could benefit from individual adaptation and openness in the learning domain. Adaptivity is needed because of the variety of users and their needs, and openness in the domain partly because of the small markets (no one can produce profitable one-purpose educational software as Soloway (1998) points out), and partly to support adaptation to individuals (personal curricula vary more in special education than in regular education). Moreover, to use a learning system in a special education setting, the systems should have support for various disabilities. An example of such support is single-switch input with scanning choices for motorically disabled users.

We are now ready to present the three dimensions in this classification (Table 3.1). The individual learning experience dimension (first row) develops from no individualization to both adaptive and adaptable systems. Therefore, we can say that there are more uses for a system, when the system is farther right on the horizontal axis. The column title **Adaptable** means that there are modifications a user/learner can make, run-time or beforehand, that affect the presentation or the flow of the learning session, to modify the system to be better suitable for a learner/user. The column title **Adaptive** means autonomous adaptation of the presentation of the learning content or adaptation of the flow of the learning session. The adaptation is based on the actions of the learner, thus requiring some form of *learner modelling*. The column title **No individualization** means that the system is neither adaptive nor adaptable, and the column title **Adaptive and adaptable** means that the system has both adaptable properties and autonomous adaptation.

Adaptation comes in a variety of forms, but meaningful adaptation can be achieved with different approaches. Therefore, the classification does not separate how the adaptation is achieved (i.e. are the teaching strategy or the learner model modifiable, and does the system support problem solving or help in navigation).

The openness in learning content dimension is presented in the second row in Table 3.1. The column title **Domain-dependent** stands for case where alterations to the learning content of a system would demand programming skills (i.e. the user needs the source code for making the alterations and the system would have to be re-compiled). The column title **Modifiable within a domain** means a slightly more open system, where material can be altered without tampering with the system around it. In these cases, the form of the content is well-defined, therefore the alterations are limited to replacing or adding the content similar to previous instances of content. The column title **Domain-independent** stands for a truly open system, where the content is not fixed by any means, i.e. the whole content can be replaced by a person not capable of conventional computer programming. It is a common conception that, to have an honestly sound pedagogical basis, a system cannot be completely domain-independent but suitable only for a class of domains (Dooley et al. 1995, Murray 1996*b*). This adjustment is made in our classification, too. If a system can be used for various domains, such as chemistry, mathematics, foreign language learning, it can be considered domain-independent. The fourth column, and the most valuable property for a learning system, is **domain-independence with ease of authoring**. Ease of authoring means both conceptual and technical ease for the learning content author. Conceptual ease means that the model and the way of representing the learning content is not too difficult to understand. Technical ease means that the actual generation of the material does not require, for example, programming skills. Of course, authoring tools or visual editors help to ease the authoring process.

The third dimension, support for special needs, is represented in the third row in Table 3.1. The first step into the desirable direction is that the system does not offer any kind of support for special needs. The next steps increase the amount of support so that the second column offers support for one disability, the third column for two disabilities, and the last column for three disabilities. Disabilities are divided into fairly large categories for the purpose, namely to motorical, sensory and mental disabilities. It is common that a system for special education has one or even two types of disabilities taken into account, mostly visual and motorical disabilities. Of course, many of the existing systems are conceptually ready to support single-switch input with scanning, i.e. the interaction between the system and the learner is such that single-switch input is possible, so they could be modified to support single-switch input with scanning. Support for mental disabilities are often tailored to a specific purpose, such as dyslexic children.

It is notable that already the concept of adaptive educational hypermedia could support special needs in particular ways. For example, sensory deficits can be supported by adaptive presentation, if the adaptation is widened to the different modalities, colors and font sizes instead of the traditionally used showing and hiding of extra explanations. Motorical support can be offered since link anchors are separate and can be scanned for single-switch input. Mental support could be achieved by limiting visible links (reducing amount of simultaneous options) or offering different forms of learning content.

Considering learning systems used at schools today, most of these off-the-shelf systems are completely domain-dependent and not adaptive. Polished as they usually are, they are trivial from computer science point-of-view. However, especially in the field of special education, some of the modern systems fall under the category domain-dependent but adaptable. This is because the needs of different users have been acknowledged, and it has not been too tedious a task to incorporate different preferences for different users. Examples include adjustable font sizes and scanning speed for single-switch input. However, the content is not open in these systems.

In software for special education, it is quite normal that the user can, for example in a program for foreign language learning, modify the system by inserting additional words into the vocabulary along with their visual representations and pronunciations. These systems are modifiable within the domain. In some highly “advanced” systems, the presentation of the

	Modest	**Desirable 1**	**Desirable 2**	**Desirable 3**
Individual learning experience	No individualization, same experience for every user	Adaptable	Adaptive	Adaptive and adaptable
Openness in learning content	Domain-dependent	Modifiable within the domain	Domain-independent	Domain-independent and easy authoring
Support for special needs	No support for disabilities	Support for motorical, sensory **or** mental disability	Support for two of the three disability categories	Support for all three disability categories

Table 3.1: Desirable properties of educational software.

learning material can even be adaptable. This is the case with e.g. MathPlusToolBox (Gardner & Lundquist 1998), a system to teach mathematical processes and allowing access for learners with various disabilities. The content in MathPlusToolBox can be modified but there is no built-in support for motorical impairments. In the field of *special education*, very few systems go further than this in dimensions concerning adaptation and domain independence. For example, TAPA (Mohamad et al. 2002) has powerful adaptation but is domain-dependent. The same applies to an intelligent tutoring system called AnimalWatch, which has been extended to accommodate learners with attention problems in the field of mathematics (Eliot et al. 2001).

In the field of non-disabled education, there are a lot of systems to employ full domain independence. Actually, after the emergence of World-Wide Web and easy-to-use authoring tools to author Web-pages, Web-courses have witnessed tremendous popularity. Normally, Web-based learning environments do not offer any kind of adaptability, but rudimentary adaptability can be included into a Web-environment without any effort: properties of a Web-based system can be adapted with the browser by changing font sizes and colors. More significant adaptability in a Web-based learning system can be made by the content author by preparing different sets of pages for different users. In such a case, a support for e.g. screen readers can be incorporated. Also single-switch input could be supported with a specially designed Web browser. In such cases, these special browsers do not normally offer support for the latest extensions to the Web.

After a paradigm change from ITS to AEH, the popularity of adaptive and domain-independent systems was evident. It should be noted that even today, the vast majority of these systems are laboratory prototypes and not used extensively in real-world settings. Some of them could face success also on the educational technology market. Regarding our intended users, there is no support for special needs in these systems; the scope of research is often higher education in AEH whereas the ITS paradigm is still very much alive in elementary learning (e.g. Nakano et al. (2002) and Hammerton (2002)).

The state-of-the-art system in many ways is the Web-based hybrid of an intelligent tutoring system and an adaptive educational hypermedia ELM-ART, since it includes autonomous adaptation, interactive problem solving support and preference-based adaptability. However, even ELM-ART is domain dependent and it does not have built-in support for special needs.

Support for mental programming. One of the interesting questions is how do these systems support mental programming. Even if the systems are domain-independent and adaptive, they do not support special education, especially supporting mental programming, because curriculum sequencing or task sequencing does not go into atomic operations, i.e. deep enough, and because the partitioning of the tasks is not enough and there is a need for upholding attentiveness. When supporting mental programming, we should keep in mind that mental programming deficits often occur with motorical impairments. Therefore, the systems should support or at least should be able to be modified to support single-switch input. Unfortunately, this is not the case even in the advanced systems developed.

As a conclusion for the chapter, we can say that there are no systems that 1) provide individualized learning experiences, 2) can be used in various domains, and 3) support special needs *to the fullest.* There is a gap left in the field to be filled.

In the following chapter, a model and a learning system employing the model are presented that try to address the issues described in the classification of desired properties, including ease-of-use for teachers and other experts to implement learning materials for the system.

Chapter 4

Learning space model

4.1 Definition of learning space

Although the use of computer software in special education is well-reported, the use of adaptive systems in special education is not common. Even though standard one-size-fits-all systems have proven to be useful (Kachelhoffer 1996), special education settings could benefit from an individualization provided by adaptive learning environments. It has been said that being able to personalize a system makes the difference whether people with disabilities can use the system at all (Treviranus 2002). Even if there is a solution (e.g. in human-computer interaction), it often serves just one disability group but fails to cater the needs of others (DeMeglio et al. 2002).

On the other hand, research around adaptive systems for non-disabled learning has been diverse. Contemporary systems use often adaptive hypermedia to provide individualized learning. In adaptive hypermedia (Brusilovsky 1996, Brusilovsky 2001), the presentation of pages is autonomically adapted and the navigation is supported by organizing or annotating links. In this chapter, however, we propose a different model to enable individualized learning sessions. The strength of the model lies in the domain independence, expressive power, straightforward use and computational simplicity, as well as easily enabled evaluational help for the teacher. The model is based on organizing the learning material into a *vector space*, where every piece of a learning material has a distinct position in the space. These pieces of learning material are called *learning seeds* (or *seeds* for short) in the model. The seeds can contain various types of learning material such as texts or multimedia, as well as interactive elements such as games, simulators, quizzes, exercises, and tests.

A vector space is a mathematical structure that has been used in a similar fashion in computer science. An example of usage of vector spaces in computer science is information retrieval, originally proposed by Salton et al. (1975). The vector space model has been widely utilized in the field (see e.g. Gravano et al. (1999)).

The vector-space model for information retrieval has a fundamental difference compared to the learning space model: the learner does not know what information to look for. There is also difference with the learning space model and standard hypermedia structure. The difference is that there are no direct links between the nodes (seeds) in the model. The seeds have their position in the space, defined by a vector containing a numerical parameter for every dimension. Formally, every seed s in learning space S is located by the corresponding n-dimensional ($n \in \mathbb{N}$) vector, called the *position* of s, denoted as

$$s = (s_1, s_2, ..., s_n),$$

where $s_i \in \mathbb{Z}$.

As a simple example, a learning space consisting of basic arithmetic might have two dimensions, namely "Addition" and "Subtraction". In such a case, the first exercise "1+1" might have a position of $(0, 0)$ along these dimensions (Fig. 4.1).

The learner is represented by a point in the learning space S at a given time t, $t = 1, 2, 3, ...$ In other words, the learner's location $s(t)$ at time t is indicated by a vector

$$s(t) = (s_1(t), s_2(t), ..., s_n(t)),$$

where $s_i(t) \in \mathbb{Z}$.

It should be noted that time t does not represent actual time; rather, time refers to discrete steps on a learner's path from one seed to another. To continue the previous example, a learner conducting the arithmetic exercises could be located in learning space S on a point $(0, 0)$, say, at time 1.

The seeds are thus positioned into the learning space similar to the way information retrieval entities (documents) are indexed in the vector space. However, in Salton's model (Salton et al. 1975), the key issue is to scatter entities in the space so that they can be retrieved efficiently, whereas the learning space model addresses guiding the learner to meaningful positions in the learning space. The guiding from one seed to another is conducted by assigning different *effects* for every action in a seed. An action refers to the choice a learner makes in a given seed s of space S. An action has an effect for the learner's position in the learning space. The effect can pertain to 0

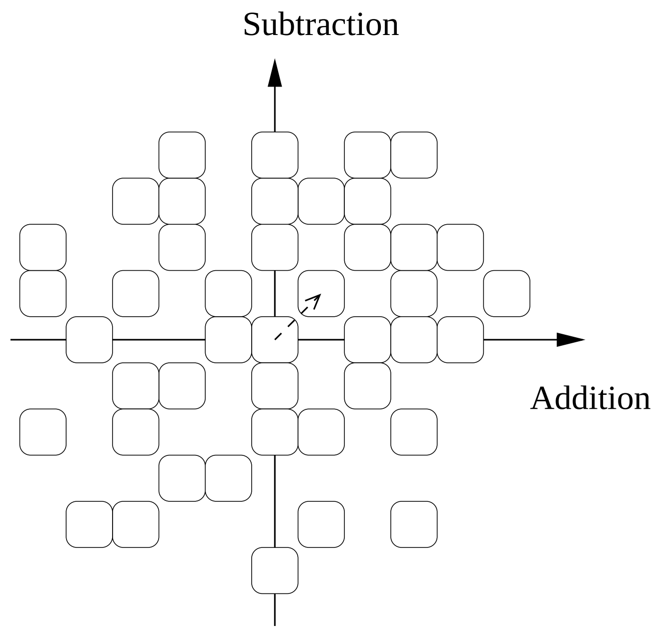

Figure 4.1: Parameters are used in situating the learning material in the learning material space. For simplicity, the space is only two-dimensional.

to n dimensions, and the strength of the effect can be arbitrary. Therefore, a negative effect is also possible. Thus, for every action a within a given seed s, the effect, or movement, is

$$\delta(s,a) = (\delta_1(s,a), \delta_2(s,a), ..., \delta_n(s,a)),$$

where $\delta_i(s,a) \in \mathbb{Z}$. For example, an action (in this example, the answer) "2" for assignment "1+1" could be parameterized to have an effect of $\delta((0,0), \text{“}2'') = (+1, +0)$ at the learner's position $(0,0)$ in the learning space. This means that the effect depends not only on the action, but also on the position of the seed. In this particular case, the learner is considered to be ready to proceed one step along the "Addition" dimension but subtraction skills cannot be evaluated and therefore the effect for the "Subtraction" dimension is zero.

The action a learner takes within a seed s moves the learner to the seed that matches the learner's previous point in the space added with the effect from the last action a. Therefore, at a given time $t+1$, the learner's location $s(t+1)$ in the learning space is

$$s(t{+}1) = (s_i(t){+}\delta_i(s(t),a))_{i=1,\dots,n} = (s_1(t){+}\delta_1(s(t),a), ..., s_n(t){+}\delta_n(s(t),a)),$$

or, in practice, the seed closest to the new potential location $s(t+1)$. Thus, if a learner in the learning space is located at point $(0,0)$ and makes a choice with effect $(+1,+1)$, he or she is taken to point $(1,1)$, as seen in Fig. 4.1.

During a session, the learner's actions are recorded. Formally, for a given learner u, the learning process record $p(t,u)$ at a given time t is a sequence of actions

$$p(t,u) = (s(1,u), a(1,u), a(2,u), ..., a(t-1,u), a(t,u)),$$

where $s(1,u)$ is the location of the first learning seed on learner u's learning path and $a(h,u)$ refers to the action learner u performed at time h. This record of a learner forms the individual path through the learning space. Therefore, the learner is not modeled as a mere point in the space but as a trail through the space, including the whole history of his or her learning process. This recorded trail is a *profile* of the learner, which can be represented visually and can be used for evaluational purposes.

A practical addition to the model is not to require that the learning space is *complete*, i.e. at least one seed in every point in the space (Fig. 4.2). This relaxing of rules is necessary, since in reality there are easily "holes" in the learning space. Therefore, a system using learning space model should have a method for moving the learner to the nearest position in the space where there is a seed. Several approaches could be taken to decide where the learner should go in a case where there are no seeds in the actual position. Straightforward and conceptually intuitive solution is to calculate the Euclidian distance from the learner's position to the seeds close by, and to take the learner to a seed where the Euclidian distance is the shortest. That is, the learner u is taken to a seed s, where the Euclidian distance between the learner profile p and the seed s

$$Euc(p,s) = \sqrt{\sum_{i=1}^{n}(p_i - s_i)^2}$$

is the shortest.

An important addition to the learning space model is that every seed in a learning space can consist of an arbitrary amount of seeds within that seed, connected to each other with traditional hyperlinks, thus forming a *collection of seeds* (Fig. 4.3). This approach is chosen because it adds simplicity when authoring the learning material, but also because it enables easy preparation of ready-made sub-problems to original problems that might be too challenging for the intended users. Supporting subtasking of problems is in complete harmony with supporting mental programming of learners.

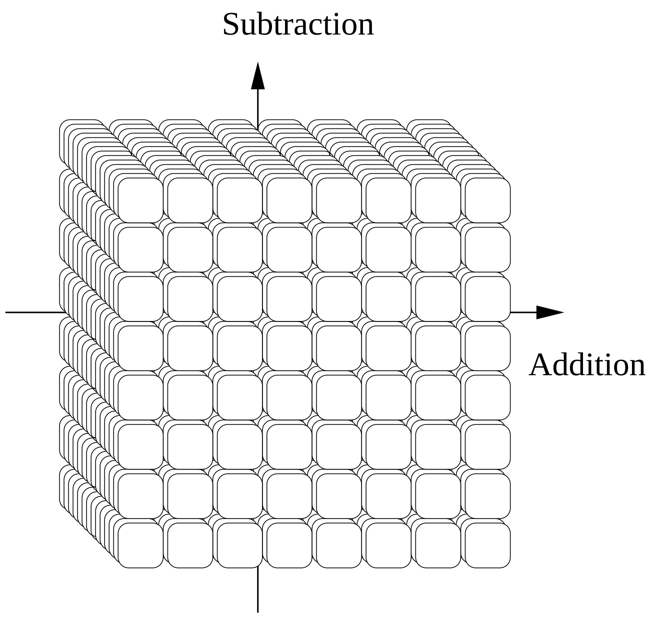

Figure 4.2: An example of a simplified but complete learning space, where every point has several learning seeds.

Every learning seed should have a position for every dimension used. However, it is not necessary in the learning space model. For example, if the dimensions used in the space are “Addition” and “Subtraction”, some seeds can be defined in the “Subtraction” dimension only. In that case, the seeds without an explicit location in “Addition” would be represented as a line covering every point in the “Addition” dimension. This procedure ensures that a person creating the learning material does not have to assign irrelevant parameters to the material. For example, if the learning content for one learning seed is “5+2”, there is no purpose in assigning parameters in the “Subtraction” dimension to those exercises. In a way, the seed can be very far and very near at the same time, as seen in Fig. 4.4. Using Euclidian distance to move to the next point is still a valid method to calculate the next point.

One additional issue is worth noticing in the learning space model not being complete. If the author has designed seeds for only some positions of the dimension, the seeds “fill in” the gap because of the Euclidian distance technique described above. That is, the seeds next to the gap in the space are larger in the metric used. Figure 4.5 shows this with an example when using one-dimensional learning space. When there are no seeds between

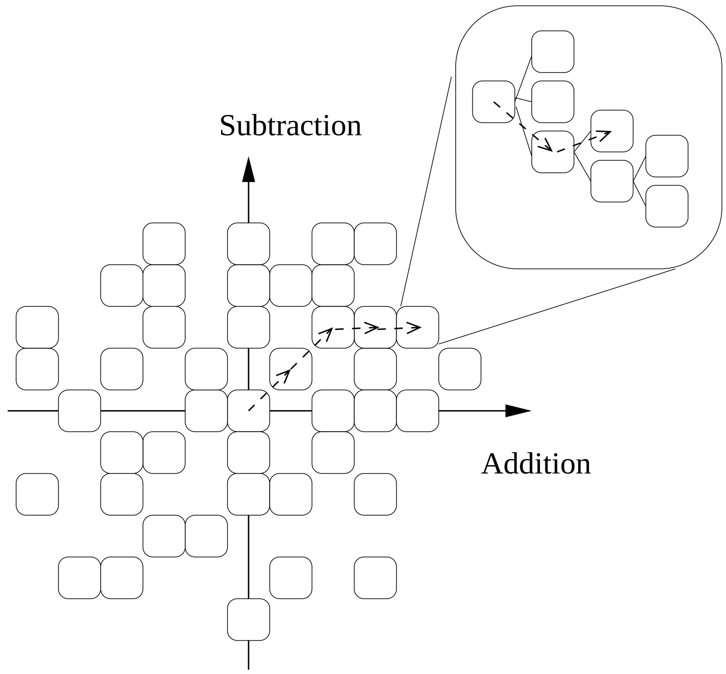

Figure 4.3: A learning material space with occasional static graphs (some seeds have absolute links to other seeds).

points 3 and 7, the seeds in those points are enlargened in the sense that they can be said to fill in the gap operationally, although in reality they stay in points 3 and 7. If the author later places seeds for points 4, 5 and 6, the gap vanishes. Figure 4.6 shows an example of the same event for a two-dimensional space. The seed covering only its actual point is located on the top-left corner in the visualization; the others are enlargened in a manner the figure presents. Of course, the same applies in higher dimensions as well.

The learning space model is completely domain-independent. It means that the dimensions and their parameters are freely selectable for the person authoring the material in the sense that they can be represented with arbitrary strings. It is natural to denote the dimensions with meaningful learning objectives such as “Multiplication skill” rather than abstract ones. In a typical way of viewing the dimensions, the dimensions of the learning space can be viewed as learning objectives, and a learner strives to achieve a higher position in every dimension.

The domain independence of the learning space model allows one to think the dimensions to be other than learning objectives. For example,

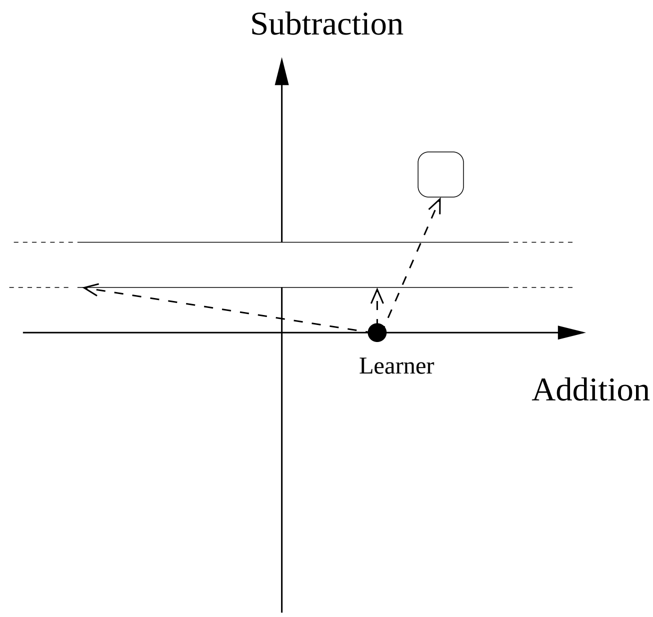

Figure 4.4: An example of a case where one seed is defined only in "Subtraction" dimension but not in "Addition" dimension.

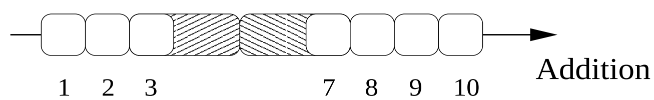

Figure 4.5: Missing learning seeds cause "enlargening" in one-dimensional learning space.

there can be a game where the learner starts from the origin and loses "life-points" according to the actions taken, thus dropping into a negative value-area in the learning space.

In addition, dimensions are not necessarily related to any objective, but they might indicate some other aspect of the learning process as well, or only those learner's properties that a teacher wants to observe. As an example of a property to observe, we can consider "Motivation". To enable the measurement of motivation, every learning seed can contain the button "I don't care". This button is equipped with an effect (relative movement) to the "Motivation" dimension. But, because none of the actual seeds is parameterized with a "Motivation" value, choosing "I don't care" does not have an effect on the movement of the learner in the learning space. Instead, it affects only the learner's profile: every time the learner presses

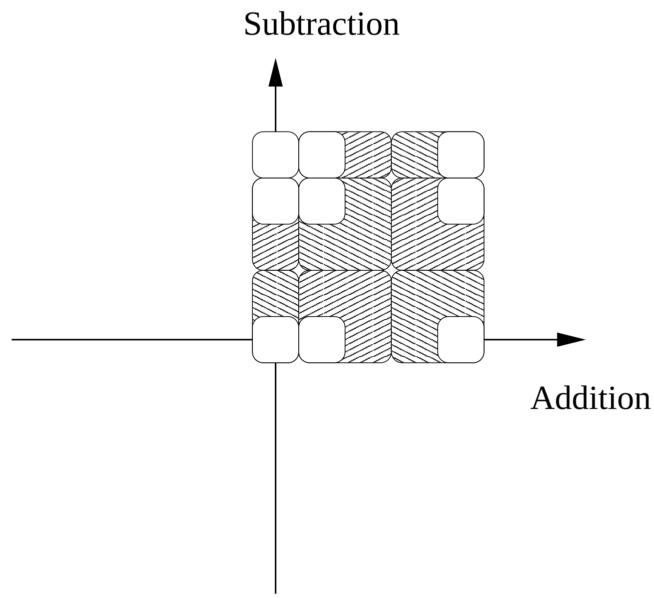

Figure 4.6: Missing learning seeds cause "enlargening" in two-dimensional learning space.

the button, the action is recorded to the learner's profile, thus enabling one potential value for the teacher to observe.

It should be emphasized that to operate properly, a learning space should have more than one dimension. If only one dimension is used, the functionality of the model is restricted to varying the step-size for every action. Therefore, multiple dimensions are needed in meaningful learning spaces.

4.2 Characteristics of the learning space from the authoring point-of-view

Authoring learning materials has never been trivial, but with the presented learning space model, the author is confronted with additional challenges compared to traditional learning material authoring. The learning space model leaves the responsibility of the learning material completely to the person authoring the learning seeds and positioning the seeds into the learning space. Because of the model, there are several properties considering the process of authoring the learning content that are of major importance:

What kind of dimensions to use. Although — or because — the dimensions are freely selectable, the author must define every dimension used. This ensures domain-independence but makes the dimensions easily incompatible with each other between the different content authors and different learning domains.

Positions of the seeds in the learning space. The relative positioning of the seeds in the space (i.e., the positioning within every dimension) has to be decided by the learning material author and is far from trivial. The seeds have to have numerical values for their n dimensions, although seeds may be defined in only some of their dimensions. This means that in the model, even a nominal dimension has to correspond to a numerical value. However, positioning the seeds along the dimensions becomes easier if the dimensions have been chosen appropriately.

Actions within the seeds. The learning material author is also responsible for defining the actions a learner can make in a given seed. Let us assume that a seed contains some task for a learner to complete. The choices presented to the user are either correct or incorrect. In an optimal case, when the choice is correct, it is correct in a way that it has an effect in relation to the learning objective. In other words, the choice can be interpreted to demonstrate a certain skill acquired or knowledge gained. And vice versa, in a case where the learner chooses an incorrect action, every incorrect action is incorrect in a distinctive way so that the effect of that particular action has a certain relation pertaining to the learning objectives. It should be noted that there can be more than one correct choice as well as more than one incorrect choice. In a certain type of learning material such as educational adventures, the choices can all be "correct" because they describe various ways to cope with the given task.

Effect of every action. As well as the actions, the learning material author must also define the effect of every action. In many cases, it is not clear how a choice relates to a certain dimension. To be more precise, even to know which dimensions an action affects is a difficult issue. The definition of the effect to every action becomes easier, if the actions themselves are chosen properly. In practice, to make the authoring of the content easier, the effects for various actions can be fixed throughout the learning material to be, for example, $+1$, 0 or -1 to every appropriate dimension.

Although the material itself is static, the parameterization can be altered manually when feedback from learners is received. This means adjusting the learning space is a process of formative evaluation (Kurhila et al. 1998).

4.3 Discussion of the schema

When comparing the learning space model to the well-known and traditional four-component schema of intelligent tutoring systems (Wenger 1987), we can identify the same four concepts in the learning space model, as well. The domain knowledge equals the contents of the seeds in a learning space. The pedagogical model corresponds to the dimensions used, the parameterization of the seeds, and the parameterization of the actions in the seeds. The student model is the trail (history of every visited seed and actions taken in them) of every individual learner. The interface model is the organization of the objects in seeds, since the learning space model does not pose any restrictions on the learning material.

The domain-independence of the system enables various possibilities. There can be exercises with right or wrong answers, but there also can be more adventurous problem solving with e.g. ethical values. An example of more adventurous problem solving could be to guide a child (or another living being the learner can relate to) to visit her Grandmother in a countryside, and there are no right or wrong answers, only different ways to cope with the problems until the final goal is achieved. Whereas an intelligent tutoring system might make the grandchild to hurry to the Grandma in the optimal time, with the minimum amount of money consumed, our system allows the child to observe the path as well, even with wolves along the route.

The learning space model as proposed in this chapter offers a versatile environment for which the teacher can relatively easily prepare his or her own materials, despite of the challenges described in the previous section. The organization of the material in the learning material space and the tracking of the actions of the users in the material provide a way to use the system in a novel manner compared to many existing adaptive hypermedia-based learning environments: the progress of every learner can be derived from the use of the system for evaluational purposes.

Today, the trend is to have simple adaptation in a learning environment. For example, Ketamo's (2002) work on Adaptive Geometry Game has shown that technically simple adaptation can improve learning outcomes, and relatively complex systems, such as Stern's (2001) iMANIC, do not show greater outcomes.

Because the model has been developed for special education and especially for tackling deficits in mental programming, we have relaxed the property of learner-driven learning. This is necessary since deficits in mental programming means that too many choices at once can cause difficulties, so that the amount of simultaneous choices has to be limited. However, the design of the learning space model allows the learner be in the center of the learning process and be the active agent. The model has been developed for special education, but there is an obvious transfer to non-disabled education, especially in elementary education.

4.4 Learning space model in relation to tutoring systems

Learner modelling in tutoring systems. Traditional intelligent tutoring systems based on cognitive modelling use computational models in explicating mental mechanisms if the mechanisms cannot be observed directly from the human behaviour in experiment settings. In addition to *ad hoc* models, there are general cognitive architectures that build upon the idea of a unified theory of cognition (Anderson 1993, Newell 1990). These simulation models contain a number of properties discovered in experimental psychological research on various domains and various levels of human cognition, including memory processes, learning, attention, natural language processing, problem solving and decision making.

These general cognitive architectures involve strong assumptions about the properties of various cognitive phenomena and the architecture that gives rise to these phenomena. Among these assumptions are memory structures and strategies: the distinction between short-term and long-term memory and between declarative and procedural memory (Anderson 1993).

There is a general problem if human mental processes are to be realized in computer software. Since the implementational or extra-theoretical assumptions are not deliberately or purposely involved in the process, their influence is difficult to analyse, and in some cases, even difficult to detect. The assumptions are incorporated into the model design along an attempt to facilitate the encoding process, the intelligibility of the system's functionality and the interpretation of the system's behaviour. It demands interpretation and encoding to transform human cognitive phenomena into a computer program, and more interpretation when translating a program's behaviour into cognitive terms.

The above mentioned problems of computerized cognitive models concern two types of intelligent tutoring systems. The first systems are those that model the cognitive development of the learner and adjust the tasks and direct the learner's progression according to the model used (Nwana 1993*a*, Weber 1996). The latter systems model the optimal problem solver that initially learns the rule set and acquires needed concepts to accomplish the goals in restricted problem solving (Anderson 1995).

Systems that do not model the ideal learner (into which the learner is forced) but adjust the instructions according to the model of the learner's abilities involve several theoretical and practical assumptions. There is a problem in resolving what and how the learner really thinks, even if the external behaviour may be traced into simple elements. Moreover, the evidence that complex adaptive problem solving support is any better than the one consisting of static, pre-made support frames, is arguable (Brusilovsky & Eklund 1998*b*).

Normally, in intelligent tutoring systems the learner is considered to have learnt the desired skills when he or she possesses the same set of production rules that the optimal problem solver would use. This method of tutoring can be seen as authoritative, since the performance of the learner is evaluated in respect to the optimal production set. It is obvious that the system does not encourage creative problem solving, because when the learner's possible deviations from the optimal solving route are immediately detected, the learner is assisted back to utilize pre-defined solution strategies, and to produce strictly determined solution outcomes.

Systems using this kind of model-tracing approach make strong assumptions about the acquisition of knowledge and development of complex skills (Anderson et al. 1990). Especially strong assumptions concern the strategies about how the problems should be solved, what knowledge is used and how that knowledge is used, as the required competence is formulated as production rules which are compared to the solution steps taken by the learner.

Of course, there are models and theories that do not make assumptions on skills or knowledge types, and do not differentiate e.g. procedural and declarative memory. One of them is the knowledge space theory (Doignon & Falmagne 1985, Falmagne et al. 1990), originally proposed for adaptive assessment of knowledge but applied to adaptive hypertext and other adaptive tutoring systems as well (Albert & Hockemeyer 1997, Albert & Hockemeyer 2002, Dowling et al. 1996).

In knowledge space theory, the knowledge state of an individual equals to the set of problems an individual is capable of solving. The set of all knowledge states forms a knowledge space. The problems are presented to the learner in an adaptive manner, since there are prerequisite relationships between the problems; prerequisite relationships defines the structure of knowledge in a given domain structure (Dowling et al. 1996, Albert & Hockemeyer 1997). Prerequisite relationships are obtained from domain experts by a querying procedure (Falmagne et al. 1990, Dowling et al. 1996).

Although there are no logical assumptions in knowledge space theory as model-tracing tutors, there are several simplifications that affect the application of the theory. For example, the learning rate is thought to be constant, and the responses of the learner are either correct or incorrect.

Differences between the learning space model and intelligent tutoring systems. There is a reason for many of the decision solutions in intelligent tutoring systems. Since the tutoring systems have often served as testbeds for cognitive theories, they also have other than pedagogical aims. Model-tracing tutoring systems involve higher-level goals to model the target skills necessary in solving certain problems (e.g. what are the essential components of knowledge and solution strategies when struggling with a limited task structure), or more generally, in what way novel knowledge is constructed upon the existing knowledge.

Systems using the learning space model are different from traditional intelligent tutoring systems because the model does not make any assumptions on the learner's cognitive skills nor the optimal problem solving strategies. Instead, the learning space model incorporates means to model various aspects of the user, depending on the learning material author. Therefore, the whole responsibility for the meaningful and pedagogically sound learning material authoring lies on a human expert.

The concept of a knowledge space is particularly interesting since it is closer to the learning space model than the model-tracing tutors. For example, the concept of a learning path is present also in the knowlegde space theory (Falmagne 1993, Albert & Hockemeyer 1997). The knowledge space theory provides means to test the learner accurately and tries to optimize the number of questions needed to evaluate the knowledge of the learner by eliminating redundant questions based on the prerequisite information. It is of importance that the structuring of the knowledge is successful.

The deepest difference between the learning space model and the knowledge space theory is the underlying pedagogical view. In the learning space model the key issue is to support the learners in their learning processes. The approach taken in the learning space model is that the learning process is by no means optimized in terms of time and effort used, so there is no harm done if the learner takes a detour and is guided to face unexpected learning experiences. In fact, the learner should go deep to the unexplored areas that can contain even harsh learning experiences (into a *jagged study* *zone*, as proposed by Gerdt et al. (2002)). Therefore, it is not necessary that the knowledge representation is as accurate as possible in the learning space model.

Differences in cognitive presuppositions. Intelligent tutoring systems based on a unified cognitive architecture are built upon the assumptions of Newell & Simon's (1972) symbol manipulating paradigm that all mental activity can be formulated as problem-solving and implemented in a rule-based system, and the learner is able to state his or her own goals and sub-goals, and execute some search in restricted problem space. They presume that the learner possesses an ability for long-span and goal-oriented behaviour, deliberate decision-making and autonomous self-evaluation. The systems are practically suitable for instruction in subject domains with well-defined structure so that the optimal solution path can be easily constructed, and the solving strategies can be distinctly stated in a goal-oriented rule-based formalism.

The learning space model contains no theoretical assumptions as to psychological theory, optimal behaviour or necessary competence components, since the model is aimed at special learners; they cannot be held responsible for their learning in the same way contemporary learning approaches suggest. They may not be capable of stating their own goals and sub-goals, they do not have long-span behaviour, deliberate decision-making or capability for self-evaluation. Therefore, the learning space model enables the construction of a novel "theory" for each individual learner and for each type of learning material. The learning space model necessitates the teacher's assistance in authoring learning objectives, adaptation parameters and strategies, and monitoring the learner's progress. Instead of the acceptance of some well established theory of cognition, the model enables independent exploration of single, precisely defined cognitive skills for each individual and for each type of learning material.

The benefit of this procedure is that the teacher does not (necessarily) need to be concerned with the underlying psychological theories and the presuppositions they hide, but he or she can concentrate on specific elements of the learner's yet to be achieved abilities, by only defining dimensions along which the learner is carefully guided. These dimensions can be some general cognitive skills, such as attention or reasoning, or some domain specific abilities, such as mental calculation, processing of perceptual information or natural language comprehension.

Intelligent tutoring systems with strong learner modelling are restricted to operate only on a narrow subject domain: the learner model constructed is hard to generalize to other topics, and the optimal strategies cannot be used in other domains. The learning space model enables the creation of freely selectable dimensions and learning material parameterization individually for each learning space. Therefore, the learning space model conveniently suits ill-structured problems and badly defined tasks. The description language for the learning material and the adaptation method do not limit the functionality and the potential of the system to well-structured problem domains.

4.5 Description of AHMED

Ahmed is a prototype system to employ the learning space model. Therefore the system serves an environment for the learners that offers adaptation to individual's learning process and at the same time is domain-independent. Ahmed is a system only for the learner. The learning environment consists of other tools as well, described later in this section. The logical structure of a learning environment using Ahmed and other tools is presented in Fig. 4.7.

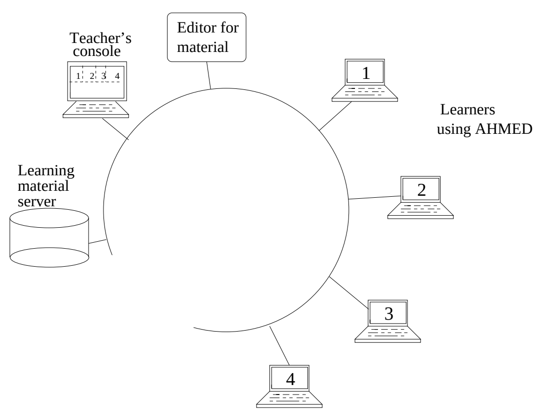

Figure 4.7: The logical structure of a learning environment using Ahmed.

User point-of-view. When the learner starts a learning session with Ahmed, the first thing is to login. Because of the intended users, single-switch input with automatic scanning of choices with adaptable step-time is available throughout the learning session. Therefore, both the login procedure and answering the exercises are designed to function with singleswitch input. The login procedure is conducted as follows. The learner starting the session is presented with the pictures of her class. The learner chooses a picture representing her (it can be a photo or a drawing). To ensure that none of the learners chooses the wrong picture and thus an incorrect profile, the learner has to choose her "secret picture". This corresponds to the password. It is also possible to require a sequence of secret pictures to diminish the probability of logging in with false pretenses. The pictures are in a single directory, so that it is fairly easy for the teacher to update the pictures of learners attending her class.

After the login procedure, the learner chooses the desired learning space from the possible options. If the learner is using the system for the first time, he or she will start from the point $(0, 0)$. If the learner has used the system before, he or she will start from the seed presented last.

The user interface of Ahmed consists of four system buttons always visible on the screen (Fig. 4.8). From left to right, they are Hint, Help, Stop, and Proceed. The proceed-button (marked "OK") is pressed when the feedback from the seed is received and the learner is ready to proceed to the next seed. The stop-button exits the learning space and takes the learner back to the login screen. The help-button alerts the teacher with a fixed message. The hint-button is enabled only if there is a hint(s) authored to the seed the learner is currently viewing.

Figure 4.8: The user interface in Ahmed presenting an arbitrary learning seed.

The use of Ahmed differs from some other systems for learning, since every action the user can make is restricted to be a multiple-choice selection. Therefore, most of the direct manipulation operations of graphical user interfaces, such as drag-and-drop, are not possible in Ahmed. This is a choice made on purpose, since the intended users might be able to use input devices with only one switch. Apart from this restriction, the learning material in AHMED can be almost anything. This may seem rigid at first, but in fact every input the user makes in whatever computer system is a form of a multiple choice, and every computer-constructed world has only a finite set of states. The real challenge is to allow learners to express themselves with as few, but relevant, choices as possible.

Description language. The author's point-of-view to AHMED is different. The author must prepare the learning seeds by using a description language defined for the purpose. The description language is based on Extended Markup Language, XML. Compared to e.g. HTML, the description language is significantly more expressive. It includes tags for buttons, graphics, text areas, and audio clips. Conditioning with an `if-then-else` structure enhances functionality and enables various effects for the actions and different responses or feedback for different actions.

Temporality is an essential feature in the learning space model utilized in AHMED. Elements in the seeds can have an attribute for duration. Constructs borrowed from the description language for synchronized multimedia, SMIL, namely `PAR`- and `SEQ`-elements, allow presentation of various media elements either in parallel or sequentially.

A description of a learning seed consists of three different operative elements. The two key elements are `EX` and `FEEDBACK`. An `EX` element consists of the presentation of the learning material (exercise or other content). In the `FEEDBACK` element, the actions of a learner are checked. A typical action is an answer to a multiple-choice question. Feedback to the learner is defined in that element. The feedback may require actions from the learner, or it can be e.g. another exercise.

The description language also allows to prepare different hints to the learning material. `HINT` differs from `EX` and `FEEDBACK` elements, because it does not support similar functionality than the two other elements: hints are meant to be pop-up windows showing text for a certain period of time without a possibility to an action. If a seed has several hints, they are shown to the learner in the order specified in the description every time the learner presses the hint-button. This ensures that an exercise can have different levels or types of hints built into it.

There are different attributes that are or can be used in conjunction with the tags. For example, multiple-choice answers can be assigned with values. These values can be summed up and the progression of a learner can be affected by those sums. This value adding is used e.g. in an exercise where a learner has to pay a bus fare by picking up a correct amount of different coins. Each coin is assigned with a value. If the subset the learner chooses has a smaller or greater value than he or she was asked to pick, the feedback or the next exercise can take this information into account. Thus, the mechanism results in a more sensitive feedback method.

After a learner has chosen the desired learning space, Ahmed reads in the positions for the seeds in that learning space. Every seed has a header that tells the position of the seed. For example, if a heading of an exercise is

```
<EEL>
    <HEAD title="Exercise1">
        <AINFO name="Skill" value="5"/>
        <AINFO name="Knowledge" value="4"/>
    </HEAD>
    ...
</EEL>
```

that exercise is situated in point $(5, 4)$ in a two-dimensional learning material space.

The effect of every action is defined using the same `AINFO` field. For example, pressing a button with id “correct” can have `AINFO` fields as follows,

```
<IF>
    <GET target="correct"/>
</IF>
<THEN>
    <AINFO name="Skill" value="+1"/>
    <AINFO name="Knowledge" value="+2"/>
</THEN>
```

has the effect of $(+1, +2)$ in the learner’s position in the space.

If multiple learning spaces have common dimensions, progress made in one dimension transfers to the next, when the learner starts a learning session in the next space.

Practical additions. Ordered subsets can also be picked up in a seed by a singleswitch input with scanning of choices. Still, it is possible for the learner to correct a misprint before giving the final answer. The ordered subset is a kind of “stack”, where re-selecting the last choice cancels it. Allowing correction is important, because misprints in answers are hard to interpret correctly by a computer. In addition, it can be demotivating for a learner not to be able to correct an accidental misprint.

As mentioned in the Section describing the learning space model, there is a practical need to allow the space to be not complete. Using the Euclidian distance to select the nearest seed was presented to solve the issue. Another practical addition is used in AHMED to ensure that the learner does not stay in the same seed forever[1]. The Euclidian distance to the nearest seed is weighted with a coefficient so that the user moves to another seed, even though the effect of the action suggests that the user should stay in the same seed. The coefficient C used in weighting the distance in the current implementation of AHMED is

$$C = \frac{A}{m(s) + \epsilon},$$

where $m(s)$ corresponds to how many other seeds has been presented after seed s has been presented last time, ϵ is a small constant to ensure that there is no division by zero, and A is a constant based on heuristics. Based on some experimentation, A is set to be 500 in the current implementation. The distance to the next seed comes from multiplying the distance $Euc(u, s)$ with the coefficient C; therefore, the next seed presented to the learner might not be the one with the shortest Euclidian distance.

Other features. Another feature in AHMED is the possibility of using learning material servers to store different learning spaces. The learner chooses the learning space when starting a session. Technically, every space is a directory. The directory containing the learning space can be situated in any remote Web server as well as in the local hard drive. This enables easy networking for the learners and centralized composing of the learning material.

To exploit the full potential of the schema, the learning material space should be at least nearly complete, i.e. filled with enough learning seeds, even more than one seed per position. In practice, this means significant effort in authoring the learning material. Therefore, centralized learning material databases are a choice to consider, to make the authoring a collaborative and co-operative effort.

To involve the teacher in the learning environment, a console for the teacher has also been implemented. If one of the learners presses the help button, a message is sent to the teacher's computer. One has to keep in mind that the learners in this learning environment can be severely disabled,

[1] Of course, there could be a need to present the same seed over and over again, but in many cases, the learning session would be more meaningful for the learner if the learning seed is not repeated.

so that they cannot draw the attention of the teacher by any other means. The message is merely help, because the schema was designed to be used in a classroom with a teacher always present. Technically the help functions properly even if the learning environment is expanded to the whole Internet, since every help message comes with the IP address of the sender's machine.

Another tool for the teacher is a system to follow the processes of each individual learner. The functions supported allow the teacher to:

- view learners logged into the Ahmed environment
- graphically view the progress of a learner for one dimension in a timeline
- view the progress of a learner in every dimension numerically
- send a seed directly to the learner, thus overriding the seed according to the learning space model.

Two editors to speed up the authoring of seeds have been implemented as well. They both support the drag-and-drop principle in constructing the seeds, offer a view to the description language presentation of the seed. However, both of them are for constructing individual seeds, so they do not offer support for organizing (or re-organizing) the space.

Chapter 5

Learning materials for the learning space model

5.1 Types of learning material suitable for learning spaces

Domain-independence of the learning space model means that the learning content is not fixed. However, the learning space model is also expressive in the sense that the structuring of the learning content has various possibilities. This chapter reviews how different types of educational material can be incorporated into the learning space model. The second part of this chapter discusses how mental programming deficits can be supported by the learning space model. The chapter is concluded by a section discussing the possibility of using the learning space model as a test-bed for neuropsychological assessment.

5.1.1 Frame-based computer-assisted instruction

Although the learning space model is designed to be more than a traditional frame-based computer-assisted instruction system, it can be used for that purpose as well. The seeds in a learning space would then act as frames (Figs. 5.1 and 5.2). Although the content in the seeds can be anything, the standard use is to expose the learner to the problems to be solved. When using the learning space model, the problems may be traditional multi-part, with one seed leading to another with a direct link, or stand-alone, so that the next seed is presented to the learner according to the operation of the learning space model. Multi-part problems are such that a collection of problems form a coherent whole with pre-defined paths between the problems. This is the case when the person authoring the material wants to ensure that the order of certain problems is always the same for every user (as in Fig. 5.1). Stand-alone problems can be used in situations where the problems do not have references to other problems, since the order of presenting stand-alone problems can and most likely will be different for every user, depending on the actions the learner takes in the seeds and the effects authored to the actions (as in Fig. 5.2). In every learning space, there can be, of course, both multi-part and stand-alone problems.

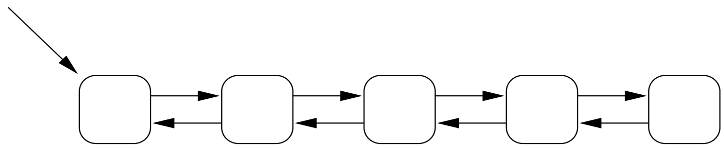

Figure 5.1: Standard computer-assisted instruction with a book metaphor, where every move corresponds to a page turn.

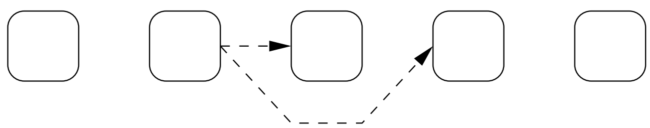

Figure 5.2: Standard computer-assisted instruction in a one-dimensional learning space, where the step-size varies according to the effect in every action.

A learning goal in traditional computer-assisted instruction might be that of dealing with real-world problem solving tasks. For example, often the most significant asset of special education for a disabled person is to learn how to handle everyday life. Therefore, we have prepared learning material with which a person can learn to handle money and public transportation systems. The learner has to pick a right amount of coins to pay a bus fare, for example.

5.1.2 Educational hypermedia

It is, of course, possible that the seeds in a learning space are linked together as in traditional hypermedia structure, thus allowing the learner to have complete control over the navigation (Fig. 5.3). Technically, a traditional hypermedia structure can be done with the learning space model by linking the learning seeds directly to other learning seeds. The description language allows using text or figures as well as other elements to serve as link anchors. The learning seeds in the learning space can form arbitrary graphs, with one or two-way links between the nodes. However, this is not the most appropriate solution for the targeted learners, since such a degree of freedom as in standard hypermedia navigation can be an inhibiting factor for persons with difficulties in mental programming.

If the learning space is constructed as traditional hyperspace, the choices for navigation can still be assigned with parameters, and those parameters can be used for other purposes than guiding the learner through the learning material. For instance, there can be some traceable features of the user that should be followed. The values are then stored in the user profile.

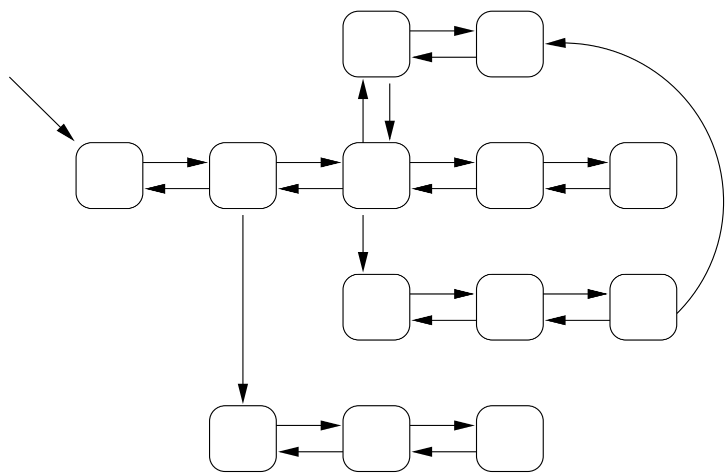

Figure 5.3: Standard educational hypermedia, where every concept is linked to some other, starting from a basic concept.

5.1.3 Adaptive educational hypermedia

In some cases, there might be a need for nominal dimensions. There are ways to use the learning space model for nominal values in dimensions, even though they are ordinal dimensions by definition. The first, trivial possibility is to test at some point what the value is for a certain dimension and fix the dimension to that value. In this case, numerical values can correspond to some nominal values.

Suppose one wants to have the same learning material as in Figure 5.3 but with presentation variants for different learning styles (as in Kelly & Tangney (2002)). Possible classes of learners could be learners who are primarily literate, illiterate but with strong visual skills, or illiterate but with strong auditorial skills. The same learning material with different presentations of every seed is prepared for these groups, to be used and presented depending on each learner's profile. The learning space should, in this case, have a "Learning style" dimension with nominal parameters for three different values, namely literacy skills, visual skills, and aural skills (Fig. 5.4).

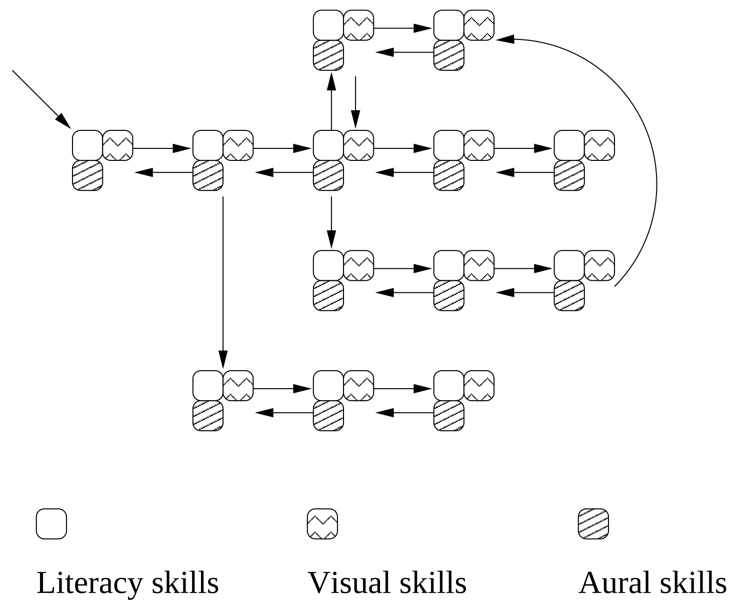

Figure 5.4: A visualization of educational hypermedia where each learning seed has three different presentations literate, visually strong or auditorially strong learners.

However, the example above shows the trivial approach which is not very usable if one wants to be able to switch from one presentation style to another more than once. If this is the case in the example above, there is a risk that adding or subtracing values for the learning style dimension makes it to correspond wrong nominal value sooner or later. Better use of nominal parameters is to have *binary dimensions* for every type of nominal value, so that the values for each dimension can be adjusted independently. For example, adaptive presentation described above could be achieved with three binary dimensions, namely literacy skills, visual skills, and aural skills. Of course, the dimensions can have more values than two, for example corresponding values such as "none", "some", "good" and "excellent". The drawback when using nominal dimensions this way (one dimension for every nominal value) is that the number of dimensions is easily multiplied, thus multiplying the amount of seed to be authored. In a case with three binary dimension, the number of seeds is multiplied by 8. Sometimes it is, of course, possible that parts of the learning space can be empty, if there is no need for seeds in some combination of nominal values.

In practice, the learning space model does not allow adaptive presentation in a way typical in some other adaptive hypermedia systems, where the presentation is constructed from snippets for every screen. Theoretically, the same functionality is possible, but it would require authoring of nearly infinite amount of seeds. It is worth noticing that serving the primary target group does not necessarily require fine-grained adaptation; what is needed is a way to cater for varying needs, since the abilities vary hugely from an individual to another, and the clusters of "similar" persons are small.

To fully exploit the possibilities of the learning space model, there should not be absolute linking between each learning seed (as in Fig. 5.4). Adaptive presentation with three binary dimension for presentation described above operates properly with relative linking, too. Using the example in the previous subsection, we can add three (binary) nominal dimensions to the previously mentioned learning space with dimensions for "Skills" and "Knowledge". Then, the learning space matches the one presented in Fig. 5.5.

The preparation of the different representations is more tedious than in computer-generated adaptive hypermedia systems. However, we postulate that the adaptive presentation mechanism works better if a human expert has designed every seed manually. This is due to the potential exceptions in every meaningful learning material. However, authoring the learning material could be helped with various semi-automaic tools depending on the domain.

Adaptive navigation support could be done similarily to adaptive presentation, but because of the intended users and the learning space model, there is no real need for navigation support in common adaptive hypermedia systems. Every move in the learning environment is a type of adaptive navigation support since the user is taken to next meaningful seed. In a way, the learning space model provides an adaptive NEXT-button for the most suitable material for the person[1]. In addition, the whole operation of the learning space model contradicts the use of standard adaptive navigation support, since taking the user from a seed to another is already a form of adaptive navigation support. The idea of presenting the most suitable material, according to the user profile, is consistent with the idea of

[1] In a study by Brusilovsky & Eklund (1998*a*), over 90% of the time learners used the unannotated NEXT-button and more or less neglected the adaptively annotated suggestions for recommended links.

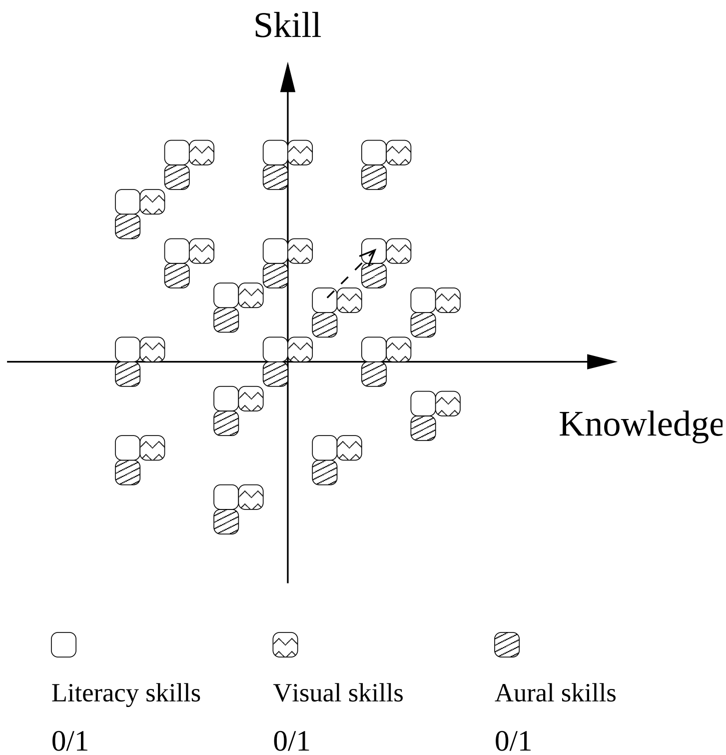

Figure 5.5: A visualization of a learning space where every learning seed has three binary dimensions for literate, visually strong or auditorially strong learners.

helping the users' mental programming, and overrides the completely free navigation which would bind the user not to act at all and lose the freedom altogether.

Adaptive hypermedia might be exploited to some extent for the intended users. Possibly, an ideal learning space consists of several fairly limited separate hyperspaces (with absolute linking), so small that our learners are not overwhelmed to navigate in them, and at some point the learner is taken to a neighboring small hyperspace. The next small hyperspace is chosen in one of the end-nodes of the small hyperspace by the parameters in the users' profile. One of the parameters can be a "counter", which decides where the learner jumps after reaching the end-node. Thus, the learning path in the global learning space occurs step by step in local neighborhoods which can be expanded or narrowed depending on the learner's orientation. This is a novel way to think adaptive hypermedia.

5.1.4 Learning through games

Slowly-paced educational computer games, where the number of states in a game is limited, can also be constructed with the learning space model. Inherently suitable types of games for the model are adventures, quizzes or simulators. Incorporating games into a learning environment can have a positive impact on the motivation of the learners (see for example Järvinen (1999) and Turkle (1996)), and using games as a part of an adaptive learning environment has provided positive results (Carro et al. 2002).

Adventures can be text-based or graphical. The key here is to arrange the learning seeds as in educational hypermedia or adaptive educational hypermedia, but create the seeds to contain the information about the situation of the simulated world. Then the learning seeds represent locations in an adventure game. The transfers between the adventure locations should be authored as absolute links to ensure that the world stays coherent and does not have unanticipated jumps. Dimensions in the space and the effect in the multiple-choices in the seeds represent the information and its change pertaining to the user's state in the game: the amount of money, health, "lives" etc. The parameterisation can also be used for other changing states of the player, for example different items carried by the player's character. Of course, building a traditional adventure game this way is straightforward and uninteresting using the learning space model, since the movements between seeds are standard absolute links. However, also relative linking can be exploited in adventure games. Using relative linking poses different challenges for the author of the game, since the author cannot predict all the orders the learner is likely to visit the seeds. If this is the case with the game, the seeds of that game cannot have direct references in them to other seeds, e.g. how many lives you have or what has happened previously.

The learning space model does not pose restrictions for using the model for action games. The restrictions can from the description language used for describing the learning seeds. However, certain types of action games are possible since the description language for the seeds includes duration-attribute. To name an example, there can be a "thing" moving on the screen (within a single learning seed), and the player's task is to catch it by clicking it with a mouse. Technically, this is possible if the thing flying around is a button to be clicked in a seed. However, introducing this type of game is in contradiction with our intended learner group: catching a flying thing with single-switch input and scanning of choices is not an easy task.

Quizzes are perhaps the best types of game for the model and the description language, since it is straightforward to create them – at least compared to action games – and quizzes can be highly educational because of their motivational property. One dimension along its parameterisation can be used in calculating and presenting points acquired, and the open structuring in learning spaces enables, for example, sub-quizzes where the learner is transferred if certain events (or a collection of events) have been triggered.

5.1.5 An example: building a simulator using learning seeds

A collection of learning seeds or the whole learning space can be constructed to be more than the mere sum of plain CAI frames. An example of such a learning space is a simple simulator where each state of the simulated phenomenon forms a seed of its own; together they form a simulator that can clarify a complex concept.

For the purpose of our users, the simulator should be simplified somehow because of the learning disabilities caused by deficiencies in mental programming. Let us say that we are trying to teach the ecosystem in a restricted environment by having a world with two kinds of animals, rabbits and foxes. When there are a lot of rabbits, the foxes have plenty of food supply, so the number of foxes starts to grow. At some point the foxes consume more rabbits than the ecosystem can hold. Therefore, the amount of foxes is bound to diminish, since their food supply is not sufficient. This procedure forms a kind of a balance where the number of foxes and the number of rabbits are alternating unless a human interferes with the ecosystem by excessive hunting of the animals.

In the simulator, the user can hunt either one of the animals (decrease the amount of rabbits or foxes), and just let the time pass. Proceeding from one time stage to another can be a user-initiated function or self-evolving step in the learning space model. In a simulator, every state of the simulated world can be presented in a seed. The problem is that it requires a vast amount of seeds, even if the simulator is restricted, since the general formula for the amount of seeds needed in a simulator is $x_1 * x_2 * ... * x_n$, where n is the amount of dimensions (the variables in the simulated world), and x_i equals the amount of different states in i dimension. However, if the author of the simulator wants the system to keep track of a learner's characteristics in addition to states in the simulated world to enable individual adaptation based on the learner's actions in the simulator, the amount of dimensions grows even larger.

Our example ecosystem (Fig. 5.6) has different states as follows:

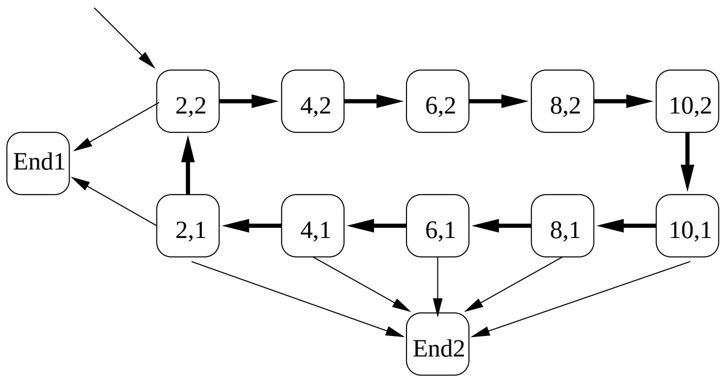

Figure 5.6: A nearly-trivial ecosystem simulator with ten regular states and two ending states. The numbers in the seeds represent the amount of rabbits and foxes.

- The amount of rabbits can vary from 2 to 10 with the interval of 2 animals, giving a total of 5 states for rabbits.
- The amount of foxes can vary from 1 to 2 with the interval of 1 animal, giving a total of 2 states for foxes.

For the sake of clarity, only the time elapsing movements and "game-ending" movements are presented in Fig. 5.6, and not the links to other states when hunting the animals. The simulator stops in a situation where the ecosystem cannot sustain hunting an animal, i.e. when the amount of that particular animal is at its lowest point (rabbits two, foxes one). The transitions between the states presented with **bold** arrows are the transitions if the learner just lets the time pass. In such a case, the ecosystem is in an eternal loop. Only human interference with the ecosystem (i.e. hunting the animals) can end the game.

This simple simulator has $5 * 2$ states in the simulated world. Therefore, one has to build 10 seeds into a learning space to have such a simulator. The more realistic the simulator is, the more seeds must be prepared. When it comes to learning, the example above is not large enough to be feasible in learning the concept. One can imagine a simulator should have something like $20 * 50$ states so that a learner can have a good grasp of how the world is evolving. The amount of seeds would in that case be 1,000.

In practice, few thousand seeds in a learning space can still be manageable, but there is a limit at some point. Therefore, the learning space model can only be used in restricted simulators, even though the seeds could be created with semi-automatic content generators. However, the restrictions in simulators are not necessarily problematic. Common wisdom in early learning is that you should start with a simple phenomenon and remove it from any unnecessary noise.

5.2 Characteristics of learning material in the learning space model

Although the typical ways used in everyday educational software can be exploited with the learning space model, the total domain-independence of the schema allows variations in learning seeds and in collections of seeds, regardless of the learning content organization.

- Reflective material has links to the seeds which ask questions such as "Do you understand?" or, "Do you know what this means?" which in turn have links to the next appropriate seeds.
- Alternatives in the style of the learning material. The learner can be forced to stay in a seed until the correct answer is found, by linking the wrong answers to the same seed, either straight or through a loop of seeds. Another extreme is that wrong or inadequate answers will be passed without any notification to the learner.
- Allowing fuzzy input, like "I don't care". Since every input is stored into the user's profile, this type of fuzzy input can be used by a teacher or significant other in evaluating extra-curricular learner properties, such as motivation, after a learning session.
- Non-factual exercises where the line between correct and erroneous answers is vague. Exercises can contain problems with ethical or moral values, and the whole learning material can consist of an adventure in an imaginary world.
- Nested seeds can cover more profound learning objectives. Assume that a seed contains a simple problem to be solved. Thus, a complex problem can be represented as a sequence or even a tree of simple problem seeds linked together. This hierarchical structure of seeds yields a more profound evaluation of learning, compared to using only one seed. Another use of "sequential" objectives is the analysis of error types. As a simple example, consider multiplication and addition. If a learner answers $1 * 1$ with 2, it is possible that he or she confuses $1 * 1$ with $1 + 1$. By presenting similar exercises the system provides a way to draw a conclusion that the learner does not

know how to multiply. From the learning theories' point of view, one can also assess learning outcomes as the level of acquired automation. During the learning process, the learner packages sequences of small seeds into a combined structure, and the level of mastery can be estimated as the size of these automated hierarchical structures. For example, most people do not partition a sum sequence $1 + 2 + 3$ into smaller chunks but consider it as one chunk containing 6.

5.3 Learning material to support brain deficits

The underlying principle behind the learning space model has been serving people with disabilities and especially deficits in mental programming. In this section the use of the learning space model in supporting various brain deficits is presented. The model behind the human brain comes from the Russian neuropsychologist Luria.

5.3.1 Luria's model of working brain

In neuropsychology, *cognitive process* is a term used to refer to those complex activities involved in receiving, processing, maintaining, storing and using information. *Processing of information* means individual's processes of thinking and drawing conclusions.

The cognitive processes of humans consist of complex functional systems, which cannot be located to a specific brain area; to produce these functions, various functional units are needed. A functional unit consists of the areas that act together to produce a certain kind of behavior (Luria 1973). However, the human brain always works as a whole when receiving and adapting information, developing directions for how to act, and controlling the resulting activities (Luria 1979).

Luria's dynamic localization theory for mental activity defines three active units that comprise the functional structure of the brain. Each unit can be shown to have its own share in the organization of the mental activity of an individual (Luria 1973, Luria 1979). According to Luria, the three functional units of the brain are 1) a unit for regulating the tone or waking, 2) a unit for obtaining, processing and storing information (from the outside world), and 3) a unit for programming, regulating and verifying mental activity. These descriptions are approximations, and the model has been refined later (see for example (Vilkki 1995, Vilkki 1990, Virsu 1991)).

1st Unit: regulating tone and waking and mental states. The task of the first functional unit is to maintain an optimal state of alertness and awareness in the cortex, i.e., *cortical tone*. It is up to this unit to set the brains for action, thereby enabling the wanted mental action. The adjustment of alertness and awareness is activated by many different new and important stimuli, both internal and external. Changes in the stimulus environment produce a reaction that sets up the individual for action. Purposes, plans and goals are typical activation sources for humans (Kuikka et al. 1991). When the cortical tone weakens, the cortex may reach a state where weak stimuli cause the same kind of reactions as strong ones. In such a situation, structured and conscious action is impossible and selective, and organized thinking becomes random (Luria 1979).

According to Kuikka et al. (1991), disorders in the first functional unit may influence the exactness of observing stimuli. In such a case, the individual may misrecognize and misinterpret the stimulus. If the maintenance of attention is disturbed, the individual is not able to carry out long-term activities that require accuracy. Disorders in attentiveness are due to problems with regulating the state of alertness (van de Meere 1996). Attentiveness disorder is one of the most common problems in children's neuropsychology. Usually, this means problems with maintaining, directing and dividing attention.

2nd Unit: receiving, analyzing and storing information. The task of the second functional unit is to receive, analyze and store information. In this functional unit, sensory signals from the outer world are analyzed and synthesized (Luria 1973, Luria 1979). The system of the second functional unit may be described as *modally specific*, since information reception, processing, and preservation in memory is done in this unit (Luria 1979).

With disorders in the second functional unit, the learner may have trouble with complex processing and interpretation of information (Luria 1979). In connection with damages to the second functional unit, there may also be partial disorders in attentivity, where the general state of alertness remains normal, but the individual finds it difficult to divide his or her attention on several matters simultaneously. Problems in observation may also be caused by perception disorders. In case of misperception, it is difficult for the individual to find critical goals in the field of perception, or to switch attention to new details. Disorders in the second functional unit may also cause difficulties in understanding speech (Kuikka et al. 1991).

3rd Unit: programming, regulation and verification of activity. It is the task of the third functional unit to program, adjust and control all mental activity (Luria 1979). The structure of the third functional unit is multiple and it co-functions closely with the first and second functional unit. In relation to the first functional unit, the third functional unit regulates the activation level of the individual. The activity goals and plans of the individual affect the activation level of the cortex. The third functional unit also adjusts the information reception and processing areas of the second functional unit to a suitable level of readiness according to the requirements of the anticipated task. This unit constantly monitors the realization of each current activity at every moment. It supervises that the activity is concurrent with the intention, and corrects it if it is not. The motor areas of the third functional unit convey the individual's movement commands. The premotor area takes care of the fluency of movements. Disorders in this area cause trouble with voluntary movements. Series of movements may be clumsy or transferring from one series of movements to another may be difficult. The frontal lobe is the third functional unit's area of higher combination (Kuikka et al. 1991).

Problems with the functions of the third functional unit cause various disturbances in target-orientation, and difficulties in the adjustment and fluidity of voluntary movements. The adjustment of personal activation may also be difficult. Typical features of the activities of an individual with disorders in the third activity unit are problems with the managing of cognitive information. The individual may have problems with constructing the information and planning activity stages. Outlining a new task may also cause problems, because it is difficult to comprehend the whole out of the different parts of information. Especially damage to the frontal lobe causes the individual to have problems with keeping to an intention. A learner moves too easily from one task to another instead, not necessarily being able to finish the task independently. An individual suffering from damage to the frontal lobe has difficulties directing his or her voluntary attention, since outer impulses may catch the individual's attention, directing it to something else than the task at hand. He or she may repeat the same mistakes, and may have trouble changing behavior patterns once learned (Kuikka et al. 1991).

5.3.2 Attracting lost attention

Guided by the third functional unit, the first unit regulates the level of consciousness and alertness (Kuikka et al. 1991). One may attempt to influence a lowered state of alertness by bringing forth new points of view, for example, or by encouraging another try. These methods of support can be incorporated directly into the learning space model by linking inaccurate answers directly to a seed with a different viewpoint.

The system should be able to strip every additional visual cue if necessary, because attentiveness may be distracted by inappropriate stimuli. An especially unnecessary stimulus is an unforeseen movement on the screen. Also, there should be no more information than necessary if the learner has severe deficits in the functioning of the first brain unit. This is the reason why e.g. visual helpers, such as Adele (Rickel & Johnson 1997), are clearly unusable for learners with this type of brain deficits. The same reasons apply also to some other support mechanisms, such as *social navigation* (Munro et al. 1999) and various forms of goal- or case-based learning (e.g. Schank et al. (1993)).

A child with an attentiveness disorder may be assisted in his or her self-instructability by e.g. asking the learner to pause before attempting the task, repeat the task in his own words, planning different methods of solving it aloud, and foresee the consequences of the planned solution. The learner may also be taught specific strategies for solving problems, such as directing the attention to central items aiding in solving the problem, or teaching strategies for retrieving information from memory (Sandberg 1999). Although it is difficult for a computer to recognize and interpret whether the learner has, for example, repeated the task in his own words or planned solutions aloud, such requests can be included into the learning material because of the domain-independent description language.

To support the functions of the first unit, a system using the learning space model could employ a "cortical tone support system", where the attention of the learner is drawn to the task at hand with rightly-timed *multi-modal* cues. This can be done if the learning material is prepared to have such cues, and there are appropriate dimensions in the learning space for such modalities. The attention span could then be a dimension in a learning space. That way, attention span would be an adaptive property. There is no direct support for upholding the attentiveness with *timed* support in the learning space model (but it can be incorporated into Ahmed by preparing similar seeds with different timed effects and placing them along one dimension). Timed support was in fact implemented in an earlier prototype version of Ahmed (see Kurhila & Sutinen (1998)), but the method of implementation was inappropriate: the pop-up window could easily distract more than help in bringing the attention back to the task at hand. The area of supporting attentiveness requires more research with extensive empirical validation before implementing a timed support model.

Because systems build upon the learning space model are likely to be used in a switched-on classroom with a teacher or some other support personnel present, another complicating issue for timed support rises. The situation in a classroom can be such that the teacher is explaining something to the class and the learners are not allowed to proceed with their computer-supported learning. Therefore, the interpretation of the time elapsed is difficult for a computer. Has the learner stopped functioning, or is he just waiting for the teacher's instructions? And if the learner is just slow, the "over-anxious" computer offering a parade of multi-modal cues gives a learner a feeling of being underestimated and lack of control.

5.3.3 Offering appropriate modalities

The second functional unit is responsible for reception, analysis and storage of the information. It includes visual, auditory and general sensory regions (Luria 1973). This unit is best supported by offering multiple representations with different modalities in the learning material, and adapting to the learner's needs autonomously according to the learner profile (i.e., the learner's individual history).

For example, the learning material can have alternative presentations by the use of a nominal dimensions for presentation style, as described in subection 5.1.3.

Since disorders in the second functional unit may also cause difficulties in understanding speech, it is important that the learning material does not rely solely on the speech modality. Since too much information in a seed for "just in case" is not necessarily a desirable property, the material for different modalities and mixtures of modalities should be prepared for every learning space.

5.3.4 Supporting the mental programming

The third unit ensures that a human not only passively reacts to incoming information, but creates intentions, forms plans and programs, inspects performance, regulates behavior and verifies conscious activity. The last task refers to the feedback mechanism: the learner compares the effects of his or her actions with respect to the original intentions and corrects the mistakes. Since mental programming is controlled by the third unit, the learning space model has been designed to support it more explicitly than other functional units.

The methodological solutions in the learning space model are made to particularly suit learners with mental programming deficits. In this section, we discuss the kind of material, which supports mental programming while enjoying a learning session in the learning environment. The goal is not to rehabilitate the deficits in mental programming, but to educate *despite* the deficits in mental programming. The general guideline to enable this is formulated by Vilkki (1995): "If the conditions of the task are flexible enough, it is possible to compensate disturbed operations with preserved ones by using effective [mental] programming."

The support for mental programming in the learning space model operates with two separate methods: 1) the possibility to partition a single task into subtasks, and 2) guidance of a learner into the area of the learning space where the appropriate support material for mental programming exists. These can be called *in-seed* and *in-space* support, respectively. The in-seed support is built locally within individual learning seeds (or rather collection of seeds), whereas the more comprehensive in-space support is based on the organization of the whole learning space and proper use of learning space dimensions.

In-seed support: Partitioning the tasks. The learning seeds containing a task supporting mental programming should be such that every task can be partitioned into simpler subtasks. The point of this *subtasking* is to find such a set of subtasks that an individual learner can do the task bottom-up from the subtasks. Because of the choices made in the learning space model, the partition must be made beforehand by the person authoring the content. While authoring the material seems laborious, it is the only way to guarantee that the tasks remain pedagogically sound and meaningful. However, in some well-defined domains such as arithmetic it is possible to help authoring by semi-automatic editors.

The partitioning should be made hierarchical, so that the subtasks can be further partitioned into finer subtasks, if needed. The simplest tasks should be such that every learner has enough operational resources to complete them.

To illustrate the concept of subtasking, we use a simple example domain, elementary arithmetic. If the learner has to evaluate a sum expression

```
3 + 2 + 1 + 0,
```

the first step of subtasking can be

```
3 + 2.
```

If the learner can accomplish the subtask, the next step in subtasking can be

```
    5 + 1
```

and the next after that

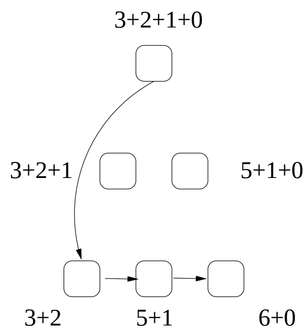

Figure 5.7: A visualization of subtasking, where the task is partitioned after the first wrong answer to the final partitioning.

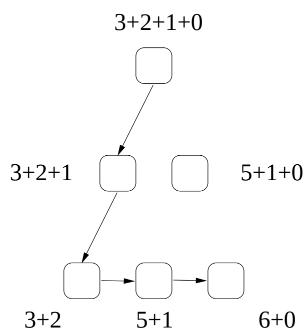

Figure 5.8: A visualization of subtasking, where the task is partitioned to the next level of hierarchy after the first wrong answer.

```
6 + 0.
```

This partitioning is equal to the visualization in Fig. 5.7. However, the subtasking can be done by another procedure. The example above immediately goes to the smallest partitioning (to the bottom of the partitioning tree in Fig. 5.7), in other words to $3 + 2$. Another method would be to partition the expression $3 + 2 + 1 + 0$ first to $3 + 2 + 1$, and after that (if the learner does not answer correctly) to the smallest possible partitioning (Fig. 5.8).

In figures 5.7 and 5.8, where the subtasking goes to the final level of partitioning, it stays there until the execution of the task. In some cases, it may be beneficial for the learner if the learner is allowed to try the "more difficult" level of partitioning after the initial wrong answer. Figure 5.9 visualizes such a case, where the learner is raised to the previous level of partitioning after visiting the final level, and figure 5.10 visualizes a case where the learner is presented with an upper-level partitioning after every

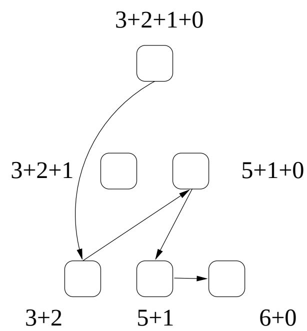

Figure 5.9: A visualization of subtasking, where the task is partitioned first to the final level, then raised to the previous level above.

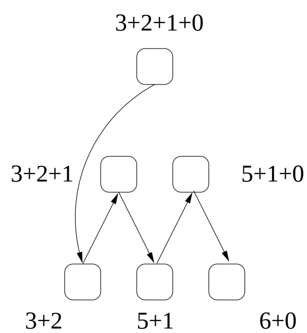

Figure 5.10: A visualization of subtasking, where the task is first partitioned to the final level, then a raise to the previous upper level is tried between each attempt on the final-level subtasks.

attempt on the final-level partitioning.

As we have seen in figures 5.7–5.10, the partitionings can be different with the same learning seed contents, since the seed contents can be the same, and only the linking is different. This helps the preparation of various partitionings. The partitionings presented in figures 5.7–5.10 all start subtasking the task from the beginning. However, there are no obstacles to starting the partitioning from the end, or even from the middle.

It should be noted that it would be easy to partition the tasks presented above automatically, but if the material is e.g. how to respect modern art, we need handcrafted partitionings.

Regardless of the partitioning styles, one issue remains: how can the learning space model adapt to various learners and their different needs in finding the best partitioning for them? The method is to use binary dimensions for subtasking-style such as “Straight-to-final-level” and “One-step-smaller" to represent the most suitable partitioning method for a specific learner. And, in a case where the first-chosen partitioning was not suitable, the next task can be partitioned differently.

To ensure that the adaptation works properly when creating the learning material into a learning space, the parameters and their values have to be consistent and accurate throughout the material. However, the person creating the material can choose these parameters according to the nature of the learning material. It is essential that each potential choice is thought through, so that the learner's selections give as much information as possible. For every error type, the learner can be taken to a different partition from the original task. This can be useful, but it requires careful planning of the input possibilities allowed for the learner when a person is authoring the learning material. However, it is clear that manually constructed helping paths may contain much more sophistication than an automatically generated one (as in Carro et al. (1999)). One possible approach is described in the next chapter, where Matinaut learning space was empirically evaluated in a classroom setting.

An important issue concerning the material supportive for deficits in mental programming is that because the need for partitioning the tasks into smaller subtasks exists, the original tasks chosen for the learning environment should be such that they *can* be partitioned. Fortunately, in many domains, this is automatically true. In any case, the subtasking needed will be different for different learners. Therefore, the subtasking should have various forms. However, the final stage of subtasking is to have the deepest partitioning possible, i.e. subtasking to atomic operations.

In some cases, the atomic operations for a task can be the same, although the previous partitionings are clearly different. The structuring of the learning material description language also supports cases of this kind, alleviating the task of preparing some partitionings. Figure 5.11 presents a visualization of such partitioning, where the original tasks (top-level) are different in some aspect because their positioning in the learning space is different in dimension C.

The creation of the learning seeds with various partitionings to enable optimal subtasking is tedious, regardless of the alleviations pointed out above. In a case where the domain is somewhat restricted and reasonably well-defined, it would be possible to have the partitioning made automatically, as well as the concatenation of the atomic operations to new unseen operations.

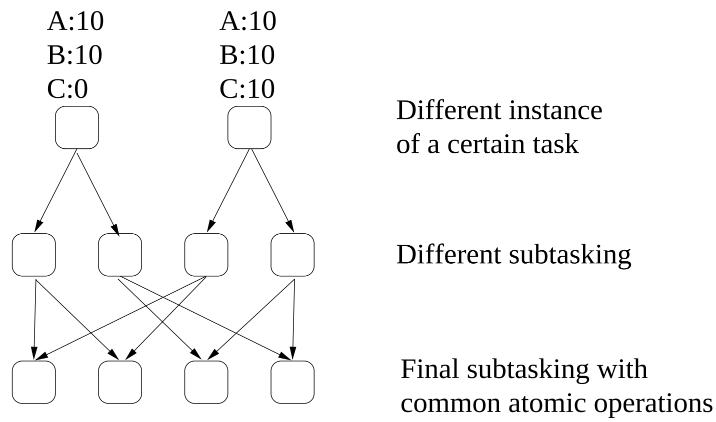

Figure 5.11: A visualization of two different learning seeds containing ultimately the same set of atomic operations.

In-space support: Guiding the learning process. The other way to support mental programming with the learning space model is to guide or steer the learner towards the area in the learning space where the seeds contain support for mental programming. This requires the use of the learning space dimensions in a slightly different manner, compared to using them as learning objectives or learning goals.

In a standard case, when using the dimensions as learning objectives, the learner strives toward a learning objective along that dimension. The better he or she performs in that dimension, the closer he or she is to the learning goals, thus indicating better performance on that particular learning objective. The objective can be e.g. multiplication, and the indication of the better performance in multiplication corresponds to the number of the correct answers to the exercises. In this case, the amount of correct answers can be combined with some other factor, such as the level of difficulty, to represent the learning objective better.

However, the dimensions can have a different use to the standard approach above. The dimension can indicate an area in which the learner needs support, and not a learning goal. This is the case especially in situations where the skills needed cannot be acquired completely by exposure to a learning environment. An example of such a situation is a lack of skills in spatial orientation. Deficits in spatial orientation often occur with *ischaemic attack* (Kuikka et al. 1991). When a deficit in spatial orientation is noticed (by presenting appropriate learning material), the learner can be moved along the dimension of "Spatial orientation" to an area, where the seeds (the tasks) do not require spatial orientation or there is additional support for spatial orientation built into the material.

There is also one additional way to support mental programming, not directly linked to guiding the learner in the appropriate area in the learning space. With empirical testing, it has become evident that in everyday life, the motives and values of an individual are in close relation to one's mental programming (Vilkki 1995). Therefore, it is important that the material a learner is exposed to is motivational. This is best acquired with values a learner can share, and possibly with characters or events a learner can relate to.

5.4 Testing the learners' cognitive abilities

One of the potential uses of the learning space model can be the assessment of a learner's cognitive skills in order to further design appropriate learning material. Long-span monitoring of the learner's development in different atomic and composite skills is possible, and a system using the learning space model can be used as an experimental aid in cognitive psychology or neuropsychology. The responsibility of interpreting the data produced by the use of the system and drawing conclusions is left to the person that has authored the learning or testing material. The learner model system produces (i.e. learner's trail in the learning space) is not based on any cognitive theory, neither does the system itself introduce any biases to the information that it provides. Furthermore, data can be collected during a long time span, so accidental slips committed by the learner or occasional defects in attentiveness do not affect the assessment results.

When using the learning space model, the assessment of cognitive functions and sub-functions can be done the way the tests are normally conducted, but without the distractions caused by the assessor's personal characteristics or other stressful factors present in ordinary psychological experiment situations. For example, subjects usually strive to perform better if they know that they are monitored and particular information of their performance is gathered for evaluation. The use of the learning space model enables data collection for study purposes as it provides a natural environment that is not prejudiced by artificial experiment settings and unnecessary cognitive load induced by extraordinary circumstances.

Especially, one hindrance of using typical learning software in assessing is that they often lack the possibility of guiding the tests according to the assessor's hypothesis. It is a standard procedure in neuropsychology first to make an orientation test, form a hypothesis, and conduct a deeper test based on the hypothesis (Korkman 1997). The learning space model provides the possibility to choose tests according to the learner's responses. In a way, the guidance is not based on the last response but the learner's whole history. The approach when using the learning space model for testing can thus be similar to the computer-adaptive testing paradigm (see e.g. Gouli et al. (2002) and Huang (1996)), where the test is stopped as soon as a sufficient conclusion can be drawn.

The learning space model enables a standardized test to be transferred into it, but in order to exploit the full potential of the model, one can develop completely novel test paradigms. Apparently, a slight drawback of novel test material is that it requires a significant amount of knowledge concerning cognitive functions in addition to the possibilities of learning space model to harness the full potential to assessment.

When testing children's cognitive functions and sub-functions, we can limit testing only to attentiveness and mental programming which are concepts commonly used in neuropsychology. Other cognitive components or more complex skills and properties that can be assessed with the learning space model include memory span, attention span, spatial orientation, conceptualization of time, facilities in natural language comprehension, difficulties in processing semantic and syntactic information both visually and auditorily. An example of using the learning space model for testing cognitive abilities is presented in the next chapter, where two of the skills needed for learning arithmetic have been tested empirically in a special education setting.

Chapter 6

Empirical studies

6.1 About the studies

It is well-known that a model such as the learning space model can only be validated through real-life use with actual users. Moreover, conducting empirical evaluations for learning systems is acknowledged to be labour-intensive (Stern 2001), which has been noted as a reason for rare evaluations (Brusilovsky & Eklund 1998*b*). In addition, empirical evaluations for models such as the learning space model have two distinctive parts: preparing the learning material and organizing the test setting. These reasons explain why only two small-scale empirical studies were possible to carry out to evaluate the functionality and potential of the learning space model in the context of this thesis.

The empirical evaluations presented in this chapter do not try to claim or validate that the learning space model is superior to some other models for similar purpose. The evaluations try to show that the scheme behind the model operates as designed. The role of the learning material is crucial in the learning space model, so the potential of the model cannot be validated by individual learning spaces; validation can only come through a long-term usage in varying contexts with a multitude of real learners.

Testing cognitive abilities with learners in a special school. The first empirical study was to test cognitive abilities. Deficiencies in mental programming were not tested explicitly in the empirical evaluation, because it was not feasible to arrange suitable test subjects. The learning space in the study was deliberaly very simple, and the idea behind the study was to compare the trails through the learning space to the neuropsychological evaluations of the learners to see if the learning space model could be employed in quick testing of some abilities for non-professional purposes.

The trails of the testees were not compared to each other. Therefore, the hypothesis is that the trails do not contradict the actual neuropsychological tests carried out in the school.

Support for partitioning the problems was not implemented in the test space, because the aim was to evaluate, not train or educate. The study was carried out in a special school with four main subjects. Such a small number of subjects suggests that the results have to be considered only indicative and validates more the operation of the model than the value of it.

The contents for the learning space used in the first empirical study were designed by consulting special teachers and neuropsychologists. The material was tested briefly with several learners in a special school to validate the decisions in presentation and actions taken in the learning seeds and the organization of the seeds into the space, before suitable test subjects (testees) were sought out.

Learning addition algorithm in elementary education. The second empirical study concentrated on elementary arithmetic. The aim was to teach and to rehearse the addition algorithm for learners in the second grade. The constructed learning space also offered support when needed. Two classes using the system were non-disabled elementary education.

Evaluating the effect on learning outcomes caused by the learning space model would be flaky at best, since the learning space model is not something that can be switched on or off for study and control groups. Even if the adaptation mechanism can be switched off for the control group, the results might not be valid since the "static" version is not designed to the best possible way, as de Bra (2000) points out. Therefore, the evaluation of the operation of the learning space model has to be based on some theoretical framework. Vygotsky's (1978) theory on zone of proximal development (ZPD) offers a suitable framework. The hypothesis for the second evaluation on the operation of the model is that the learners should progress rapidly to their ZPD, and after that, their progress is slowed but not stopped completely because of the seeds containing teaching material in the learning space constructed for the purpose.

The contents for the learning space used in the second empirical study were designed and authored in co-operation with two researchers from the area of teacher education, specializing in elementary and mathematics/science education. Two distinctive tests were organized in separate schools, but only the second test run gave usable results, since the learning space design in the first trial contained significant flaws.

The two studies, even put together, do not reflect the true potential of the model. Therefore, to exploit the potential of the model, collaborative efforts with teams including expertise on several disciplines would be needed. The possibility to use centralized learning material servers and a ready-to-use description language with learning seed editors can help in this aspect. One clear benefit of the learning space model, as the results show, lies within the useful evaluational help the teacher or the experts assessing the learning processes can derive from the use of the system.

Learning space model for the material authors. The third chapter described ease-of-use as a desirable quality in a learning system. The learning space model was designed to be used by non-expert authors, such as teachers interested in authoring material for the system but not equipped with e.g. programming skills. Therefore, a simple evaluation of the learning space model from the authors' point-of-view was conducted. The evaluation concentrated on the model itself, not the description language used in the prototype system or the editors for designing individual seeds. The hypothesis for the third study is that potential authors understand the model without problems, and can author seeds into a common learning space (i.e. the position of the seeds and the effect of actions do not differ substantially).

6.2 Test space for number word sequence skills

The organisation of the material to the learning space and the tracking of the actions of the users in the material provide a way to use the system in a novel manner: the progress of every learner can be derived from the use of the system for evaluational purposes.

In this following sections, we briefly describe the collection of test material for evaluating two of the components concerning basic skills for learning arithmetics in the context of special education. The evaluation was carried out empirically in a special school with four learners with cerebral palsy. The test results derived from the system are then reflected againts the standard neuropsychological tests for the learners.

6.2.1 Study setting

Description of the test material. The functionality of the learning space model was tested empirically by constructing a learning space for evaluating two of the key components for number word sequence skills for children with cerebral palsy. Number word sequence skills are prerquisites skills for elementary arithmetic. For simplicity, we call the constructed learning space the *test space* from here on.

Although the authoring of a learning space is technically relatively easy using the description language designed for the purpose, the real challenge lies in the conceptual design of the space. The challenge is amplified by the freedom for the learning material author. As stated earlier, there are no restrictions for what kind of dimensions to use, what the positions of the seeds in the learning space are, what the actions within the seeds are and what the effect of every action is.

The test space was deliberately designed to be small and simple to make the evaluation of the use of the learning space model more straightforward. Only two dimensions were chosen for the test space and the size of the space was only 25 seeds (Fig. 6.1). The dimensions had a certain "metric" and were enumerative, i.e., there was a clear order for the seeds in both dimensions.

Although number word sequence skills are said to include at least four sub-skills (Kinnunen et al. 1994), only two of them were chosen to be included in the test space. The sub-skills chosen to be dimensions in the test space were "The capability of the working memory" and "Step-size for enumeration". In the scope of this study, working memory means the group of systems associated to the short-term memory functions (Baddeley 1997). The capability is essential in many cognitive functions, such as reasoning, learning and comprehending. The step-size for enumeration means the capability of enumerating strings of numbers forward in different step-sizes, for example 1,2,3,4,... or 1,3,5,7,... where the step-sizes are 1 and 2, respectively. Using these dimensions for the test space has two benefits: they are easily ordered along the dimensions, and they are independent of each other. Independence means that they can be arranged so that the space is complete, i.e. there is a seed in every point of the space.

To the testees, the task to be completed in every test question had the same pattern. The testee had to memorize the instruction and pick the right number of visual objects from the presented ones. The instructions were written at the top of the screen, and each instruction was only shown for a fixed time.

The first and the easiest test question is shown in Fig. 6.2. The text at the top of the screen says "Pick three balls". The text on the button below the objects says "Ready", and should be pressed after the testee thinks he or she has the right answer. The value for the dimension "step-size" is 1, since there are only single elements (group size is 1) on the screen. The

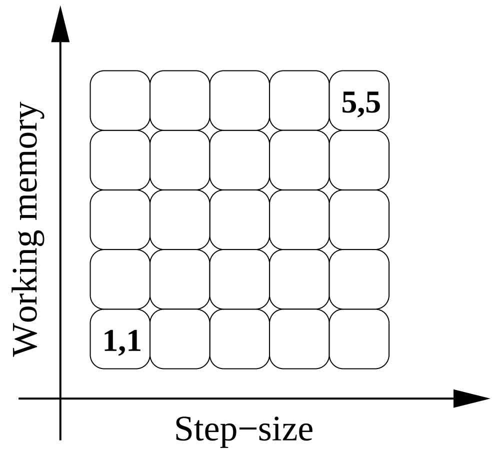

Figure 6.1: The test space. Starting point is at (1,1) and the most demanding test question is at (5,5).

value for the dimension "working memory" is also 1, since there is only one thing to remember (the amount to pick, "3").

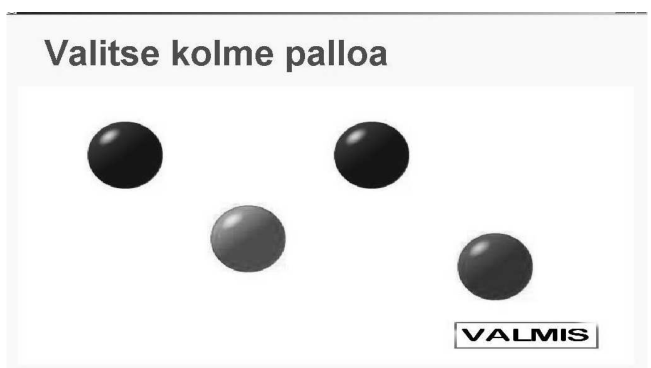

Figure 6.2: What the testee sees on the screen. The easiest test question: "Pick three balls".

Figure 6.3 shows the most demanding test question (i.e. the one in the upper right-hand corner of the test space in Fig. 6.1.). The task is to pick four balls and six blue squares. There are balls and squares of various colours, and grouped visually in various groups. The value for the dimension "step-size" is 5, since the grouping has group-sizes up to five. The value for the dimension "working memory" is also 5, since there are five things to remember (the two amounts, “4” and “6”, the two shapes, “balls” and “squares”, and one colour, “blue”)[1].

Figure 6.3: What the testee sees on the screen. The most difficult test question: “Pick four balls and six blue squares”.

The effect of every answer in the test questions followed a particular pattern. If there was an error regarding the *number* of objects picked by the testee, the error was considered to be caused by an error in step-size. If the error was with picking up wrong colors or shapes, the error was considered to be caused by an error regarding the working memory dimension. Figure 6.4 shows the effect of every action in all possible answers.

Because the author of the material has the freedom to choose the effect for every action, two types of effects were tried out. The “optimistic” test space never lowers the values for the dimensions (position of the testee in the test space never moves down or left in Fig. 6.1), even in a case where there is an error regarding both of the dimensions. In a case where there is an error regarding only one of the dimensions, the value for the other dimension is raised (the upper effects presented in Fig. 6.4). The “pessimistic” test space raises the values only when the answer is correct regarding both the dimensions, and lowers the value in case of an error for one or two dimensions, as presented in Fig. 6.4 by using bold typeface.

Descriptions of the testees. All the testees are motorically disabled and use wheelchairs. They have cerebral palsy and they are all male. The

[1] It should be noted that even though the value for the dimension is five, it does not mean that the test question equals the five memory chunks commonly used in neuropsychology. The test question, however, demands more working memory capability than the most simple one, therefore the ordering of the test questions along the dimension is possible.

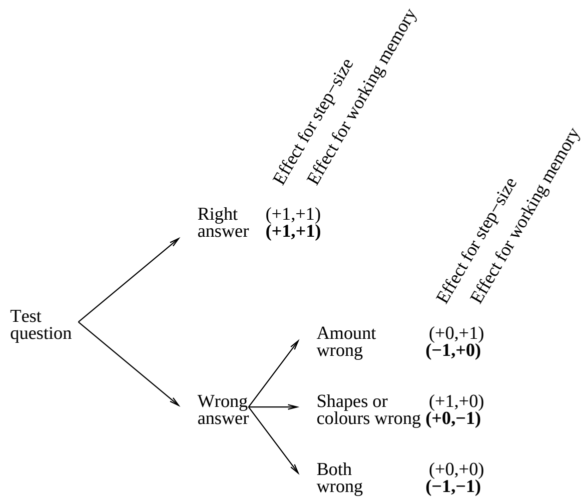

Figure 6.4: All the possible answers and their corresponding effects to the dimensions. The effects presented with normal typeface are for "optimistic" test space and the effects presented with **bold** are for "pessimistic" test space.

description of every testee (below) considers the aspects that are relevant to the use of the test space. The descriptions are based on personal neuropsychological evaluations conducted by a professional neuropsychological tester, but they are abridged and simplified for the purpose. The neuropsychological tests have been carried out during the testees' school years, and the most recent test results dates back to January 2002.

Testee 1 has significant difficulties in gazing; the gaze "drops" easily and starts again arbitrarily. This leads to difficulties in reading, when the testee has difficulties in staying on the line and return to the most recent point after the gaze drop. The testee also has deficits in mental programming. However, the testee does not have difficulties in perception. The testee also has learning difficulties and difficulties concerning memory. The testee's working memory is volatile to outside impulses, but the size of the auditive working memory is up to six units forward.

Testee 2 has difficulties in reasoning when given oral or written instructions and vocabulary is limited. Visual reasoning is difficult and failing to complete a task is especially hard. The testee can concentrate for a long period of time. Auditive working memory is a size of five units forward. Visuo-spatial perception is difficult, especially finding the essential in visual exercises. The testee understands oral assignments and is capable of remembering given instructions.

Testee 3 has learning difficulties but learns best via the auditive channel. Using gaze is also difficult. Thinking is impulsive. As in learning, associations help also when memorizing. The testee has difficulties in arithmetic exercises and in logical reasoning, e.g. has not been able to learn multiplication tables or more complex addition (e.g. when two additives form round numbers as in 23+77). Instructions and things learned are easily forgotten. The testee is motivated and tries hard to succeed.

Testee 4 has difficulties in attentiveness. The use of language is good but processing language is difficult (i.e. to take instructions, memorize, and recall). The testee gets tired easily. Visual ability suffers from persevering: the gaze can be "stuck" to a point. The concepts of time and numbers have been difficult. The most notable difficulties concern perception and mathematical abilites. Numbers "rotate" easily and comparing the size of numbers is also difficult.

Test setting. In every session, there were two testers present and the testee. One tester stayed with the testee by the computer, and the other tester stayed silent in the background taking notes. If the testee had trouble reading the instructions, the tester by the computer read the instructions out loud once. If the testee had difficulties using the mouse, different input devices were used, including a track ball and a single-switch input device. If it proved to be too time-consuming to find a usable input device, the tester used the mouse to pick the objects according to the instructions given by the testee. The testees were free to quit at any point.

The tests were carried out twice for each testee to find out the effect of different input devices and the different test spaces ("optimistic" and "pessimistic" test spaces). Other than that, the test setting was exactly the same. There was roughly a month between the first and the second round of tests. On both round of tests, the learners started from the origin. Since the test spaces were not designed to teach but to test cognitive abilities, the fact that the testees were already exposed to the test space a month earlier should not affect on how far the testees progress along the dimensions, especially when the tested qualities are abstract. Observations during the test supported the view that the previous test round did not help them in achieving better results but helped the testees in orienting themselves to the test situation.

6.2.2 Test results

The results from using the test space are shown visually in Fig. 6.5. The upper row ('A') presents the first round of tests and the lower row ('B') presents the second round of tests. A black rounded rectangle represents a test question with a correct answer from the testee to both of the dimensions or for just one dimension. A patterned rounded rectangle represents an answer which was wrong for both of the dimensions. Empty rounded rectangles are test questions that were not visited by the testee.

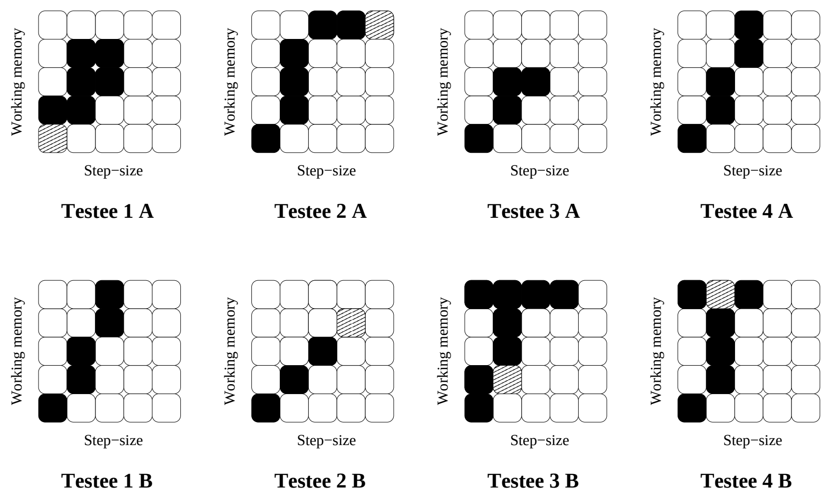

Figure 6.5: Figure of trails of the testees.

The test space was organized so that if a testee answers all the presented test questions correctly, the testee traverses along the diagonal from the point (1,1) to the point (5,5) and answers only to five questions altogether. If the testee cannot answer correctly, he will diverge from the diagonal. The interesting issue to watch is to which part, upper (stronger in working memory) or lower (stronger in step-size), the testee will diverge.

After conducting both the test rounds and observing the test situations and the results, the most valid tests are considered to be 1B, 2A, 3B and 4A. Tests 1A and 3A used the pessimistic test space, which seemed to be too punitative. The testee in 2B was tired and demotivated whe he realized that the test appeared to be like it was the first time. Tests 4A and 4B were both observed to be successful, and the similar results support the observation.

Testee 1: The first round of tests was conducted using the pessimistic test space. The instructions were read once to the testee and the choices were made according to the instructions given by the testee. At the end of the test, the testee complained about the monotony of the questions. On the second test round, optimistic test space was used and the testee used a single-switch input device. This time the testee was able to progress better, since the test space was less punitive. For both the tests, the testee was showing more progress to the working memory dimension.

The testee has relatively strong working memory, which was shown also in the test result. Diagnosed learning difficulties and deficits in mental programming imply the step-size dimension should be less strong, which was the case in the tests. The risk of outside impulses in the test space was eliminated, and no movement or other attention-grabbing effect was used. The difficulties in gaze are hard for a computer to interpret unless some extra hardware is harnessed to the system. Gazing difficulties could well affect the test results, and the effect is not easily separated from the results.

Testee 2: At the beginning of the first test round, the testee stated that "it is very hard to read from the screen" but had no trouble seeing the objects or reading the instructions when they were short enough. For more complex test questions, the instruction was read to the testee. The testee was highly motivated to succeed and was tired after the test. On the second test round, the testee was not motivated since the test appeared to be the same. The testee was not happy with any of the input devices, so the tester used the mouse according to the instructions given by the testee. When comparing the two test results (both with optimistic test spaces), it was clearly shown that the second time the testee was not as motivated anymore. The testee still went on answering the questions but was willing to quit sooner when the questions became more difficult. The result from the first test showed very good capabilities for the test. Testee 2 is the only one not biased towards the working memory dimension. Testee 2 was the only one to reach the final question of the test.

Working memory is relatively strong compared to mathematical ability. Therefore, it is no surprise that the testee went up the working memory dimension before proceeding to the step-size dimension in the test space. The testee's difficulties in visuo-spatial reasoning did not pose too many problems, since there was no need to find the essential in the test questions; everything was essential. The possibility to measure time and include elapsed time in the test result could help in evaluating the persistance of the testee in this particular type of tests, but it is not included here[2].

[2] If e.g. a teacher is conducting this kind of tests in the classroom, he or she can observe elapsed time with other means and subtract possible extracurricular delays.

Challenging tasks, such as tests where the subject has to memorize something, should be motivating in some sense. If the subject is not motivated, the results might get significantly worse. This was clearly seen in testee 2's efforts, when the second round was much shorter than the first attempt.

Testee 3: The first test was conducted using the pessimistic test space. The progress was not as smooth as during the second test round when optimistic test space was used. The early mistake in point $(2, 2)$ during the second test was an involuntary mistake caused by the deficiencies with the accuracy of single-switch input and scanning of choices. As is the case with testee 1, the results for testee 3 tend to be more emphasized on the working memory dimension. The instructions were read out loud by the tester.

Learning difficulties and difficulties in mathematical ability suggest that the working memory dimension should be stronger with the testee 3. High motivation to succeed is visible in the test results. The result from standard neuropsychological tests, "association helps memorizing", is not tested in this space. However, it could be included in the test space easily by presenting associations if the first answer was not correct. The same applies to channels and modalities; whether the testee learns best via auditive channel was not tested but would easily be tested.

Testee 4: The optimistic test space was used for both of the tests, and the trails are much the same. The stronger dimension is clearly the dimension for the working memory. The first test round for testee 4 is exactly the same as the second test round for testee 1. The instructions for both test rounds were read out loud, and for the second test round, the testee used a single-switch input device.

The test result for testee 4 was fairly good, so there was no support for the difficulties in processing the language and take instructions. Also, observations in the testing situation did not show these difficulties. Numbers and mathematical ability are diagnosed to be problematic for testee 4, and the step-size dimension was weaker in the result. Number rotation and problems with size comparison were not tested, but they could be tested with appropriate material. Unfortunately, working memory capability was not tested with traditional methods, so comparing the test space results in this aspect is impossible.

6.2.3 About the results

It is clearly not an easy task to design a learning space for testing cognitive abilities, since it is inherently difficult task as a domain *per se*. That is why it is advisable to take the test results with a grain of salt.

Clearly, dimension for step-size can be interpreted as a dimension for mathematical ability, as step-size is too fine-grained an ability in neuropsychological testing. Given that, the results from the test space are generally in line with the results of the standard neuropsychological tests. This, in turn, suggests that the test space was not completely erroneous in testing the qualities described above, and, moreover, the learning space model can be used in preparing rudimentary tests for at least some cognitive qualities. Considering a different point-of-view, the question whether standard neuropsychological testing can benefit from the use of learning space model is a more difficult one. First of all, there were no radical differencies in the test results. This was not surprising; even though the testees were individuals, they all possessed common qualities in the areas tested. The test showed that the testees are all capable of performing task involving stress on working memory; they possessed the concept of small mathematical quantities and were capable of picking the right number of objects from the given set, if the number to be remembered does not grow too high. Observations in the testing situation suggested that the mathematical dimension felt harder for the testees than the memory dimension. One possible reason might be a question priority: the testees started working on the exercises by trying to memorize the task, and only after that they started searching for the correct objects. This bias towards working memory dimension was also seen in the trails. More importantly, the observation showed that the dimensions are not completely independent from each other. Memorizing is difficult when you have to add groups of various sizes, and adding is difficult when you have to memorize the details given for the task.

Yet another issue in the test space is that it was assumed that the dimensions are (approximately) homogenous. The dimensions used in the test space are likely to be non-homogenous. In reality the learning spaces are rarely homogeneous in terms of similar metrics for every dimension. For example, the test space had apparently an enumerative working memory dimension, but is there a direct mapping from the memory units to memory units on the screen, since different variables (shape, colour, location on the screen) were used for the same purpose in the working memory dimension? And, is the metric for the step-size dimension comparable to the working memory dimension (a step in step-size is the same level of achievement as a step in working memory)? Moreover, what was the effect of Gestalt principles for grouping in the test: what is the effect of proximity of objects on the screen, what is the effect of using colors or shapes? How this effect is to be measured, and how will the difficulties in visuo-spatial reasoning affect to these principles? In thisstudy, no effort was made to evaluate these effects. However, when the learning space is used for giving additional information rapidly and not a basis for some critical evaluation, the use of a test space can bring benefits by giving comparable guidelines for a teacher.

6.3 Learning space for the addition algorithm

6.3.1 Study setting

Description of the test material. In addition to the test space presented in the previous section, the functionality of the learning space model was also tested empirically by constructing a learning space for arithmetic addition. The learning space for the experiment is called Matinaut. The Matinaut material consisted of drilling material presenting addition exercises (Fig. 6.6). An exercise screen presents an exercise and two columns for multiple-choice answers. The learner is requested to select the answer from the first column if he or she has used the addition algorithm to solve the exercise. The second column is to be used if the answer was achieved by mental computation (as seen on the left in Fig. 6.6).

In case of two erroneous answers to an exercise, there were general teaching material called "videos" shown to learners. A video presented appropriate steps to solve a similar type of exercise using the addition algorithm (Fig. 6.7) without using the same numbers than in exercises. The videos used animations and speech to explain the steps in detail.

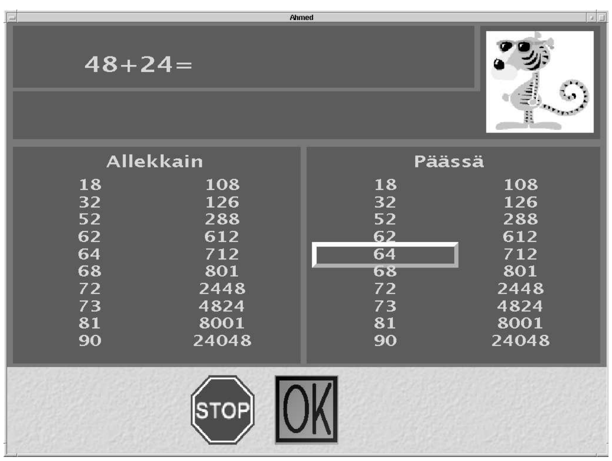

Figure 6.6: The learner's view to the learning material: an exercise with multiple-choice answers. Column titles are Addition algorithm ("Allekkain") and Mental computation ("Päässä").

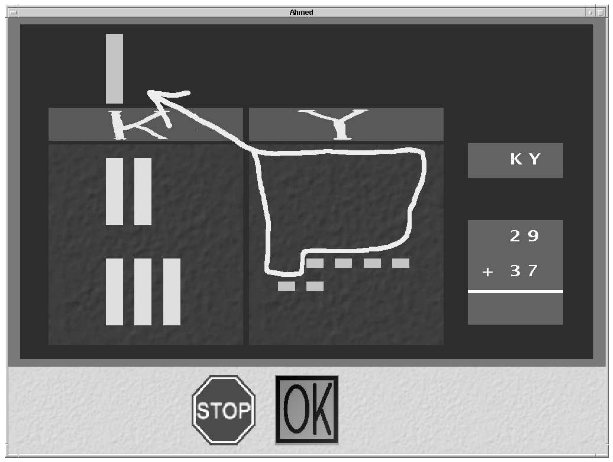

Figure 6.7: A still picture of a "video" (general solving procedure to an exercise type) is presented.

Authoring a learning space is technically easy using the description language designed for the purpose, but there are conceptual difficulties caused by the freedom for the learning material author. As stated earlier, there are at least four kinds of questions to be answered: What kind of dimensions to use, what the positions of the seeds in the learning space are, what the actions within the seeds are, and what the effect of every action is.

The first issue to consider is to break the learning topic into meaningful dimensions. For addition exercises, several possibilities exist, but the Matinaut learning space was chosen to include three dimensions, namely "Number field", "Mental computation" and "Addition algorithm". "Number field" was divided into five discrete steps: numbers between 0 and 10, 10 and 20, 20 and 100, 100 and 1000, and 1000+. The corresponding dimension values were 0, 30, 70, 90, and 100. "Mental computation" was also divided into different categories, namely "Addition with no composing, bigger number first", "Addition with no composing, smaller number first", "Adding to 10/100/1000[3]", "Adding a ten or tens", "Addition with composing for ones", "Addition with composing for tens", "Addition with composing for hundreds or thousands", and "More than one addition with composing". The corresponding dimension values for "Mental computation" were 0, 10, 20, 30, 70, 80, 90 and 100. "Addition algorithm" was divided into seven categories: "No reason for addition algorithm", "no

[3]The numbers to add give an answer of 10, 100 or 1000, such as 7+3, 40+60 etc.

carry-over", "one carry-over", "more than one carry-over", "carry-over to the empty unit (such as 911+200)", "carry-over bigger than 1", and "more than one carry-over, zero to tens or hundreds slot". The corresponding dimension values were -1, 10, 40, 50, 70, 90 and 100.

Examples of the position $(0, 0, 0)$ are 2+1 and 4+2. Examples from a position of $(70, 70, 40)$ are 29+32 and 48+24, because the Number field is from 20 to 100, there is a need for addition with composing for ones, and when using the addition algorithm, there is a need for one carry-over. Every possible position of the learning space had several exercises, and there were a total of 347 different exercises authored into the space.

It should be noted that the Matinaut space was not *complete*, i.e. the space had several "holes" since the dimensions chosen are not independent of each other. In other words, many locations in the space do not have seeds, since an exercise cannot fulfil the requirements for every dimension. For example, there cannot be a seed in a point $(0, 0, 40)$ since an exercise cannot have a number field between 0 and 10 and have carry-over. The space not being complete does not affect the functionality of Ahmed, but it means that the movement from one point to another can be a "jump" even though the values for the learner's position would indicate only a small step. Figure 6.8 shows the actual positions of the seeds for the Matinaut learning space. Dark rectangles mark the positions where there are seeds and white rectangles mark the "holes" in the space. The three dimensions are shown pairwise.

The learning material author is also responsible for defining the actions a learner can make in a given seed as well as the effect of the action. Considering the Matinaut space, in the case of a correct answer by mental computation, the effect was $(+4, +2, +0)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of a correct answer by addition algorithm, the effect was $(+4, +0, +2)$. The learner progresses more rapidly on the Number field dimension to ensure that the learner does not have to stay with too easy problems too long.

All the erroneous answers for the multiple-choices were generated according to the known error types for both mental computation and addition algorithm. The errors for every exercise are straightforward to produce automatically. If the number of generated errors based on the known error types was less than 20 (the number of multiple choices was fixed to 20, see Fig. 6.6), the rest of the errors were produced by a random generator.

In the case of a wrong answer by mental computation with a choice that had an error-type generated error, the effect was $(+0, -1, +0)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of

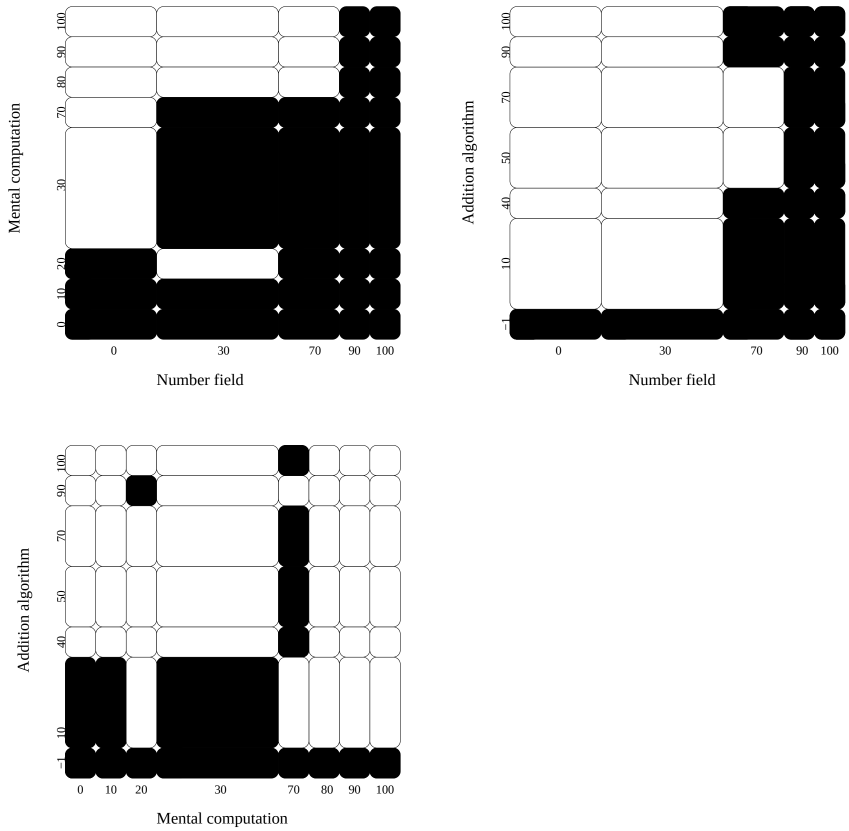

Figure 6.8: The three-dimensional learning space for the Matinaut test is shown with two dimensions at a time.

a wrong answer by mental computation with a choice that had a randomly generated error, the effect was $(-1, -1, +0)$.

In the case of a wrong answer by addition algorith with a choice that had an error-type generated error, the effect was $(+0, +0, -1)$ to dimensions Number field, Mental computation, and Addition algorithm. In the case of a wrong answer by addition algorithm with a choice that had a randomly generated error, the effect was $(-1, +0, -1)$. The effect on the values for the dimensions for every answer is illustrated in Figure 6.9.

Apart from the guidelines presented above, some erroneous answers had slightly different effects based on the authors expert opinion. This approach is in line with the expected use of the learning space model; more variation in the effects means more possibilities for individual learning paths.

A single point in the Matinut learning space contained several different

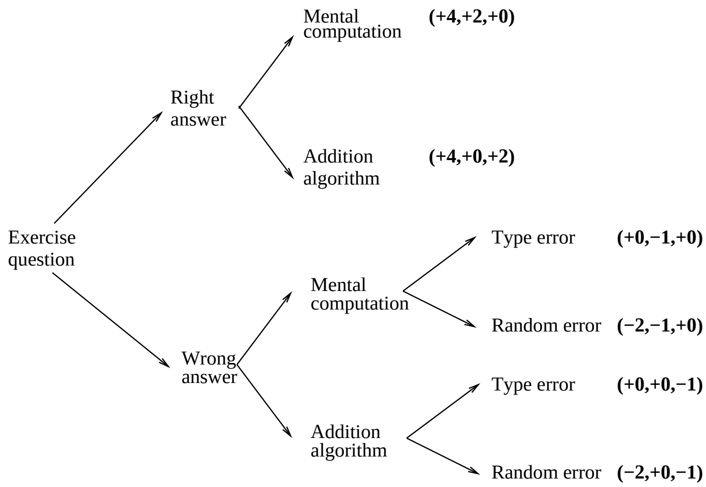

Figure 6.9: The effects of possible actions for the values for every dimension.

exercises of the same type. In addition, every one of the seeds actually contained a chain of seeds. The rationale behind this was that it should be possible to try the same exercise after an error. After a second error, a video for general solving practice was to be presented. After the video, the same exercise can be tried once more. The effects on the values for the dimensions are the same as above for every time an exercise in the exercise chain is answered, except after the video the last trial of an exercise will not lower the values.

There are admittedly many possible ways to construct a learning space for addition. The approach taken in this experiment is partly based on the existing knowledge about the error-types and difficulty order of tasks included in mental computation and addition algorithm (Grinstein & Lipsey 2001), and partly based on hands-on experiences of teaching elementary arithmetics.

The testees. Two classes of learners (N=41) at the age of 7 and 8 in an elementary school were chosen to be testees and were exposed to the system. Everyone in the class attended the tests. The testees were free to use the system during their spare time and during mathematics classes. There were only three computers in each class so there was competetion in who could have access to the system. The log files from the system were gathered after two weeks, during which time the learners started to learn the addition algorithm as a part of their curriculum.

The hypothesis. The expected result was that the learners should progress rapidly to their skill level and after that the progress is slow unless the testees have some outside help (the teacher, the videos) to learn new things. In other words, the learners were assumed to achieve their zone of proximal development (ZPD) (Vygotsky 1978) and after that their progress is slowed but not stopped because they have the teacher teaching the addition algorithm and the videos showing different methods of solving the exercises. In other words, the learners are in their zone of instructional interaction as defined by Murray & Arroyo (2002), where the learning material presented to the learners is neither too difficult nor too easy.

6.3.2 Test results

The evaluation was carried out without a control group and the testees and their test results were not compared to the average. The focus of the evaluation was to see the individual trails left by the testees and find out if the trails alone can give any valuable information. In a way, the question is to evaluate the learning space schema by evaluating an instance of a learning space in a real-world setting.

The data was gathered after two weeks of using the system. Some of the testees were still observed to be enthusiastic after two weeks, e.g. competing about who has the access to the system during the spare time on a lunch break.

Figures 6.10 to 6.15 show a collection of trails of various learners. The trails are individual trails chosen to represent different categories of the progress expressed by the testees. It should be noted that the individual scores should not be compared against each other since the learners spent different amounts of time working with the system.

The trails in Figures 6.10 to 6.15 do not visualize the trails from a seed to another but the points gathered for each dimension. The points gathered are more informative for this purpose compared to the example of visualizing the trails between the actual seeds presented in the previous section. In the Figures from 6.10 to 6.15, values for the x-axis indicate the exercises tried, and the values for the y-axis indicate the points gathered. In addition, the solid lines indicate progress in Number field, the dashed lines indicate progress in Mental computation, and the dotted lines indicate progress in Addition algorithm.

The testee presented in the first diagram in Fig. 6.10 has reached her level on the mental computation dimension just before the 30th exercise. After not progressing for a while, the testee has moved from using mental computation to addition algorithm and ended up in her zone of proximal

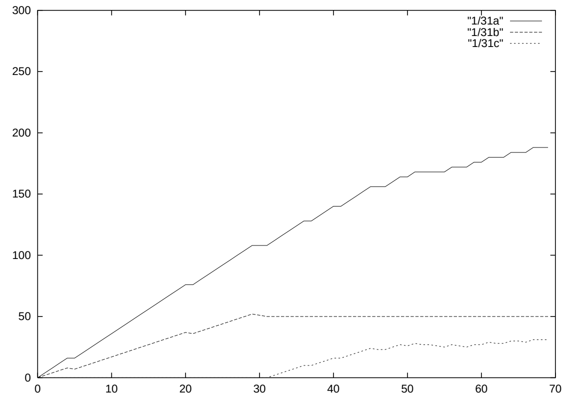

Figure 6.10: The progress along three dimensions for Testee 1.

development.

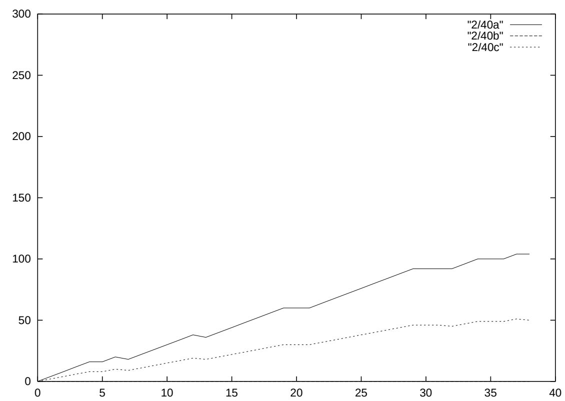

Figure 6.11: The progress along three dimensions for Testee 2.

The testee presented in the second diagram in Fig. 6.11 has not used mental computation at all. The progress has been slow but nearly constant. She has not reached her ZPD, but she has tried only less than forty exercises.

The testee presented in the third diagram in Fig. 6.12 has apparently reached his ZPD even though he has used both the mental computation and the addition algorithm.

The testee presented in the fourth diagram in Fig. 6.13 has reached her ZPD with mental computation after 50 exercises, and switched to using the addition algorithm at the very end. After the switch, her progress boosted.

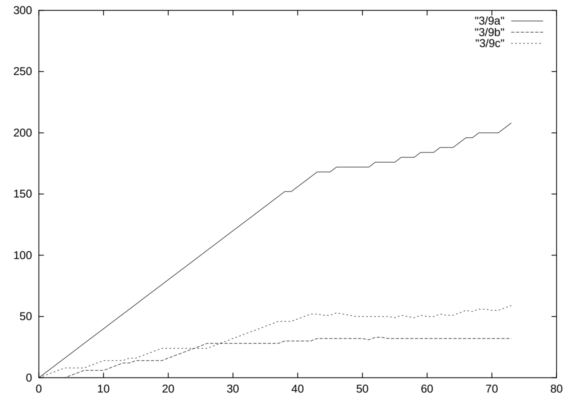

Figure 6.12: The progress along three dimensions for Testee 3.

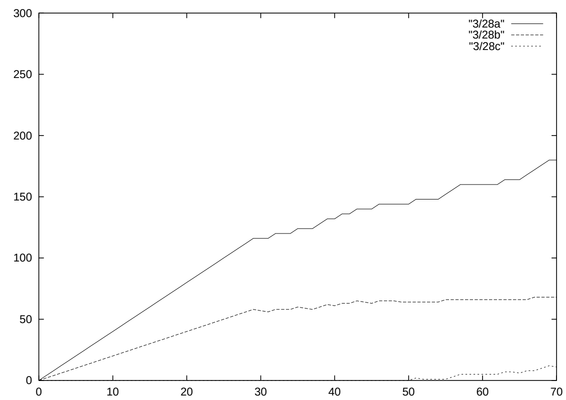

Figure 6.13: The progress along three dimensions for Testee 4.

The testee presented in the fifth diagram in Fig. 6.14 has reached her ZPD with mental computation but has not started to use the addition algorithm even though she has not progressed for the last 25 exercises.

The testee presented in the last diagram in Fig. 6.15 has used only addition algorithm and has progressed rapidly (virtually error-free and over 100 exercises completed).

The effect of videos. The seeds were organized in the Matinaut space so that after two wrong answers to an exercise, a video presenting the general solving method for that particular exercise was shown to the testee. An interesting issue to study is whether presenting the videos have any effect on the correctness of the answers. As anticipated, the effect of videos was

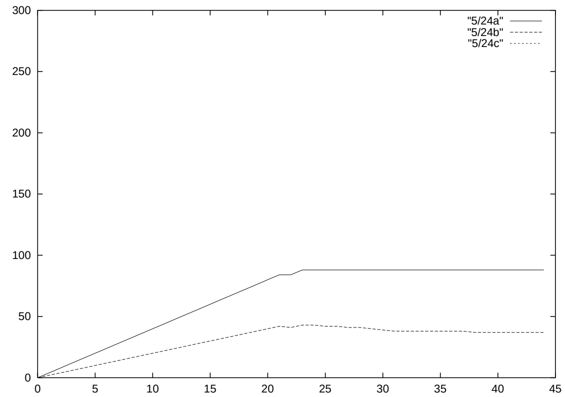

Figure 6.14: The progress along three dimensions for Testee 5.

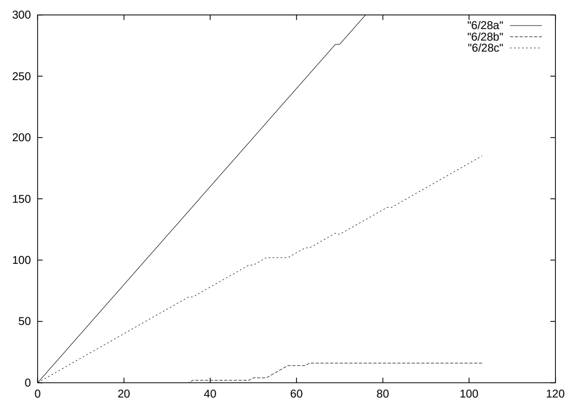

Figure 6.15: The progress along three dimensions for Testee 6.

not remarkable. The video was shown 357 times, and after watching the video, the correct answer was given 95 times (27%). Although the videos were informative and included animations and speech for the solving of the exercise, they did not demand any interactivity and there was no direct reward for watching the video (since the video did not show an answer to that particular exercise). Also the observations in the classroom suggested only a small effect for videos, since in some cases when a video appeared after two wrong answers, the testee was not paying attention to the video.

However, interviews with the testees in the classroom indicated that some learners can indeed benefit from the videos if they possess metacognitive skills to understand the connection between the general solving startegy for the exercise and the actual exercise. When studying the effect of the videos individually, several testees showed much clearer effect than the average: 57% correct answers after a shown video (4/7), 50% (4/8), 40% (8/20), and 38% (5/13). In contrast, there were also several zero effects: 0/10, 0/10, 0/4, and 0/3 among others.

6.3.3 About the results

The results supported the hypothesis: the learners reached their ZPD, and progressed slowly after that. The trails of each individual learner give rise to another added value of the system and the learning space schema, that the teacher (the tutor or the evaluator) can instantly see by a glimpse at the visualizations which routes the learners have traversed and what kind of progress they have presented. It would be possible to make the information visible also for the learners for self-evaluation but in this version of the system it has not been implemented.

Although the material for the Matinaut learning space had to be authored beforehand, various generators and semi-automatic editors were used to speed up the authoring. The learning space schema enables adding seeds to an existing space directly. The teacher can add seeds without making any connections to seeds authored earlier: setting the position of a seed for each dimension is sufficient.

6.4 Evaluating the learning space model with potential material authors

One of the aspects in adaptive learning systems is how easy it is for a learning material author to produce learning material for the system. As is often the case, the material itself is relatively easy to author with appropriate editors. The problematic issue is the knowledge representation; how to include a semantic model or some other necessary structure for the material so that it can be used adaptively.

This section presents a study where a group of potential authors for the learning space model were asked to position various seeds into the space, and to define effects for various actions a learner can take in the seed. In other words, the study concentrated on the in-space authoring process rather than the in-seed authoring process.

6.4.1 Study setting

As mentioned earlier, there are four major responsibilities for the learning material author: what dimensions to use, how to position the seeds along those dimensions, actions in the seeds and the effect of every possible action. Only two of these responsibilities were chosen to be evaluated in this study, namely *how to position the seeds into the space* and the *effect of every action*. Defining dimensions for a learning space was not chosen to be evaluated, since it can be considered as a higher order skill; dimensions should be chosen by those who are truly comfortable with the model and with the domain in question. On the other hand, every teacher interested in authoring material for the model should be able to prepare material after the dimensions are fixed. This would enable collaborative authoring of learning seeds, thus easing the burden of preparing thousands of seeds for individual authors.

The other quality of seeds tested in this study was the effect of the actions. The actions themselves are not interesting in this context, since arbitrary actions are hard to compare. What is more interesting is the logic behind the possible actions. In this study, the actions were chosen according to certain logic concerning error types.

Ten students (called *test subjects* from here on) studying to become teachers in an elementary school with emphasis in mathematics teaching took part in the test. Nine out of ten test subjects were female. The test subjects were explained that they should give a value for each example exercise in respect of four learning objectives. It should be noted that the subjects were not asked to position the seeds into the four-dimensional space. Instead, they were asked to think each dimension individually as a learning objective and give an exercise a value in relation to the particular learning objective.

The four learning objectives (dimensions) given were **addition**, **multiplication**, **calculation order** and **mathematical problem solving**. These dimensions were used both for the study of position of the seeds and for study of the effect of the actions.

Position of the seeds. A collection of eight seeds as arithmetic exercises was to be positioned in the space. These eight exercises were:

- $3 + 2$
- $2 * 3$
- $2 * (3 + 2)$
- $2 * (3 + (2 * 3))$
- $2 + (3 * (2 + 3))$

- $(2+3)*(2+2)$
- **Verbal 1**: Carl steals two apples from a tree. Lisa gives Carl three apples more. How many apples does Carl have?
- **Verbal 2**: Carl steals a lot of apples. First Carl steals two apples. After that, Carl goes to steal two times. On both times, he steals three apples more. How many apples has Carl stolen altogether?

The test subjects were advised that they can base their positioning on four aspects: 1) magnitude of the result of the exercise, 2) number of terms in the exercise, 3) number of parenthese of the exercise, and 4) nested parenthese. In a case where the value cannot be assigned to a particular dimension, the subjects were instructed to put in a dash (–), and if the subject did not have an opinion about the value, the value should be left empty.

Effect of the actions. The second part of the study included two seeds with pre-defined actions, i.e. exercises with multiple-choice answers. These two exercises were:

- $(2*(3+2))$ with possible answers of

  8
  10
  12
  something else

- $2*3$ with possible answers of

  5
  6
  something else.

The test subjects were asked to assign an effect of -1, 0, or $+1$ according to the effect of the answer in respect to the learning objective. In case of not being able to assign a value for the effect, the value was to be left empty.

The logic behind the erroneous multiple-choice answers was as follows. In the first exercise, the answer 8 indicates that the learner has not paid attention to the calculation order but multiplied first and added after. The answer 12 can suggest that the learner has made an error in interpreting addition as multiplication. In the second exercise, the answer 5 suggest that the learner has made an error in interpreting multiplication as addition. This logic was told explicitly to the test subjects. Including the time to instruct the task, the test subjects took 15 to 20 minutes to complete the task.

Expected results. For the positioning of the seeds, it is expected that the values for addition, multiplication and calculation order dimensions are easier to assign than the more vague mathematical problem-solving dimension. Moreover, the verbal exercises are thought to be harder to assign than numerical exercises. Although it is easy to transform (these) verbal exercises to numerical ones, it requires different skills from the learners than standard numerical exercises.

For the effects of the actions, it is similarly expected that it is easier to assign effects for addition, multiplication and calculation order than to mathematical problem solving. It is also expected that it is easier to assign unified effects for correct answers than for incorrect ones. Moreover, it is easier to assign unified effects for an incorrect answer if the error-type can be traced from the answer, and the error-type has an easy-to-see relation to the dimensions used in the learning space.

6.4.2 Test results

Position of the seeds. The results of the study for positioning the seeds are gathered in Table 6.1. Because the test subjects were asked to consider each dimension individually as a learning objective, we can examine the similarity of individual positions without a need to use vector similarity measures. The position averages are presented along with the standard deviation. Since there is no one right answer for the positions, we will concentrate on how much the positions alternate; the smaller the standard deviation is, the greater is the similarity between the mental images of the learning space within the test subject group.

The test subjects were allowed to use their own scale for the positions, although a scale from 1 to 10 was encouraged. Every student started from 1, but only few of them used a scale from 1 to 10. To calculate the average for every seed, the values from the test subjects were scaled to a range between 1 and 10, and rounded to a nearest whole number.

In several cases, the test subjects used a dash to indicate that the value cannot be assigned for some dimension. For example, most of the test subjects (8/10) indicated that the multiplication values for the exercise $3 + 2$ as well as for the first verbal exercise ("$2 + 3$") cannot be assigned. Five out of ten test subjects thought that these two cases were the only

	Addition	**Multiplication**	**Calculation order**	**Problem solving**
3+2	1.8 (0.79)	1.0 (n/a)	1.5 (0.76)	1.7 (0.87)
2x3	4.0 (1.26)	1,7 (0.82)	1,6 (0.73)	2,3 (1.41)
2x(3+2)	4,6 (1.84)	3,9 (1.37)	4,0 (1.25)	3,9 (1.76)
2x(3+(2x3))	8,2 (2.15)	8,0 (1.33)	7,9 (1.52)	5,4 (2.01)
2+(3x(2+3))	8,6 (2.46)	5,9 (2.42)	8,3 (1.89)	5,4 (2.01)
(2+3)x(2+2)	5,6 (2.63)	6,0 (1.83)	5,9 (2.18)	5,0 (1.50)
Verbal 1	4,4 (1.58)	1,0 (0.00)	2,2 (1.79)	4,1 (2.60)
Verbal 2	7,9 (1.73)	5,4 (1.90)	5,4 (2.32)	7,3 (2.75)

Table 6.1: Average positions of the seeds with standard deviation in parenthesis.

ones that cannot be assigned. One test subject interpreted the dimensions so that the numerical exercises do not have a position in mathematical problem solving dimension; the only two exercises that could be assigned for mathematical problem solving dimension were the two verbal exercises for this test subject.

In total, only one value for dimensions was left empty (one test subject left the value for the addition dimension in exercise $2 * 3$ empty) indicating that in most cases it is a possible to assign a value for the dimension. This suggests that the dimensions chosen for this purpose are suitable and independent. Moreover, the results suggest that model itself was easily understood by potential authors with no prior experience on explicit knowledge representation for learning materials.

As mentioned earlier, it was expected that the straightforward dimensions, namely addition and multiplication, would have the smallest deviation. This was not the case, however. The deviations within the dimensions were roughly similar.

Effect of the actions. Table 6.2 shows the averages of the effects for answers in the exercise $(2 * (3 + 2))$.

(2*(3+2))	**Addition**	**Multiplication**	**Calculation order**	**Problem solving**
8	0.4 (0.84)	0.6 (0.70)	-1.0 (0.00)	-0.4 (0.53)
10	1.0 (0.00)	1.0 (0.00)	1.0 (0.00)	0.9 (0.33)
12	-0.7 (0.48)	0.6 (0.88)	-0.6 (0.70)	-0.4 (0.53)
else	-0.6 (0.52)	-0.6 (0.52)	-0.5 (0.53)	-0.4 (0.53)

Table 6.2: Average effects of the actions for the seed with standard deviation in parenthesis.

As expected, the "easiest" action to define the effect is the correct answer; the effects for incorrect answers vary hugely. The random erroneous answer ("something else") appeared to be difficult to define for two of the test subjects, so they left it blank. One test subject left the effect for the mathematical problem solving dimension undefined. Two out of ten test subjects used only $+1$ and -1 effects, and no zero effects. No test subject used the dash to imply that the effect cannot be defined.

Table 6.3 shows the averages for the effects for answers for the exercise $2 * 3$.

2*3	**Addition**	**Multiplication**	**Calculation order**	**Problem solving**
5	-0.1 (0.88)	-0.8 (0.42)	0.1 (0.57)	-0.5 (0.53)
6	0.3 (0.48)	1.0 (0.00)	0.4 (0.52)	0.8 (0.46)
else	-0.3 (0.50)	-0.7 (0.50)	-0.1 (0.33)	-0.3 (0.49)

Table 6.3: Average effects of the actions for the seed with standard deviation in parenthesis.

Somewhat surprisingly the correct answer did not appear to be easily defined for the test subjects, and the values given deviated as much as with the incorrect answers (except the effect for multiplication dimension). The most notable observations from the test subjects' point-of-view seem to be 1) uncertainty if multiplication involves addition, 2) does simple multiplication involve calculation order skills, and 3) does simple multiplication involve mathematical problem solving. If the learning spaces need to be unified, these kind of questions must be answered explicitly beforehand when instructing the authors to prepare the material for a learning space.

6.4.3 About the results

The variation between the vectors was more than expected thus making authoring of a single learning space with distinct authors not reliable. On the positive side the learning space model itself is simple grasp, especially in a case where the space is considered one dimension at a time and dimensions are thought as learning objectives. There were significant deviation in the positions and in the effects assigned, but every test subject understood the model without difficulties.

The deviation among the test subjects in the study proved to be so significant that it is not advisable to have separate authors create seeds for the same learning space, or to use completely or partially same dimensions in different learning spaces. The deviation is remarkable even though the tested population was relatively small and homogeneous (i.e. age, gender and study history). This result is not surprising; similar results have been reported for same type of joint knowledge representation, e.g. in building knowledge spaces by querying separate experts (Dowling & Hockemeyer 2000).

Different authors for one learning space need harmonization of some sort. One way to harmonize the positions and effects used is to prepare qualitative explanations for different values so that the values are unambiguous. An example of a qualitative explanation could be as follows: "value is 1 for the addition dimension if the result is less or equal to five and there are only two numbers to add". Naturally, the same applies to defining the effects of actions.

Chapter 7

Learning spaces as vector spaces

7.1 Aspects of the concept of the learning space

Designing a platform for a learning environment benefits from a model-theoretic analysis of an appropriate space. In the case where the learning environment adaptively supports the learners to traverse through the learning space (and the seeds in it), the environment should be designed in a way that makes full use of the properties of the underlying mathematical structure. The learning space can be interpreted as a vector space or a more general metric space, with respective consequences on its semantic interpretation, potential functionality, and feasibility. The need for a robust model for an adaptive learning environment is apparent. Self (1995) lists several reasons for formalisations of any scientific endeavour and as such, adaptive educational systems. One of the reasons in the list states:

> "An analytical study of a component of AI-ED [artificial intelligence in education] systems can lead to precise statements about the power and shortcomings of that component and enable comparative studies of various proposed implementations of that component. Thus, formal tools may help us manage the complexity of AI-ED systems."

Following the guidelines presented by Self (1995), the learning space model can also be further discussed towards the formalisation of the concept. For the discussion, we classify learning spaces to be 1) homogeneous and 2) heterogeneous. In *homogenous* learning spaces, all dimensions are graded with the same domain. That is, one step in a dimension is computationally equal to a step in some other dimension. In *heterogeneous* learning spaces, two or more dimensions are graded with different domains.

7.1.1 Basic definitions

Here we recall briefly the main concepts regarding vector spaces. We will consider only vector spaces over the real numbers.

Let $\mathcal{V}$ be a non empty set. The elements of $\mathcal{V}$ are called *vectors*, and the real numbers are called *scalars* in this context. $\mathcal{V}$ is a *vector space* over the real numbers, if it satisfies the two following conditions:

1. There exists an operation called *vector addition*, denoted as $+$, and a vector $0 \in \mathcal{V}$, such that $(\mathcal{V}, +, 0)$ is an abelian group. That is, $\mathcal{V}$ is closed under $+$, and $+$ is associative, commutative, with 0 as the neutral element, and every vector in $\mathcal{V}$ has an inverse with respect to $+$.

2. There exists an operation called *multiplication by a scalar*, denoted as $\cdot$, such that for all reals α, β, and for all vectors V_1, V_2 in $\mathcal{V}$:

   1) $\alpha \cdot V_1 \in \mathcal{V}$;

   2) $V_1 \cdot \alpha = \alpha \cdot V_1$;

   3) $\alpha \cdot (V_1 + V_2) = \alpha \cdot V_1 + \alpha \cdot V_2$;

   4) $(\alpha + \beta) \cdot V_1 = \alpha \cdot V_1 + \beta \cdot V_1$;

   5) $\alpha \cdot (\beta \cdot V_1) = (\alpha \cdot \beta) \cdot V_1$;

   6) $0 \cdot V_1 = 0$;

   7) $1 \cdot V_1 = V_1$.

Let $V, V_1, \ldots, V_k \in \mathcal{V}$, and let $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$. The vector $V = \alpha_1 \cdot V_1 + \ldots + \alpha_k \cdot V_k$ is said to be obtained as a *linear combination* from the vectors $V_1, \ldots, V_k$. The vectors $V_1, \ldots, V_k$ are *linearly independent* if $V = 0$ (i.e., the vector V is the null vector), *only* when $\alpha_i = 0$, for every $1 \leq i \leq k$. Otherwise, the vectors $V_1, \ldots, V_k$ are *linearly dependent* (i.e., when the null vector can be obtained as a linear combination of them, with some scalar α_i being non zero).

The vector space $\mathcal{V}$ has *dimension* k if there exists vectors $V_1, \ldots, V_k \in \mathcal{V}$, such that for every vector $V \in \mathcal{V}$, there exists scalars $\alpha_1, \ldots, \alpha_k$, such that $V = \alpha_1 \cdot V_1 + \ldots + \alpha_k \cdot V_k$, and does not exist any set of vectors in $\mathcal{V}$ with less than k members with the same property. The set of vectors $\{V_1, \ldots, V_k\}$ is called a *basis* for $\mathcal{V}$, and every set of vectors from $\mathcal{V}$ with the same property is called a basis as well. The scalars $\alpha_1, \ldots, \alpha_k$ are called *coordinates* of the vector V with respect to the given basis.

Note that given a basis for a vector space, the representation of every vector in the space as a linear combination of the vectors in the given basis is unique. Hence,a vector of a space of dimension k can be represented as a k-tuple, formed with the scalars which multiply the vectors of the basis, respectively. Note, further, that every set of vectors $V_1, \ldots, V_k \in \mathcal{V}$ which are linearly independent, form a basis for the vector space $\mathcal{V}$, if $\mathcal{V}$ has dimension k. For every pair of positive integers $1 \leq j \leq k$, the k-tuple with 1 in the j-th component, and zero in all the other components, is called an *elemental vector*. The elemental vectors $E_1, \ldots, E_k$ form a basis for every vector space of dimension k. This basis is called *standard*. Then, when not considering any specific basis, the tuple of coordinates which represents a given vector is interpreted as a tuple of coordinates in the standard basis.

7.1.2 Formalization of a learning space as a vector space

Once we know that we can consider a learning space as a vector space, we must try to find out the corresponding semantics in our context, of the formal notions, properties and operations of that model. First of all, note that learning spaces fit in this formal model only when they are homogeneous. Secondly, as is always the case when an informal concept is formalized, we cannot make use of the interpretation too literally or too precisely (as opposed to an interpretation we do for every formal notion). We must consider the information which we can get out of a given learning space by using the model of vector spaces as a guideline to learn more about our framework.

The formal language should not be a "steel suit" which prevents the concepts from being displayed with their complete meaning. On the contrary, the formal language should provide us with a set of new notions and new perspectives with the help of which we can enrich the information we have about the framework and objects that we are studying. In addition, an analytical study of a component in an adaptive learning environment can lead to better understanding of the power and shortcomings of the component (Self 1995).

Next we proceed to give an initial and tentative interpretation of the main concepts in our formalism.

1. **A seed being expressed as a linear combination of the seeds of a given set:**

   Let $s = \alpha_1 \cdot s_1 + \ldots + \alpha_k \cdot s_k$.

   Suppose that the given set of seeds $\{s_1, \ldots, s_k\}$ is a proper subset of the set of seeds which form the learning space. We can interpret this fact as meaning that the specific skills which seed s is supposed to provide to the learner, can also be obtained from the seeds $\{s_1, \ldots, s_k\}$, by modifying the skills corresponding to the dimensions of the learning space for every seed in the set, according to their respective constants in the expression of s. The way in which the set of coordinates which correspond to every seed in the set, say s_i, must be modified, is given by the corresponding constant c_i. Depending on the value of c_i, it will mean to emphasize, or to restrict, the learning experience represented by s_i, in a proportion represented by c_i. Then, if every seed in the set is modified according to its corresponding constant, we can *delete* the seed s from the learning space, without changing its learning objectives.

Although this is formally true, it induces us to think the underlying pedagogical view behind the model. The view that redundant pieces of knowledge can and should be avoided is present in the knowledge space theory (Doignon & Falmagne 1985, Falmagne et al. 1990) used in efficient assessment of knowledge, but the view contradicts the pedagogical approach taken in the design of the learning space model. The learning process is not tried to be optimized in terms of time consumed; the learner should be guided to the unexplored areas of meaningful learning experiences.

2. **Linear independence of a given set of seeds:**

   Linear independence of a given set of seeds means that no seed in the given set can be expressed as a linear combination of the other seeds in the set. That is, there is no seed in the given set, whose skills the learner is expected to get once that seed was accomplished in its learning process, could be got also through the other seeds in the set, perhaps with some intensification in their coordinates in the different dimensions of the learning space (i.e., through a multiplication of the different seeds in the set by some factors from the set of real numbers).

   Thus, every seed in the given set is *necessary*, in the sense that the learning experience that it represents cannot be ignored by the learner in her learning process, no matter what her skills are in the other seeds, without strictly restricting the global knowledge which the learning space represents as a whole.

3. **Basis:**

   Given any set of seeds of a learning space S, we can form the span of these seeds, which in turn is again a learning space contained in S. If the span of the elements is S, and the seeds themselves are linearly independent, they form a basis.

The learning seeds in the set could be considered as dimensions for a new learning space, where the different grades mean a different degree of skill in the given learning seed. The new learning space would be equivalent to the original one, in the sense that every learning seed in the learning space should be given a new set of coordinates, in terms of the new dimensions (which are learning seeds in the original learning space). In this way, a learning space with the "same capabilities" as the original one could be obtained.

The learning space model is, however, a modified vector space in a way that every seed in the learning space can consist of an arbitrary amount of seeds within that seed (i.e. a collection of seeds), connected to each other with traditional hyperlinks. This simplified version of "sub"-spaces is also used in the prototype implementation of AHMED. In addition, the learner cannot be taken to a seed other than the starting seed of the collection to ensure that the learner does not miss earlier parts of a multi-part problem. This different approach to subspaces is chosen because it adds to simplicity when authoring the learning material since it is possible to present certain seeds in a fixed order, but also because it enables easy preparation of ready-made sub-problems to original problems when original problems are too challenging for the intended users. As stated earlier, supporting subtasking of problems is in harmony with supporting mental programming of learners.

The consideration of a learning space from the perspective of the theory of vector spaces makes clear the fact that dimensions and seeds are quite similar. Indeed, they both represent skills or knowledge in some learning objective. The difference is that the skills represented by the dimensions form a reference system of skills, in terms of which every skill represented by a seed can be expressed. The concept of basis induces us to think about transforming a given learning space by re-considering the reference system of skills. If a given set of seeds forms a basis, then it can be interpreted as the fact that they can be considered as dimensions, and all the seeds in the learning space may be expressed in terms of these dimensions.

7.1.3 Learning spaces as metric spaces

Every vector space is also a metric space, so that some appropriate and meaningful distance function should be found.

If the learning space is not complete (i.e. at least one seed in every point in space S), Ahmed uses Euclidean distance to take the learner to the nearest matching seed. However, the metric of Euclidean distance strongly assumes homogeneity. It is not clear if Euclidean distance is a suitable distance metric, since dimensions in a learning space are easily not homogeneous, at least in a general case. In fact, it is difficult to design a learning space where the dimensions are homogeneous. In practice, using Euclidean distance to pick up the nearest seed can still be sufficient, since learning spaces are always approximations; when breaking learning objectives down into different dimensions for a learning space, some approximation is needed. Even in cases where we consider the dimensions to be something else than learning objectives (e.g. motivations or points gathered), dimensions and the seeds' positions along the dimensions are approximations.

7.2 Partial order for the dimensions

The most obvious challenge in this formalisation is the assumption for learning spaces to be homogeneous. One can imagine that it is not easy to author learning materials for a learning space unless it is heterogeneous. Now we have assumed that the model for the learning space is homogeneous, but there is no guarantee that the learning material can be homogeneous in reality. For some domains, and perhaps for some learning material authors, truly homogeneous learning spaces could be possible. An example of such could be standardized test-type learning material. One could argue that even then the space is homogeneous only "evaluation-wise" and not "learning-wise".

However, if we want to examine methods to relax the rules of the learning space model to allow learning spaces to be non-homogenous, one possible approach is to demand that the dimensions in a learning space have *partial order*. That is, the relation pertaining a learning space dimension is reflective, antisymmetric, and transitive. In other words, when considering partial order as a tree-structure, there is an order from a root node to a child node, but no order between the sibling nodes.

It should be remembered that every time there is an addition or enhanced expressiveness in the model, it demands more from the learning material author. However, partial order is a strong candidate for a modification to the learning space model, since it is "natural" in the sense that the concept is easily visualized in two dimensions, and the tree-like structure is intuitive with strong hierarchy, directed paths and lack of cycles in the graph.

The use of partial order for the learning space dimensions is not straightforward. Although the concept of partial order is intuitive, using it in the learning space model is not. A learning space constructed using dimensions with only partial order is hard to imagine. The effect a cannot operate similarily to the learning space model presented in Chapter 4. The effect has to incorporate the same partial order as the dimensions, and joining the dimensions (mentally) together can be overwhelmingly complicated for the author. Moreover, the visualization of the learners' trails is difficult with partial order.

As a conclusion, we can say that considering the intended users and the motivation for the adaptation presented previously in this thesis, the need for learning spaces to be truly homogenous is not crucial. Simplifications and approximations are needed in preparing learning material, no matter what the representation for the knowledge is taken. Coarse adaptation mechanisms provided by the learning space model are able to function properly.

7.3 Discussion

The reason for this initial attempt to formalize the learning space model is to open a way to structure and study the model so that the ideas can be developed further. By using a formalism, it might help in finding new potential ways to exploit the learning space by making the learning experiences more meaningful, motivating, efficient and rewarding.

In addition to the discussion above, one possibility could be to consider the representation of a learning space as a relational database. That is, we can think of it as a finite set, together with a family of relations. As such the representation is straightforward. However, there are of course different ways in which a learning space can be represented as a database. The main point, though, is that in this way we could benefit from the deep knowledge about relational databases, which has already been developed, and which is still being developed. From a practical perspective, once we have a representation of a learning space as a database, the learning space administrator can query the database, by using any relational query language (e.g. SQL) to find out different properties of the learning space. This way we can provide the learning space administrator with an excellent tool to operate.

However, in order to design learning platforms which utilize the properties of the underlying model to its fullest, we have to start with simple cases, even if they sound trivial from the practical application's point of view. This process also forces us to understand the originality of a given platform. In the context of the learning space model, the question of efficient retrieval is not that important as providing a student with a meaningful learning environment. To achieve that goal, the model has to support operations which support and intensify the learning process.

Chapter 8

Conclusion

Learning space model. In a broad sense, adaptive learning systems are systems that enable individual paths through the knowledge space. The thesis presents a model for organizing the learning material into a vector-space and describes how the model was used in a real-life setting. The benefit of this model – in addition to supporting domain-independent learning materials – is indeed the adaptivity in learning events. Other issues include supporting evaluation of learners' learning processes.

The key issue in adaptive learning systems is to take the learner closer to the learning objectives, whatever the objectives are. Typically in adaptive learning systems, there are four different aspects in adaptation: knowledge, goals, background and experience, and preferences (Brusilovsky 1996). The learning space model incorporates these aspects into the concept of learning space. The seeds in the space are the knowledge, and by acting upon it, the learner gains knowledge. The dimensions are typically used as learning goals, although they can be considered differently. The learning space model adapts to the background knowledge or experience (i.e. knowledge gained before entering the system) by rapidly advancing the learner into his or her zone of proximal development. The model does not offer adaptation to user preferences *per se*; the reason is that unlike information retrieval (where adaptation to user preferences is used heavily), learning involves an element of surprise. The learner does not know what to learn in advance, therefore the learner cannot set the preferences for the learning process. The learning model, however, offers indirect support for adapting to user preferences by allowing the author of the material to build in support for e.g. various learning styles.

The author of the learning space has complete freedom in the authoring process, but the freedom does not come cheap. The author must design the space carefully, since the model itself does not offer any support. However, every author of the learning material is able to define the seed positions and the effects of the actions based on his or her own experience, or some theoretical model. The construction of the learning space is open enough to enable several theoretical approaches in the learning material.

Somewhat similar constructs are proposed in other systems for learning. For example, the underlying idea in the ASK systems is to capture important aspects of a conversation with an expert (Ferguson et al. 1992). In ASK systems, the learning is seen as a process of retrieving relevant cases (i.e. experiences) at the right time (Osgood 1994). The essential issue is how the cases are indexed.

Similarily, there is a relation between Interbook's (Brusilovsky 1998) underlying model and learning space model. Interbook offers a versatile "shell" which also enables easy authoring of the learning material and offers adaptively functioning result. The difference is that both ASK-systems and Interbook use absolute linking. The relative linking offered in the learning space model is different in the sense that it is *unpredictable*; the author cannot know beforehand which seeds and in which order the seeds are presented to the learner. This is the basis for the individual learning paths, but it requires a novel way of thinking the learning material from the author.

Learning materials for the model. The learning space model is open to various learning materials. Typical page-turning metaphors as well as hypermedia constructs are of course possible. Games and simulators can also be embedded into the model. One of the most interesting ways of using the learning space model is to use adaptive hypermedia in a novel manner. The learning space can consist of several small (traditionally linked) hypermedia structures. When the learner is ready to proceed to the suitable, larger hypermedia structure, he or she will "jump" to it. This way, the learning path of an individual in the global learning space is ultimately composed of small steps in local hyperspaces. The steps expand or narrow depending on the learner's action in the space.

Another suitable way to use the learning space model is to exploit the unpredictable nature of the learning session. Cyberliterature offers an ill-defined domain that can be used in learning and experiencing a contemporary art form. The text (i.e. the story) can be broken down into text snippets, which in turn are authored into seeds. The author of the material cannot know in advance how the story unfolds in front of the learners. Therefore, the story is different for every learner, and the way the learners experience it are different. An example of a story using the learning space model could be a fabel filled with moral dilemmas; the way the learner reacts to the situations in the text, the deeper the learner is sucked into the depths where there are no good answers, only less damaging ones.

Testing cognitive abilities with the model. Using a computer as a tool to test human abilities is possible when the limitations of a computer are acknowledged. The computer does not possess such a sense for nuances as human experts for interpreting subtle cues like facial expressions. The key issue is to rely on and exploit the benefits offered by a computerized environment. The test environment with a computer is value-free and the computer acts the same for every testee regardless of the interpersonal or other issues involved in situations where there are two humans present. The computer is not affected by the long hours spent in a testing situation. It might also be possible that using a computer for neuropsychological testing can reveal issues missed in standard face-to-face neuropsychological testing.

The value of the learning space schema to the evaluation of learner's abilities is that the evaluator (teacher) can see the overall picture of the testee by a glimpse at the test results. The challenge is to prepare material that gives information that is usable and accurate enough. It is clear that the test results should not be used as standardized test results, but more as a guideline for the non-expert of neuropsychology so that he or she can benefit from the results. For example, a special teacher in mathematics could use the information during the teaching process. Similar systems for testing exist for special teachers, but the content is often fixed. The power of the learning space model is that the same system can be used as a platform for learning as well as testing, and only different learning spaces have to be prepared. When using the learning space model in the learning environment, the same ease of using the trails as a basis for evaluation applies to assessing learning.

Learning elementary arithmetic with the model. The presented learning space, Matinaut, is an example of using the learning space schema in elementary education. The constructed space is straightforward and simple, even though the model allows a general, more creative use of learning materials. Particularly suitable materials could be the badly-defined domains where there are no right and wrong answers but different possibilities to cope with situations (sometimes referred to as *adventuresome learning*).

The evaluation of the Matinaut-space showed that the learning space model operated as designed, by rapidly taking the learner to the correct area and then providing the learner with learning seeds that are challenging but not too challenging. After reaching the zone of proximal development, the progress is slow but still observable. However, interviews during the study revealed that for some less-able learners, the learner's position in the number field dimension grew too fast. This occurred because the effects were designed so that the position was virtually never decreased in that dimension.

Learning space model for the authors. A significant advantage for a learning system is that the underlying model is simple and usable for non-experts of the system. Therefore, a small-scale study testing the variations between different authors positioning the seeds into a learning space was conducted. The effect for actions was similarily tested.

The study revealed that there are variations between different potential authors. A possible remedy could be to explicitly assign qualitative explanations for different values for each dimension as well as effects for actions *in advance*, so that different authors could rely on unambiguous guidelines for creating a learning space.

Formalising the learning space model. The learning space model and the way it is practically implemented suggests that the learning spaces should be homogeneous. The seventh chapter makes a tentative effort towards formalizing the concepts relating to the learning space model. The discussion is not aimed at providing a thorough treatment, but to open up the possibility of giving a structure for the model.

Formalisation of the learning space schema can help in developing the learning space model further. There are basically two ways to develop the model: to make it simpler (for the author) and to make it more complicated. Possible ways to make the model simpler are restrictions to the model, such as pre-defining some of the dimensions or the steps in them. Ways to make the model more complex are to use partial order (as described in Chapter 7) in dimensions, allowing the author to set the value for dimensions for the learner regardless of the learner's current position in the space, allowing a learner to enter a collection of seeds from other than the initial seed in the collection, alternating the effect in actions according to individual properties, or dynamically arranging the learning space after it has been used enough.

The problem with making the model more complicated is that the author of the material must learn the different possibilities offered, unless authoring tools are offered. The problem with authoring tools is that they tend to restrict the expressiveness of the model. Simpler models tend to be generally more usable (see e.g. Martin & Mitrovic (2002)).

One possible alteration for systems using the learning space model could be that the progress of a learner is visualized for the learners themselves. In addition, the system could pre-analyze the path and the progress, so that the data is in usable form. Experiments with this type of open or scrutable learner models have been promising (see e.g. (Mitrovic & Martin 2002, Carkovski & Kay 2002, Zapata-Rivera & Greer 2002, Hartley & Mitrovic 2002)).

Final words. From the educational technology point of view, special education provides researchers and developers with a challenging laboratory of highly specialized requirements, both technical and pedagogical. Therefore, solutions in the narrow area of special education can also be applied in other areas of education – in much the same way as a telephone is the outcome of research to support deaf people and a tape recorder is the outcome of research to produce talking books for blind people. Highly specialized areas can be the ones with fruitful and wide-spread research outcomes.

As it is presented in this thesis, the learning space model, originally designed for learners with disabilities, is also transferable to learners other than those having difficulties in mental programming.

References

Ainsworth, S. & Grinshaw, S. (2002), Are ITSs Created with the REDEEM Authoring Tool More Effective than 'Dumb' Courseware?, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 883–892.

Ainsworth, S., Underwood, J. & Grimshaw, S. (1999), Formatively Evaluating REDEEM – An Authoring Environment for ITSs, *in* S. Lajoie & M. Vivet, eds, 'Artificial Intelligence in Education', IOS Press, pp. 93–100.

Albert, D. & Hockemeyer, C. (1997), Adaptive and Dynamic Hypertext Tutoring System Based on Knowledge Spacee Theory, *in* B. du Boulay & R. Mizoguchi, eds, 'Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education', IOS Press, Amsterdam, pp. 553–555.

Albert, D. & Hockemeyer, C. (2002), Applying Demand Analysis of a Set of Test Problems for Developing Adaptive Courses, *in* Kinshuk, ed., 'Proceedings of International Conference on Computers in Education (ICCE 2002)', IEEE Computer Society Press, pp. 69–70.

Alessi, S. & Trollip, S. (1985), *Computer-Based Instruction: methods and development*, Prentice-Hall.

Allen, R., ed. (1994), *Encyclopedic English Dictionary*, Chambers.

Anderson, J. & Reiser, B. (1985), 'The LISP Tutor', *Byte* **10**(4), 159–175.

Anderson, J., Boyle, C. & Yost, G. (1986), 'The geometry tutor', *The Journal of Mathematical Behavior* pp. 5–20.

Anderson, J., Boyle, C., Corbett, A. & Lewis, M. (1990), 'Cognitive modeling and intelligent tutoring', *Artificial Intelligence* **42**(1), 7–49.

Anderson, J. R. (1993), *Rules of the Mind*, Lawrence Erlbaum Associates.

Anderson, J. R. (1995), 'Cognitive tutors: Lessons learned', *The Journal of Learning Sciences* **4**, 167–207.

Anonymous (1975), 'Declaration on the Rights of Disabled Persons'. United Nations High Commissioner for Human Rights ,Proclaimed by General Assembly resolution 3447 (XXX) of 9 December 1975 http://www.unhchr.ch/html/menu3/b/72.htm.

Archambault, D. & Burger, D. (2002), The Vickie Project: Visually Impaired Children Kit for Inclusive Education, *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002: proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 90–97.

Aroyo, L. & Dicheva, D. (2001), Concept-based approach to support learning in a Web-based support environment, *in* J. Moore, C. Redfield & W. Johnson, eds, 'Proceedings of World Conference on Artificial Intelligence in Education AI-ED 2001', IOS Press, Leipzig, pp. 1–12.

Atkinson, J. (1964), *An Introduction to Motivation*, Van Norstrand.

Ayala, G. & Yano, Y. (1996), Intelligent Agents to Support the Effective Collaboration in a CSCL Environment, *in* P. Carlson & F. Makedon, eds, 'Proceedings of the ED-TELECOM 96 World Conference on Educational Communications', AACE, AACE, Charlottesville, VA, pp. 19–24.

Baddeley, A. (1997), *Human Memory Theory and Practice*, Psychology Press, Midsomer Norton, Somerset.

Bell, B. (1996), A Special-Purpose Architecture for the Design of Eductional Software, PhD thesis, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Boyle, T. (1997), *Design for Multimedia Learning*, Prentice-Hall.

Brusilovsky, P. (1996), 'Methods and techniques of adaptive hypermedia', *User Modeling and User-Adapted Interaction* **6**(2-3), 87–129.

Brusilovsky, P. (1998), Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies, *in* 'Proceedings of Workshop WWW-Based Tutoring at 4th International Conference on Intelligent Tutoring Systems (ITS'98)', San Antonio, Texas. http://www-aml.cs.umass.edu/∼stern/webits/itsworkshop/.

Brusilovsky, P. (2001), 'Adaptive Hypermedia', *User Modeling and User-Adapted Interaction* **11**(1–2), 87–110.

Brusilovsky, P. & Eklund, J. (1998*a*), 'A Study of User Model Based Link Annotation in Educational Hypermedia', *Journal of Universal Computer Science* **4**(4), 429–448.

Brusilovsky, P. & Eklund, J. (1998*b*), The value of adaptivity in hypermedia learning environments: A short review of empirical evidence, *in* 'Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia Held in Conjunction with HYPERTEXT 98: The Ninth ACM Conference on Hypertext and Hypermedia', Computing Science Reports No. 98/12, Eindhoven University of Technology, Eindhoven, pp. 13–19. http://wwwis.win.tue.nl/ah98/Proceedings.html.

Brusilovsky, P. & Pesin, L. (1995), Visual annotation of links in adaptive hypermedia, *in* I. Katz, R. Mack & L. Marks, eds, 'Proceedings of CHI'95 Conference Companion', ACM, ACM Press, pp. 222–223.

Brusilovsky, P., Eklund, J. & Schwarz, E. (1998), 'Web-based education for all: A tool for developing adaptive courseware', *Computer Networks and ISDN Systems. (Proceedings of Seventh International World Wide Web Conference)* **30**(1-7), 291–300.

Brusilovsky, P., Schwartz, E. & Weber, G. (1996*a*), ELM-ART: An intelligent tutoring system on World Wide Web, *in* C. Frasson, G. Gauthier & G. McCalla, eds, '3rd Int'l Conference on Intelligent Tutoring Systems (ITS'96)', Springer-Verlag, Montreal.

Brusilovsky, P., Schwarz, E. & Weber, G. (1996*b*), A Tool for Developing Adaptive Electronic Textbooks on WWW, *in* H. Maurer, ed., 'Proceedings of WebNet '96 - World Conference of the Web Society', AACE, AACE, San Francisco, CA, pp. 64–69.

Buaud, A., Svensson, H., Archambault, D. & Burger, D. (2002), Multimedia games for visually impaired children, *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002:

proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 173–180.

Bush, V. (1945), 'As We May Think', *Atlantic Monthly* **176**(1), 101–108. Available at: http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm.

Carbonell, J. (1970), Mixed-Initiative Man-Computer Instructional Dialogues, PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Carkovski, M. & Kay, J. (2002), A scrutable adaptive hypertext, *in* P. de Bra, P. Brusilovsky & R. Conejo, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May, 2002: proceedings', Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 384–387.

Carro, R., Breda, A., Castillo, G. & Bajuelos, A. (2002), A methodology for developing adaptive educational-game environments, *in* P. de Bra, P. Brusilovsky & R. Conejo, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May, 2002: proceedings', Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 90–99.

Carro, R., Pulido, E. & Rodriguez, P. (1999), Object Orientation in the TANGOW System: Working with Dynamic Task Trees, *in* 'World Conference on the WWW and Internet, WebNet '99, Proceedings', AACE, AACE, pp. 171–176.

Chan, T.-W. (1991), Integration-kid: A learning companion system, *in* R. Myopoulos and J. Reiter, ed., 'Proceedings of the 12th International Joint Conference on Artificial Intelligence', Morgan Kaufmann, Sydney, Australia, pp. 1094–1099.

Chan, T.-W. & Baskin, A. (1990), Learning Companion Systems, *in* C. Frasson & G. Gauthier, eds, 'Intelligent Tutoring Systems', Ablex Publishin Corp., pp. 6–33.

Cheikes, B. (1995), Should ITS Designers Be Looking For A Few Good Agents?, *in* N. Major, T. Murray & C. Bloom, eds, 'Proceedings of a workshop on Authoring Shells for Intelligent Tutoring Systems in conjunction of AI-ED 95 conference'. www.pitt.edu/∼al/aied/workshop.html.

Conati, C. & VanLehn, K. (2000), Further results from the evaluation of an intelligent computer tutor to coach self-explanation, *in* G. Gauthier, C. Frasson & K. VanLehn, eds, 'Intelligent tutoring systems: Fifth International Conference, ITS 2000 Montreal, Canada, June 2000: proceedings', Vol. 1839 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 304–313.

da Silva, D. P., Durm, R. V., Duval, E. & Olivié, H. (1998), Concepts and documents for adaptive educational hypermedia: a model and a prototype, *in* 'Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia Held in Conjunction with HYPERTEXT 98: The Ninth ACM Conference on Hypertext and Hypermedia', Computing Science Reports No. 98/12, Eindhoven University of Technology, Eindhoven. http://wwwis.win.tue.nl/ah98/Proceedings.html.

Damasio, A., Tranel, D. & Damasio, H. (1991), Somatic markers and the guidance of behavior: Theory and preliminary testing, *in* H. Levin, H. Eisenberg & A. Bentol, eds, 'Frontal Lobe Function and Dysfunction', Oxford University Press, pp. 217–229.

de Bra, P. (2000), 'Pros and Cons of Adaptive Hypermedia in Web-Based Education', *Journal on CyberPsychology and Behavior* **3**(1), 71–77.

de Bra, P. & Calvi, L. (1998), AHA: a Generic Adaptive Hypermedia System, *in* 'Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia Held in Conjunction with HYPERTEXT 98: The Ninth ACM Conference on Hypertext and Hypermedia', ACM, Eindhoven University of Technology, Eindhoven. http://wwwis.win.tue.nl/ah98/Proceedings.html.

DeMeglio, M., Hakkinen, M. & Kawamura, H. (2002), Accessible interface design: Adaptive multimedia information system (amis), *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002: proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 406–412.

Doignon, J.-P. & Falmagne, J.-C. (1985), 'Spaces for the assessment of knowledge', *International Journal of Man-Machine Studies* **23**(2), 175–196.

Dooley, S., Meiskey, L., Blumenthal, R. & Sparks, R. (1995), Developing reusable intelligent tutoring system shells, *in* N. Major, T. Murray &

C. Bloom, eds, 'Proceedings of a workshop on Authoring Shells for Intelligent Tutoring Systems in conjunction of AI-ED 95 conference'. www.pitt.edu/∼al/aied/workshop.html.

Dowling, C. & Hockemeyer, C. (2000), Integrating knowledge spaces obtained by querying different experts, *in* M. F. Janowitz, ed., 'Electronic Notes in Discrete Mathematics', Vol. 2, Elsevier Science Publishers.

Dowling, C., Hockemeyer, C. & Ludwig, A. (1996), Adaptive Assessment and Training Using the Neighbourhood of Knowledge States, *in* C. Frasson, G. Gauthier & A. Lesgold, eds, 'Intelligent tutoring systems: Third International Conference, ITS '96 Montréal, Canada, June 12–14, 1996: proceedings', Lecture Notes in Computer Science, Springer-Verlag, pp. 578–586.

Edwards, A., Arato, A. & Zagler, W. (1998), *Computers and Assistive Technology ICCHP'98, Proceedings of the XV. IFIP World Computer Congress*, IFIP/OCG.

Edwards, A., ed. (1995), *Extra-Ordinary Human-Computer Interaction*, Cambridge series on human-computer interaction: 7, Cambridge University Press.

Eliot, . C., Woolf, B. P., Arroyo, I. & Beck, J. (2001), Software support for students with attention problems., *in* J. Moore, C. Redfield & W. Johnson, eds, 'Proceedings of World Conference on Artificial Intelligence in Education AI-ED 2001', IOS Press, Leipzig, pp. 467–475.

Eriksson, T., Kurhila, J. & Sutinen, E. (1997), An agent-based framework for special needs education, *in* G. Grahne, ed., 'Sixth Scandinavian Conference on Artificial Intelligence SCAI '97', IOS Press, Amsterdam, pp. 270–271.

Falmagne, J.-C. (1993), 'Stochastic learning paths in a knowledge structure', *Journal of Mathematical Psychology* **37**(4), 489–512.

Falmagne, J.-C., Doignon, J.-P., Koppen, M., Villano, M. & Johannesen, L. (1990), 'Introduction to Knowledge Spaces: How to Build, Test, and Search Them', *Psychological Review* **97**(2), 201–224.

Ferguson, W., Bareiss, R., l. Birnbaum & Osgood, R. (1992), 'ASK Systems: An Approach to the Realization of Story-based Teachers', *Journal of the Learning Sciences* **2**, 95–134.

Gardner, C. & Lundquist, R. (1998), MathPlusToolBox, a computer application for learning basic math skills, *in* A. Edwards, A. Arato & W. Zagler, eds, 'Computers and Assistive Technology ICCHP'98, Proceedings of the XV. IFIP World Computer Congress', IFIP, IFIP/OCG, Vienna, pp. 231–240.

Gerdt, P., Kurhila, J., Meisalo, V., Suhonen, J. & Sutinen, E. (2002), Rambling through the Wilds: the Concept of Jagged Study Zones, *in* 'Proceedings of ED-MEDIA 2002', AACE, AACE, Charlottesville, VA, pp. 596–601.

Gouli, E., Papanikolaou, K. & Grigoriadou, M. (2002), Personalized assessment in adaptive educational hypermedia systems, *in* P. de Bra, P. Brusilovsky & R. Conejo, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May, 2002: proceedings', Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 153–163.

Gravano, L., Garcia-Molina, H. & Tomasic, A. (1999), 'Gloss: Text-source discovery over the internet', *ACM Transactions on Database Systems* **24**(2), 229–264.

Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S. & Kettel, L. (2001), Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience, *in* J. Moore, C. Redfield & W. Johnson, eds, 'Proceedings of World Conference on Artificial Intelligence in Education AI-ED 2001', IOS Press, Leipzig, pp. 410–421.

Grinstein, L. & Lipsey, S. (2001), *Encyclopedia of mathematics education*, Routledge, New York.

Hammerton, R. L. L. (2002), Getting to know me: Helping learners understand their own learning needs through metacognitive scaffolding, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 759–771.

Hartley, D. & Mitrovic, A. (2002), Supporting learning by opening the student model, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 453–462.

Heller, R. (1991), 'Evaluating Software: A Review of the Options', *Computers and Education* **17**(4), 285–291.

Hietala, P. & Niemirepo, T. (1996), Studying learner-computer interaction in agent-based social learning environments, *in* P. Brna, A. Paiva & J. Self, eds, 'Proceedings of the European Conference on Artificial Intelligence in Education', Colibri, Lisbon, Portugal, pp. 386–392.

Hietala, P. & Niemirepo, T. (1998), 'The competence of learning companion agents', *International Journal of Artificial Intelligence in Education* **9**(3/4), 178–192.

Höök, K. (2000), 'Steps to take before IUIs become real', *Journal of Interacting with Computers* **12**(4), 409–426.

Huang, S. (1996), A Content-Balanced Adaptive Testing Algorithm for Computer-Based Testing Systems, *in* C. Frasson, G. Gauthier & G. McCalla, eds, '3rd Int'l Conference on Intelligent Tutoring Systems (ITS'96)', Springer-Verlag, Montreal, pp. 306–314.

Hübschner, R. & Puntambekar, S. (2002), Adaptive navigation for learners in hypermedia is scaffolded navigation, *in* P. de Bra, P. Brusilovsky & R. Conejo, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May, 2002: proceedings', Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 184–192.

Järvinen, A. (1999), 'Are you sure you want to quit? (in Finnish)', *Parnasso* **49**(3), 372–379.

Jona, M. (1995), Representing and Applying Teaching Strategies in Computer-Based Learning-By-Doing Tutors, PhD thesis, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Kachelhoffer, A. (1996), The Use of Computer Assisted Education for Children with Dows Syndrome, *in* 'Proceedings of ICCHP '96', OCG, OCG, Linz, Austria, pp. 559–567.

Kelly, D. & Tangney, B. (2002), Incorporating learning characteristics into an intelligent tutor, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 729–738.

Ketamo, H. (2002), User and Platform Adaptation in Web-based Learning Environments, PhD thesis, Tampere University of Technology, Tampere, Finland.

Kinnunen, R., Lehtinen, E. & Vauras, M. (1994), Evaluating the mathematical ability (in Finnish, *in* M. Vauras, E. Poskiparta & P. Niemi, eds, 'The evaluation of the cognitive abilities and motivation with the learners in the 1st grade (in Finnish)', Publications 3, University of Turku, Centre for Learning Research, Turku, Finland, pp. 55–76.

Kiswarday, V. (1996), "A-B-C and 1-2-3 on Computers" - Computer as a Didactic Tool in Education of Children with a Severe Mental Retardation, *in* ' 5th International Conference on Computers Helping People with Special Needs ICCHP '96', IFIP/OCG, pp. 689–695.

Klaus, J., Auff, E., Kremser, W. & Zagler, W. (1996), *Proceedings of the 5th International Conference on Computers Helping People with Special Needs (ICCHP'96)*, IFIP/OCG.

Koedinger, K., Anderson, J., Hadley, W. & Mark, M. (1997), 'Intelligent tutoring goes to school in the big city', *International Journal of Artificial Intelligence in Education* **8**(1), 30–43.

Korkman, M. (1988), Nepsy, A Proposed Neuropsychological Test Battery for Young Developmentally Disabled Children, Theory and Evaluation, PhD thesis, University of Helsinki, Helsinki, Finland.

Korkman, M. (1997), Applying Luria's method to children: the development of NEPSY-tests, *in* T. Ahonen, T. Korhonen, T. Riita, M. Korkman & H. Lyytinen, eds, 'Brains and learning: Clinical children's neuropsychology (in Finnish)', Niilo Mäki Instituutti, Atena Kustannus, chapter 4, pp. 52–76.

Kuikka, P., Pulliainen, V. & Hänninen, R., eds (1991), *Introduction to Neuropsychology (in Finnish)*, Werner-Söderström.

Kurhila, J. (2003), Building Adaptive Learning Material with Distinct Authors, to appear in Proceedings of 10th International Conference on Human-Computer Interaction (HCI International 2003), Lawrence Erlbaum Associates.

Kurhila, J. & Laine, T. (2000), 'Individualized Special Education with Cognitive Skill Assessment', *British Journal of Educational Technology* **31**(2), 163–170.

Kurhila, J. & Sutinen, E. (1998), Agents in an Adaptive Learning Environment for Special Needs Education, *in* A. Edwards, A. Arato & W. Zagler, eds, 'Computers and Assistive Technology ICCHP '98, Proceedings of the XV. IFIP World Computer Congress', IFIP, IFIP/OCG, Vienna, pp. 241–249.

Kurhila, J. & Sutinen, E. (1999), 'Sharing an open learning space by individualizing agents', *Journal of Interactive Learning Research* **10**(3-4), 287–300.

Kurhila, J. & Sutinen, E. (2000*a*), Adaptivity in AHMED, *in* P. Brusilovsky, O. Stock & C. Strapparava, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: International Conference, AH 2000 Trento, Italy, August 2000: proceedings', Vol. 1892 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 351–354.

Kurhila, J. & Sutinen, E. (2000*b*), From intelligent tutoring systems to intelligent learning material, *in* J. Bourdeau & R. Heller, eds, 'Proceedings of ED-MEDIA 2000', AACE, AACE, Charlottesville, VA, pp. 511–516.

Kurhila, J. & Varjola, H. (2002), Using adaptive hypermedia to evaluate basic arithmetic skills in special education, *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002: proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 98–106.

Kurhila, J., Lattu, M. & Pietilä, A. (2002), Using vector-space model in adaptive hypermedia for learning, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 129–138.

Kurhila, J., Sutinen, E. & Torres, J. T. (2001), How to Model a Platform for Adaptive Learning Environment?, *in* C.-H. Lee, ed., 'Proceedings of ICCE/SchoolNet 2001', AACE, Incheon National University of Education, Seoul, Korea, pp. 416–423.

Kurhila, J., Sutinen, E., Jokinen, S., Nyman, R. & Väisänen, P. (1998), Developing an Adaptive Learning Environment for the Disabled, *in* I. P. Porrero & E. Ballabio, eds, 'Improving the Quality of Life for the European Citizen. Proceedings of the 3rd Technology for Inclusive Design and Equality Congress TIDE '98', Assistive Technology Series 4, IOS Press, pp. 126–130.

Luria, A. (1973), *The Working Brain*, Penguin Modern Psychology Texts, Penguin Books.

Luria, A. (1979), *Psychology and the development history of mental activity (Finnish translation, originally published in the Soviet Union 1975)*, Kansankulttuuri.

Major, N. (1995), REDEEM: Creating Reusable Intelligent Courseware, *in* J. Greer, ed., 'Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education', AACE, AACE, Charlottesville, VA, pp. 75–82.

Major, N. & Reichgelt, H. (1991), 'Using COCA to build an intelligent tutoring system in simple algebra', *Intelligent Tutoring Media* **2**(3/4), 159–169.

Major, N., Ainsworth, S. & Wood, D. (1997), 'REDEEM: Exploiting symbiosis between psychology and authoring environments', *International Journal of Artificial Intelligence in Education* **8**(3/4), 317–340.

Martin, B. & Mitrovic, A. (2002), Automatic problem generation in constraint-based tutors, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 388–398.

Miesenberger, K., Klaus, J. & Zagler, W., eds (2002), *Computers Helping People with Special Needs*, Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag.

Mitrovic, A. & Martin, B. (2002), Evaluating the Effects of Open Student Models on Learning, *in* P. de Bra, P. Brusilovsky & R. Conejo, eds, 'Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May, 2002: proceedings', Vol. 2347 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 296–305.

Mohamad, Y., Hammer, S., Haverkamp, F., Nöker, M. & Tebarth, H. (2002), Evaluational study: Training with animated pedagogical agents, *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers

Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002: proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 117–124.

Moreno, L., González, C., Munoz, V., Estevez, J., Aguilar, R., Sanchez, J., j. Sigut & Pineiro, J. (2002), Integrating multimedia technology, knowledge based system, and speech processing for the diagnostic and treatment of developmental dyslexia, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 168–177.

Munro, A., Höök, K. & Benyon, D., eds (1999), *Social Navigation in Information Space*, Springer-Verlag.

Murray, T. (1996*a*), From Story Boards to Knowledge Bases, the First Step in Making CAI 'Intelligent', *in* Carlson & Makedon, eds, 'Proceeding of ED-MEDIA 96', AACE, AACE, Charlottesville, pp. 509–514.

Murray, T. (1996*b*), Having it All, Maybe: Design Tradeoffs in ITS Authoring Tools, *in* C. Frasson, G. Gauthier & A. Lesgold, eds, 'Proc. of Intelligent Tutoring Systems (ITS'96)', Springer-Verlag, Berlin, pp. 91–101.

Murray, T. (1999), 'State-of-the-Art in Intelligent authoring tools', *International Journal of Artificial Iintelligence in Education* **10**, 98–129.

Murray, T. & Arroyo, I. (2002), Towards measuring and maintaining the zone of proximal development, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 749–758.

Nakabayashi, K., Hoshide, T., Seshimo, H. & Fukuhara, Y. (1998), An Object-Oriented Architecture for a Web-based CAI System, *in* 'Proceedings of ED-MEDIA/ED-TELECOM98', AACE, AACE, Freiburg, Germany.

Nakano, A., Hirashima, T. & Takeuchi, A. (2002), An evaluation of intelligent learning environment for problem posing, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002:

proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 861–872.

Newell, A. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Newell, A. & Simon, H. A. (1972), *Human Problem Solving*, Prentice-Hall, New Jersey.

Nwana, H. (1993*a*), 'An approach to developing intelligent tutors in mathematics', *Computers & Education* **20**(1), 27–43.

Nwana, H. (1993*b*), The Anatomy of FITS: a Mathematics Tutor, *in* H. Nwana, ed., 'Mathematical Intelligent Learning Environments', Intellect, chapter 5, pp. 79–109.

Osgood, R. (1994), The Conceptual Indexing of Conversational Hypertext, PhD thesis, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Paiva, A. (1996), Learner Modelling Agents, *in* P. Brna, A. Paiva & J. Self, eds, 'Proceedings of the European Conference on Artificial Intelligence in Education', Colibri, Lisbon, Portugal, pp. 261–268.

Perez, T., Gutierrez, J. & Lopisteguy, P. (1995), An adaptive hypermedia system, *in* J. Greer, ed., 'Proceedings of AI-ED'95 7th World Conference on Artificial Intelligence in Education', AACE, AACE, Charlottesville, VA, pp. 351–358.

Rickel, J. & Johnson, W. (1997), Integrating pedagogical capabilities in a virtual environment agent, *in* W. Johnson & B. Hayes-Roth, eds, 'Proceedings of the First International Conference on Autonomous Agents', ACM, ACM Press, pp. 30–38.

Ritter, S., Anderson, J., Cytrynowicz, M. & Medvedeva, O. (1998), 'Authoring Content in the PAT Algebra Tutor', *Journal of Interactive Media in Education.* http://www-jime.open.ac.uk/.

Rosenberg, R. (1987), 'A critical analysis of research on intelligent tutoring systems', *Educational Technology* **27**(11), 7–13.

Ross, D. & Blasch, B. (2000), Wearable Interfaces for Orientation and Wayfinding, *in* 'Proceedings of the ASSETS 2000: The fourth ACM Conference on Assistive Technology', ACM, ACM Press, New York, NY, pp. 193–200.

Salton, G., Wong, A. & Yang, C. (1975), 'A Vector Space Model for Automatic Indexing', *Communications of the ACM* **18**(11), 613–620.

Sandberg, M. (1999), Attentiveness-hyperaction disorder and its medical care (in Finnish), *in* T. Ahonen & T. Aro, eds, 'Learning disabilities (in Finnish)', Werner-Söderström.

Schank, R. (1990), Teaching Architectures, Technical Report 3, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Schank, R. & Cleary, C. (1994), 'Engines for Education'. http://www.ils.nwu.edu/∼e_for_e/.

Schank, R. & Osgood, R. (1992), The Communications Story, Technical Report 37, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Schank, R., Fano, A., Jona, M. & Bell, B. (1993), The Design of Goal-Based Scenarios, Technical Report 39, The Institute for the Learning Sciences, Northwestern University, Chicago, IL.

Self, J. (1995), 'Computational Mathetics: Towards a Science of Learning System Design. Computer Based Learning Unit, University of Leeds'. http://cbl.leeds.ac.uk/∼jas/cm.html.

Shaw, E., Johnson, W. & Ganeshan, R. (1999), Pedagogical Agents on the Web, *in* O. Etzioni, J. Müller & J. Bradshaw, eds, 'Proceedings of the third annual conference on on Autonomous Agents', ACM, ACM Press, pp. 283–290.

Smith, A., Francioni, J. & Matzek, S. (2000), A Java programming tool for students with visual disabilities, *in* 'Proceedings of the ASSETS 2000: The fourth ACM Conference on Assistive Technology', ACM, ACM Press, New York, NY, pp. 142–148.

Smith, J. & Weiss, S. (1988), 'Hypertext', *Communications of the ACM* **31**(7), 816–819.

Soloway, E. (1998), 'No-one is making money on educational software', *Communications of the ACM* **41**(2), 11–15.

Specht, M., Weber, G., Heitmeyer, S. & Schöch, V. (1997), AST: adaptive WWW-courseware for statistics, *in* P. Brusilovsky, J. Fink & J. Kay, eds, 'Proceedings of Workshop Adaptive Systems and User Modeling on the World Wide Web at 6th International Conference on User Modeling, (UM'97)', Chia Laguna, Sardinia, Italy, pp. 91–95.

Squires, D. & McDougall, A. (1994), *Choosing and Using Educational Software*, The Farmer Press.

Steinberg, E. (1991), *Computer-assisted instruction : a synthesis of theory, practice, and technology*, Lawrence Erlbaum Associates.

Stern, M. (2001), Using Adaptive Hypermedia and Machine Learning to Create Intelligent Web-based Courses, PhD thesis, University of Massachusetts, Amherst, MA.

Tebarth, H., Mohamad, Y. & Pieper, M. (2000), Cognitive Training by Animated Pedagogical Agents (TAPA) Development of a Tele-Medical System for Memory Improvement in Children with Epilepsy, *in* P. Emiliani & C. Stephanidis, eds, '6th ERCIM Workshop on User Interfaces for All', CNR-IROE. http://ui4all.ics.forth.gr/UI4ALL-2000/proceedings.html.

Treviranus, J. (2002), Making yourself at home — portable personal access preferences, *in* K. Miesenberger, J. Klaus & W. Zagler, eds, 'Computers Helping People with Special Needs: 8th International Conference, ICCHP 2002 Linz, Austria, July 2002: proceedings', Vol. 2398 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 643–648.

Turkle, S. (1996), *Life on the Screen. Identity in the Age of the Internet*, Weidenfeld and Nicolson.

van de Meere, J. (1996), The role of attention, *in* M. Sandberg, ed., 'Hyperactivity disorders of childhood', Cambridge Monograps in Child and Adolescent Psychiatry, Cambridge University Press, pp. 69–109.

van Marcke, K. (1992), Instructional Expertise, *in* C. Frasson, G. Gauthier & G. McCalla, eds, '2rd Int'l Conference on Intelligent Tutoring Systems (ITS'92)', Springer-Verlag, pp. 234–243.

VanLehn, K., Lynch, C., Linwood, T., Weinstein, A., Shelby, R., Schulze, K., Treacy, D. & Wintersgill, M. (2002), Minimally invasive tutoring of complex physics problem solving, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 367–376.

Vassileva, J. (1990), 'An Architecture and Methodology for Creating a Domain-Independent Plan-Based Intelligent Tutoring System', *Education and Training Technology International* **27**(4), 386–397.

Vassileva, J. (1992), Dynamic CAL-courseware generation within an ITS-shell architecture, *in* I. Tomek, ed., 'Proceedings of Computer Assisted Learning (ICCAL '92)', Vol. 602 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 581–591.

Vassileva, J. (1997), Dynamic course generation on the WWW, *in* B. Boulay & R. Mizoguchi, eds, 'Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education', IOS Press, Amsterdam, pp. 498–505.

Vilkki, J. (1990), Disorganization of mental programming after frontal lobe damage, *in* M. Hietanen, J. Vilkki, M.-L. Niemi & M. Korkman, eds, 'Clinical Neuropsychology. Excursions into the field in Finland', Acta Psychologica Fennica, Monographies in Applied Psychology 4, The Finnish Psychological Society, pp. 60–64.

Vilkki, J. (1995), ' Neuropsychology of mental programming: an approach for the evaluation of frontal lobe dysfunction', *Applied Neuropsychology* **2**, 93–106.

Virsu, V. (1991), *The shaping and rehabilitation of the brain (in Finnish)*, Research of Rehabilition Foundation 26, Helsinki University Press.

Vygotsky, L. (1978), *Mind in Society: The development of higher psychological processes*, Harvard University Press, Cambridge, MA.

Weber, G. & Möllenberg, A. (1994), ELM-PE: A knowledge-based programming environment for learning LISP, *in* T. Ottmann & I. Tomek, eds, 'Proceedings of ED-MEDIA '94', AACE, AACE, Charlottesville, VA, pp. 557–562.

Weber, G. (1996), 'Episodic learner modeling', *Cognitive Science* **20**, 195–236.

Weber, G. & Brusilovsky, P. (2001), 'ELM-ART: An adaptive versatile system for Web-based instruction', *International Journal of Artificial Intelligence in Education* **12**(4), 351–384.

Weber, G., Bögelsack, A. & Wender, K. (1993), When can individual student models be useful?, *in* G. Strube & K. Wender, eds, 'The cognitive psychology of knowledge. The German Wissenspsychologie project', North-Holland, pp. 263–284.

Wenger, E. (1987), *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann.

Zapata-Rivera, D. & Greer, J. (2002), Exploring various guidance mechanisms to support interaction with inspectable learner models, *in* S. Cerri, G. Gouardères & F. Paraguacu, eds, 'Intelligent tutoring systems: Sixth International Conference, ITS 2002 Biarritz, France, June 2002: proceedings', Vol. 2363 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 442–452.

ISSN 1238-8645
ISBN 952-10-1201-3 (paperback)
ISBN 952-10-1202-1 (PDF)
Helsinki 2003
Helsinki University Printing House

Jaakko Kurhila: Considering Individual Differences in Computer-Supported Special and Elementary Education A-2003-5