Int J Artif Intell Educ (2013) 23:115–135
DOI 10.1007/s40593-013-0003-7

RESEARCH ARTICLE

Modelling and Optimizing Mathematics Learning in Children

Tanja Käser · Alberto Giovanni Busetto · Barbara Solenthaler · Gian-Marco Baschera · Juliane Kohn · Karin Kucian · Michael von Aster · Markus Gross

Published online: 9 November 2013

Abstract This study introduces a student model and control algorithm, optimizing mathematics learning in children. The adaptive system is integrated into a computer-based training system for enhancing numerical cognition aimed at children with developmental dyscalculia or difficulties in learning mathematics. The student model consists of a dynamic Bayesian network which incorporates domain knowledge and enables the operation of an online system of automatic control. The system identifies appropriate tasks and exercise interventions on the basis of estimated levels of accumulated knowledge. Student actions are evaluated and monitored to extract statistical patterns which are useful for predictive control. The training system is adaptive and personalizes the learning experience, which improves both success and motivation. Comprehensive testing of input data validates the quality of the obtained results and confirms the advantage of the optimized training. Pilot results of training effects are included and discussed.

T. Käser (✉) · A. G. Busetto · B. Solenthaler · G.-M. Baschera · M. Gross
Department of Computer Science, ETH Zurich, Zurich, Switzerland
e-mail: kaesert@inf.ethz.ch

A. G. Busetto
Competence Center for Systems Physiology and Metabolic Diseases, Zurich, Switzerland

K. Kucian · M. von Aster
Center for MR-Research, University Children's Hospital Zurich, Zurich, Switzerland

J. Kohn · M. von Aster
Department of Psychology, University of Potsdam, Potsdam, Germany

M. von Aster
Department of Child and Adolescent Psychiatry, German Red Cross Hospitals Westend, Berlin, Germany

Keywords Learning · Control theory · Optimization · Dynamic Bayesian network · Dyscalculia

Introduction

Arithmetic skills are important in modern society, as numerical cognition and calculations are omnipresent in everyday life. However, many children suffer from difficulties in learning mathematics. Developmental dyscalculia (DD) is a specific learning disability affecting the acquisition of arithmetic skills (von Aster and Shalev 2007). Genetic, neurobiological, and epidemiological evidence indicates that DD is a brain-based disorder, although poor teaching and environmental deprivation might also be relevant (Shalev 2004). Children with DD show a deficit in basic numerical skills such as number comparison (Butterworth 2005a, b; Landerl et al. 2004; Rubinsten and Henik 2005) and exhibit fundamental problems in number processing (Cohen Kadosh et al. 2007; Kucian et al. 2006; Mussolin et al. 2010; Price et al. 2007). Furthermore, they tend to experience difficulties in acquiring arithmetic procedures and show a deficit in fact retrieval (Geary et al. 1992; Ostad 1997, 1999). DD has an estimated prevalence of 3–6 % in English- and German-speaking countries (Badian 1983; Lewis et al. 1994; Shalev and von Aster 2008).

The relatively high prevalence of DD suggests that it is important to investigate intervention approaches to prevent or remediate learning difficulties in mathematics. The range of existing interventions includes remedial programs for elementary school children (Dowker 2001; Kaufmann et al. 2003; Kucian et al. 2011; Lenhard et al. 2011; Wilson et al. 2006) as well as preventive programs for pre-school children at risk of developing mathematical difficulties (Griffin et al. 1994; Van De Rijt and Van Luit 1998; Wright 2003). However, only a few of these programs are computer-based (and have been scientifically evaluated). ‘Number Race’ (Wilson et al. 2006) focuses on the training of basic numerical skills, while ‘Rescue Calcularis’ (Kucian et al. 2011) combines the training of basic-numerical abilities with the training of arithmetic skills. ‘Elfe and Mathis’ (Lenhard et al. 2011) aligns the training to the German scholar curriculum. All of these approaches are carefully designed for children with difficulties in learning mathematics, however, they lack user adaptation.

Yet, adaptability is very important for children suffering from learning disabilities as these children are highly heterogeneous and thus a high grade of individualization is necessary. Intelligent tutoring systems can contribute to this need. Current systems use approaches such as knowledge tracing (Corbett and Anderson 1994), performance factors analysis (Pavlik et al. 2009a, b) and Bayesian networks (Mislevy et al. 1999) to estimate, assess and predict the knowledge of the user. In the domain of mathematics, existing systems mainly focus on specific aspects of the domain (Koedinger et al. 1997; Mislevy et al. 1999; Rau et al. 2009). Previous work exists not only for student models, but also for control mechanisms. A plethora of advanced control approaches aimed at optimization of complex mechanisms was proposed (Garcia et al. 1989). Controllers can be based upon explicit models obtained through intervention-driven identification (Busetto and Buhmann 2009).

Related predictive models aimed at treating learning disabilities have been introduced for spelling learning (Baschera et al. 2011; Baschera and Gross 2010).

The present study is based on the intelligent tutoring system 'Calcularis' (Käser et al. 2012). In this system, we model the cognitive processes of mathematical development using a dynamic Bayesian network. Our student model represents different mathematical skills and their dependencies. An automatic control mechanism aimed at optimizing learning and acting on the skill net is introduced. The design of the skill net allows for a non-linear control mechanism. In contrast to previous approaches, we allow movements along all edges of the skill net (particularly also backward movements), which enables us to implicitly model forgetting and knowledge gaps. The model's predictive control enables a significant level of cognitive stimulation which is user- and context-adaptive. We assess the efficiency and adaptability of the introduced student model and control mechanism based on input logs from two user studies in Germany and Switzerland. Furthermore, we analyse properties of users and skills used in the model. Finally, we also include first pilot results of the obtained training effects.

Training Environment

Current neuropsychological models postulate the existence of task-specific representational modules located in different areas of the brain. The functions of these modules are relevant to both adult cognitive number processing and calculation (Dehaene 2011). Dehaene's triple-code model (Dehaene 1995) presumes three representational modules (verbal, symbolic, and analogue magnitude) related to number processing. These modules develop hierarchically over time (von Aster and Shalev 2007) and the overlap of the number representations increases with growing mathematical understanding (Kucian and Kaufmann 2009). The development of numerical abilities follows a subject-dependent speed which is influenced by the development of other cognitive as well as domain general abilities and biographical aspects (von Aster and Shalev 2007). Hence, when teaching mathematics, a substantial degree of individualization may not only be beneficial, but even necessary. The introduced computer-based training addresses these challenges by

1. structuring the curriculum on the basis of the natural development of mathematical understanding (hierarchical development of number processing).
2. introducing a highly specific design for numerical stimuli enhancing the different representations and facilitating understanding. The different number representations and their interrelationships form the basis of number understanding and are often perturbed in dyscalculic children (von Aster and Shalev 2007).
3. training operations and procedures with numbers. Dyscalculic children tend to have difficulties in acquiring simple arithmetic procedures and show a deficit in fact retrieval (Geary et al. 1992; Ostad 1997, 1999).
4. providing a fully adaptive learning environment. Student model and controlling algorithm optimize the learning process by providing an ideal level of cognitive stimulation.

The training program is composed of multiple games in a hierarchical structure. Games are structured according to number ranges and further grouped into two areas. The first area (*Part A*) focuses on “number representations and understanding”. It trains the transcoding between alternative representations and introduces the three principles of number understanding: cardinality, ordinality, and relativity. Games in this area are structured according to current neuropsychological models (von Aster and Shalev 2007; Dehaene 1995). The first area is exemplified by the LANDING game (Fig. 1a). In this game, children need to indicate the position of a given number on a number line. To do so, a falling cone has to be steered using a joystick. The second area (*Part B*) is that of “cognitive operations and procedures with numbers”, which aims at training concepts and automation of arithmetical operations. This is illustrated by the PLUS-MINUS game (Fig. 1b). Children solve addition and subtraction tasks using blocks of tens and ones to model them. The different games are categorized according to their complexity and relative importance. Main games require a combination of abilities to solve them, while support games train specific skills and serve as basic prerequisites. Difficulty estimation and hierarchy result from the development of mathematical abilities.

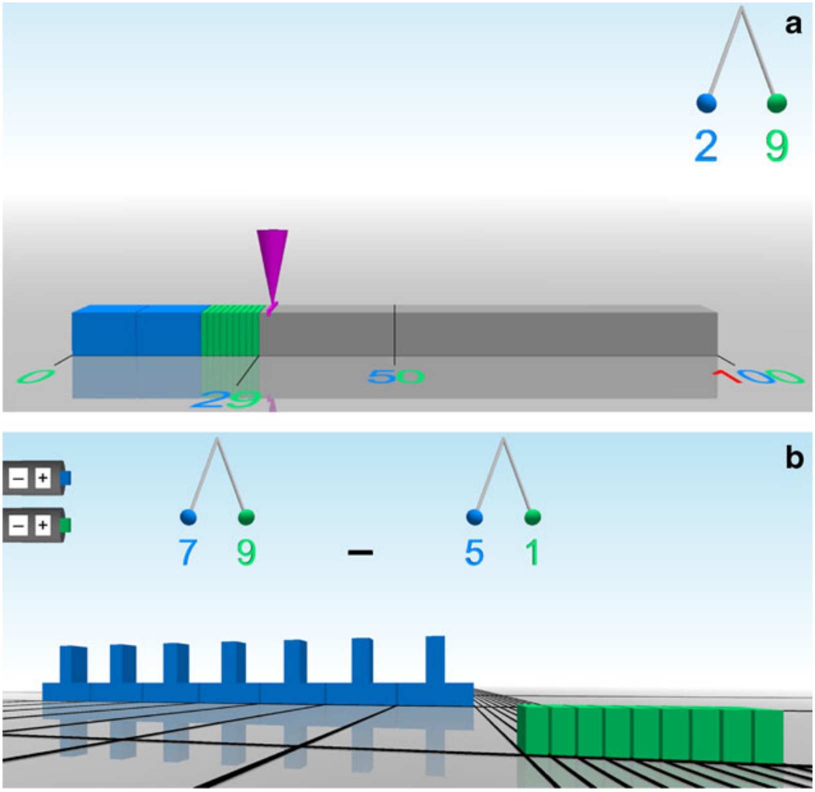


Fig. 1 In the LANDING game (**a**), the position of the displayed number (29) needs to be indicated on the number line. In the PLUS-MINUS game (**b**), the task displayed needs to be modelled with the blocks of tens and ones

Selection of Actions

A fundamental component of the intelligent tutoring system is its pedagogical module: the subsystem making the teaching decisions. It selects the skills for training and determines the actions for the selected skill. To adaptively assess user inputs and dynamically optimize decisions, the system consists of mechanisms of model predictive control (Garcia et al. 1989). The state of the learner is estimated by the system and thus identified according to its internal representation: the student model. An attached bug library enables recognition of error patterns.

Student Model

The mathematical knowledge of the learner is modelled using a dynamic Bayesian network (Friedman et al. 1998). The network consists of a directed acyclic graphical model representing different mathematical skills and their dependencies. Two skills s_A and s_B have a (directed) connection if mastering skill s_A is a prerequisite for skill s_B. The belief of a skill s_{Ai} (probability that the skill is in the learnt state) is conditioned over its parents π_i (see Charniak (1991) for an introduction to Bayesian networks):

$$p(s_{A1}, ..., s_{An}) = \prod_i p_{s_{Ai}} \quad \text{where} \quad p_{s_{Ai}} := p(s_{Ai}|\pi_i) \tag{1}$$

As the skills cannot be directly observed, the system infers them by posing tasks and evaluating user actions. Such observations (E) indicate the presence of a skill probabilistically. The posteriors $p_{s_{Ai}|E_k}$ of the net are updated after each solved task k using the sum-product algorithm (libDAI Mooij 2010).

We initalize all probabilities to 0.5 as we do not have any knowledge about the mathematical proficiency of a learner at the beginning of the training (the students are of different age and have different mathematical skill levels).

This initalization is in accordance with the principle of maximum entropy. The dynamic Bayesian net has a memory of 5, i.e. posteriors are calculated over the last five time steps.

The skill net representation is ideal for modelling mathematical knowledge as the learning domain exhibits a distinctively hierarchical structure. The structure of the net was designed using experts' advice and incorporates domain knowledge. The design of the net was inspired by work from Falmagne et al. (1990). Like in knowledge space theory, we order skills hierarchically and assume that some skills can be surmised by others. If a child for example can compute additions involving a ten crossing, we assume that the child also knows addition without ten crossing. The basic assumption is that to know skill s_A, the child needs to know all the precursor skills of s_A. However, in our case, each skill is assigned to exactly one task. Our work can also be related to partial order knowledge structures (Desmarais et al. 1995) which also model dependencies between skills as conditional probabilities. Our resulting student model contains 100 different skills as illustrated in Fig. 2. Table 1 explains the different skills of the skill net and their notation used in Fig. 2.

The skills in *Part A* are ordered and colour-coded according to the different number ranges 0–10, 0–100, and 0–1000. Within each number range, the hierarchy follows the four-step developmental model (von Aster and Shalev 2007): The linguistic symbolization (step 2), arabic symbolization (step 3), and analogue magnitude representation (step 4) develop based on a (probably) inherited representation of cardinal magnitude of numbers (step 1). Following this model, the transcoding between the linguistic and arabic symbolization ($Verbal \rightarrow Arabic$) is trained before giving the position of a written number on a number line ($Arabic \rightarrow Numberline$).

Skills in *Part B* can also be divided into the number ranges 0–10, 0–100 and 0–1000 (colour-coded in Fig. 2). Furthermore, they are ordered according to their difficulties. The difficulty of a task depends not only on the magnitude of the numbers included in the task and the complexity of the task, but also on the representation of the task and the means allowed to solve it. A task such as '$65 + 22 = 87$' (*Addition 2,2*) is considered more difficult than computing '$13 + 5 = 18$' (*Addition 2,1*). On the other hand, modelling '$65 + 22 = 87$' with one, ten and hundred blocks (*Support Addition 2,2*) is easier than calculating it mentally. And finally, tasks including ten (or hundred) crossings such as '$65 + 27 = 92$' (*Addition 2,2 TC*) are more complex to solve than tasks without crossings.

In general, each skill of the hierarchical network is associated with a task, i.e., there exists a game type for each skill in the network. The PLUS-MINUS game (section "Training Effects") is for example associated with all addition and subtraction skills allowing the use of material (for example *Support Addition 2,2*). On the other hand, the LANDING game (section "Training Effects") is assigned to all skills involving the positioning of a number on a number line (for example $Arabic \rightarrow Numberline$).

Controller

The selection of actions is rule-based and non-linear. Rather than following a specified sequence to the goal, learning paths are adapted individually. Therefore, each child trains different skills and hence plays different games during training (Fig. 4). This increases the set of possible actions (due to multiple precursors and successors). After each solved task, the controller selects one of the following options based on the current state:

1. **Stay**: Continue the training of the current skill;
2. **Go back**: Train a precursor skill;
3. **Go forward**: Train a successor skill;

The decision is based on the posterior probabilities delivered by the student model. After each solved task, the controller fetches the posterior probability $p_{s|E}(t)$ of the skill s being trained at time t. Then, $p_{s|E}(t)$ is compared against a lower and an upper threshold, denoted by $p_s^l(t)$ and $p_s^u(t)$. The resulting interval defines the optimal training level: if the probability lies between the thresholds, 'Stay' is selected. In contrast, 'Go Back' and 'Go forward' are selected if $p_{s|E}(t) < p_s^l(t)$ and if $p_{s|E}(t) > p_s^u(t)$, respectively. Thresholds are not fixed: they converge with more played samples (n_c):

$$p_s^l(t) = p_s^{l0}(t) \cdot {l_c}^{n_c} \quad \text{and} \quad p_s^u(t) = p_s^{u0}(t) \cdot {u_c}^{n_c} \tag{2}$$

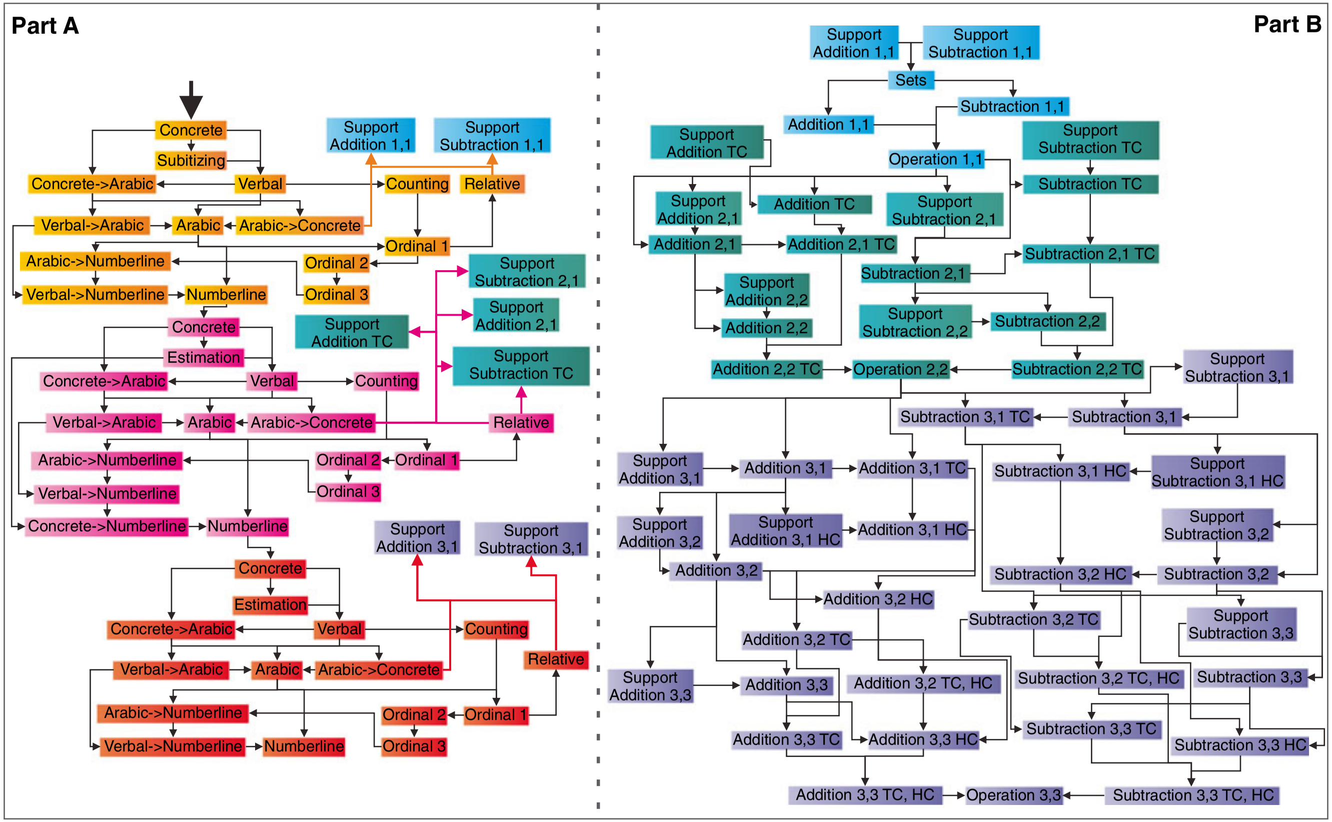


Fig. 2 Skill net containing the skills of part A (*left*) and part B (*right*). The different colours denote the different number ranges 0–10 (*yellow*, *blue*), 0–100 (*pink*, *green*), and 0–1000 (*red*, *purple*). Some skills from part B have been duplicated to part A to denote the connections between the two parts

Table 1 Explanation of skills and notations used in the skill net

Area	Notation	Definition
Part A		
Number representations	Concrete	Number represented as a set of objects.
	Verbal	Spoken number.
	Arabic	Written number.
	Numberline	Number represented as a position on a number line.
Transcoding	r1→r2	Translation of number from number representation r1 to r2.
Ordinality	Ordinal 1	Predecessor and successor of a number need to be given.
	Relative	Calculate indirect (+/-2, +/-3) predecessors and successors of a given number.
	Ordinal 2	Judge, if the given numbers are sorted in ascending order.
	Ordinal 3	Guess a secret number.
Other	Subitizing	Simultaneous perception of numbers from 1-4.
	Estimation	Which of three displayed point sets corresponds to the given number?
	Counting	Forwards (and backwards) counting in the according number range.
Part B		
Mental calculation	Addition a1,a2	Addition of two numbers. a1 and a2 denote the number of digits of the addends. TC denotes a ten crossing and HC a hundred crossing.
	Subtraction s1,s2	Subtraction of two numbers. s1 and s2 denote the number of digits of the minuend and the subtrahend. TC denotes a ten crossing and HC a hundred crossing.
	Addition TC	Addition with bridging to ten in the range from 0–20.
	Subtraction TC	Subtraction with bridging to ten in the range from 0–20.
	Operation o1,o2	Addition or subtraction of two numbers used as a repetition of the whole number range. o1 and o2 denote the number of digits of the operation. Operation 2,2 for example denotes any addition or subtraction skill in the number range 0–100.
Calculation concepts	Support Addition	Addition of two numbers. The task can be solved using one, ten and hundred blocks.
	Support Subtraction	Subtraction of two numbers. The task can be solved using one, ten and hundred blocks.
	Sets	Understanding of operations on sets.

Initial values of the upper $\left(p_s^{l0}(t)\right)$ and lower $\left(p_s^{u0}(t)\right)$ thresholds as well as the change rates (l_c, u_c) are heuristically determined. The convergence of the thresholds ensures a sufficiently large number of solved tasks per skill and prevents training the same skill for too long without passing it.

When 'Stay' is selected, a new appropriate task is built. Otherwise, a precursor (or successor) skill is selected by fetching all precursor (successor) skills of the current skill and feeding them into a decision tree. Figure 3 shows the simplified decision trees for 'Go Back' and 'Go Forward'. The nodes of the trees encode selection rules that were designed using experts' advice.

For the 'Go Back' option, remediation skills are preferred: If error matching patterns of the bug library are detected, the relevant remediation skill is trained. A typical mistake in addition involving two digit numbers would be to sum up all the digits, i.e. '$23 + 12 = 8$' (skill *Addition 2,2* in Fig. 2). This mistake indicates that the child has not yet understood the Arabic notation system in the number range from 0–100. A remediation skill for this error is the training of the Arabic notation system in this range, i.e. decomposing numbers between 0 and 100 into tens and units and thus learning the meaning of the digit position of a number (skill $Arabic- > Concrete$ in Fig. 2). If the child did not commit any of the typical errors, the controller prefers unplayed precursor skills. The hierarchical skill model assumes that the precursor skills of a skill s are a prerequisite for knowing s. If the child fails that skill s, the controller tries to find the particular precursor skill that might cause the problem. For the played precursor skills, the controller assumes that the child already knows them (since they have been played and passed) and hence an unplayed precursor skill is selected. Finally, main skills are preferred over support skills. Main skills require a combination of abilities to solve them, while support skills train specific abilities and serve as basic prerequisites. In arithmetic operations, main skills involve mental calculation, while support skills involve the use of material (unit, ten and hundred blocks) to solve the task. Therefore, if a child fails in solving addition problems with two-digit numbers (for example '$23 + 12 =?$') the controller first checks if the child can do mental calculation (= main skill) of simpler addition problems (for example '$23+2 =?$'). If this is the case, the support skill modelling the operation with material can be picked. If however the child also fails in solving the simpler addition problem, this easier skill needs to be trained first. Hence, the main skills are always checked first. If there is more than one candidate precursor skill after crossing the decision tree (i.e. going through all the rules), the candidate skill with the lowest posterior probability is selected. Therefore, the controller selects the skill where the child has the lowest proficiency.

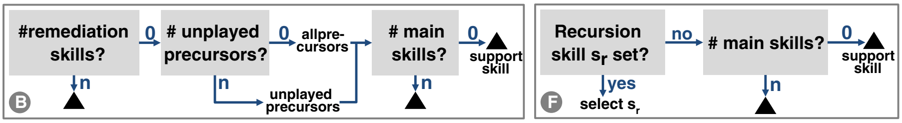


Fig. 3 Decision trees for 'Go Back' (*left*) and 'Go Forward' (*right*) options. At the end nodes (*triangles*), the candidate skill with lowest posterior probability ('Go Back' option)/with posterior probability closest to 0.5 ('Go Forward' option) is selected

Fig. 4 Skill sequences of three children in addition. The notation is consistent with Fig. 2. User 2 and 3 passed all skills in the range, while user 1 did not pass this range within the training period. The length of the rectangles indicates the number of samples

For the 'Go Forward' option, recursion skills are preferred. If a user fails to master skill s_A and goes back to s_B, s_A is set as a recursion skill. After passing s_B, the controller will return to s_A. If a child for example fails solving addition problems with two-digit numbers (for example '$23 + 12 = ?$') and goes back to train an easier skill (for example '$23 + 2 = ?$'), the child will go back to the addition problems with two-digit numbers after passing that easier skill. If no recursion skill is set, the controller again prefers main skills over support skills. If the child masters solving addition problems with two-digits (for example '$23 + 12 = ?$') the controller will go further to ask addition problems involving a ten crossing (for example '$23 + 18 = ?$'). This rule ensures that children having a good mathematical knowledge take the fastest way through the skill net. The support skill modelling the task '$23+18 = ?$' using material will only be played if the child does not master the mental calculation. If there is more than one candidate successor skill at the end of the decision tree, the candidate skill with posterior probability closest to 0.5 (maximization of entropy) is selected. This final rule ensures that the gain of knowledge about the child is maximized.

To consolidate less sophisticated skills and to increase variability, the controller uses selective recalls. This control design exhibits the following advantages:

1. *Adaptability*: the network path targets the needs of the individual user (Fig. 4).
2. *Memory modelling*: forgetting and knowledge gaps are addressed by going back.
3. *Locality*: the controller acts upon current nodes and neighbours, avoiding unreliable estimates of far nodes.
4. *Generality*: the controller is domain model-independent: it can be used on arbitrary discrete structures.

Experimental Setup

To measure the quality of the controller and the student model, the training program was assessed in two user studies. All the analyses performed are based on external effectiveness tests and input data from participants of these two studies.

Study Design and Participants

Experimental data stem from 63 participants (45 females, 18 males) of two ongoing large-scale studies (Germany and Switzerland). Participants were divided into a training group (n $=$ 33, $66.\bar{6}$ % females) completing a 6-weeks training and a waiting group (n $=$ 30, $76.\bar{6}$ % females) starting with a 6-weeks rest period, followed by a 6-weeks training. The groups were matched according to age (training group: $M = 9.26$ years (SD 0.94), waiting group: $M = 9.39$ years (SD 1.09), $t(61) = -0.49$, $p = .63$) and intelligence (training group CFT/BUEGA-score: 101.09 (SD 11.38), waiting group CFT/BUEGA-score: 100.13 (SD 10.74), $t(61) = 0.34$, $p = .73$) (Cattell et al. 1997; Esser et al. 2008). All participants attended the 2.–5. grade of public elementary schools and were German-speaking. Mathematical performance of the participants was evaluated at the beginning of the study (t_1), after 6 weeks (t_2) and after 12 weeks (t_3). The children exhibited difficulties in learning mathematics indicated by a below-average performance in the standardized arithmetic test **HRT** (addition T-score: 34.14 (SD 6.71), subtraction T-score 33.76 (SD 7.36)) (Haffner et al. 2005). At the beginning of the study (t_1), there was no significant difference in arithmetic performance between the training and waiting group (addition: $t(61) = -0.25$, $p = .80$, subtraction: $t(61) = -1.30$, $p = .20$). The participants were required to train with the program for a period of 6 weeks with a frequency of five times per week, during sessions of 20 min. For the present analyses, only children with at least 24 complete training sessions were included.

External Instruments

Training effects were measured using paper-pencil and computer-based mathematical performance tests. On the one hand, arithmetic performance was assessed using the addition and subtraction subtests of the **HRT** (re-test reliability: addition $r_{tt} = .82$, subtraction $r_{tt} = .86$). In these subtests, children are provided with a list of addition (subtraction) tasks ordered by difficulty. The goal of the test is to solve as many tasks as possible within a time frame of 2 min. Thus, the **HRT** measures speed. On the other hand, arithmetic performance was also measured with the **AC** (arithmetic test), which exists in a paper-pencil and a computer-based version. In this test, children solve a series of addition (and subtraction) tasks ordered by difficulty. Tasks were presented serially in a time frame of 10 min. In contrast to the **HRT**, the **AC** also contains more complex tasks in the number range from 0–100.

Input Data

Experimental data consisted of input logs recorded from 63 participants and containing six weeks of training (training group: $t_1 - t_2$, waiting group: $t_2 - t_3$). On average, each user completed 29.77 (SD 2.43, min 24, max 36) sessions. The total number of solved tasks was 1540 (SD 276, min 1011, max 2179), while the number of solved tasks per session corresponded to 51.70 (SD 7.86, min 37.63, max 75.1).

To facilitate the analysis of the log files, the concept of ‘key skills’ is introduced. Key skills are defined in terms of subject-dependent difficulty, they are the hardest skills for the user to pass. More formally,

Definition 1 A skill s_A is a **key skill** for a user U, that is $s_A \in \mathcal{K}_U$, if the user went back to a precursor skill s_B at least once before passing s_A.

From this follows that the set of key skills $\mathcal{K}_U$ may be different for each user U (and it typically is). In the sequence in Fig. 4, user 2 has no key skills, while user 3 has one key skill (coloured in green) and user 1 has several key skills.

Results & Discussion

The analyses performed on the input data and the external effectiveness measures assess the quality of the training program and in particular the quality of the student model and the controller mechanism according to different criteria:

1. Efficacy of training program: We show that the participants improved over the course of the training. This improvement is demonstrated by an increased mathematical performance within the system (section “System-internal Improvement Analysis”). Furthermore, we also include first pilot results of external arithmetic tests (section “Training Effects”).
2. Assessment of controller design: We show that the introduced control mechanism significantly speeds up learning (section “Controller Design”).
3. Adaptability: We show that the program rapidly adapts to the knowledge level of the user (section “Controller Adaptability”).

However, the analyses of the logfiles are not only useful to assess the quality of the training program, but also to understand properties of the users and the skills of the student model. We analyse the performance of the users in the program as well as properties of skills (section “Key Skills”). Such analyses can lead to a better understanding of the mathematical knowledge of the users.

Training Effects

A repeated measures general linear model (GLM) analysis was conducted to evaluate training effects ($t_1 - t_2$) as a within-subject factor and group (Training/Waiting) as a between-subject factor. Parametric t-tests were used to calculate differences between measurement points (paired-sample t-test, $t_1 - t_2$, $t_2 - t_3$). Table 2 summarizes the means and standard deviations of the mathematical performance measures for all measurement points, including calculated statistical results. There were no between-group performance differences prior to the intervention. The training induced a significant improvement in subtraction, while no improvement was found after the waiting period (HRT interaction: $p < .001$, AC interaction: $p < .001$). Children also improved significantly in addition, however no significant difference between the training and the waiting group was found (HRT interaction: $p = .18$, AC interaction:

Table 2 Mathematical performance of training and waiting group over the course of the study: Mean (SD) test scores (number of correctly solved tasks) for t_1, t_2 and t_3. Interaction between training and group (F-score) as well as differences between measurement points (t-score)

	t_1	t_2	t-score ($t_1 - t_2$)	F-score ($t_1 - t_2$)	t3	t-score ($t_2 - t_3$)
HRT Add.						
TG	15.64 (5.22)	18.36 (5.31)	5.20***	1.82	–	–
WG	16.53 (6.10)	18.23 (6.00)	3.08**		19.37 (5.74)	1.97
HRT Sub.						
TG	12.06 (5.27)	16.15 (5.17)	8.36***	15.71***	–	–
WG	14.00 (6.65)	14.63 (6.25)	0.86		17.33 (6.04)	4.84***
AC Add.						
TG	68.58 (25.82)	77.22 (24.73)	3.15**	1.99	–	–
WG	67.83 (29.79)	69.94 (27.83)	0.55		73.60 (20.92)	1.41
AC Sub.						
TG	50.91 (26.12)	63.13 (26.98)	5.40***	14.39***	–	–
WG	53.54 (25.29)	53.21 (27.19)	0.14		65.38 (23.26)	4.22***

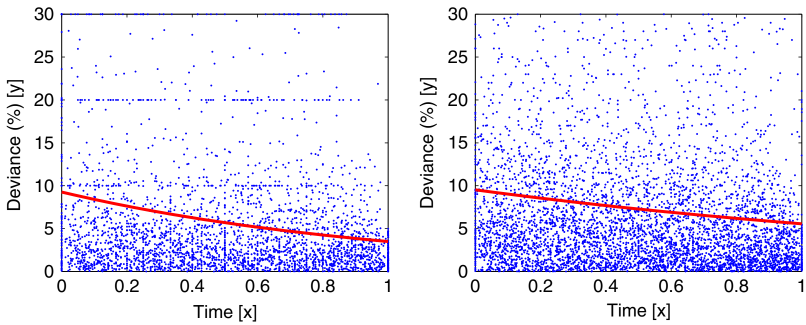
*$p < .05$, **$p < .01$, ***$p < .001$

$p = .16$). Surprisingly, also the waiting group improved significantly in the HRT addition test, this effect can however be attributed to outliers (one child probably not understanding the test correctly at t_1 and therefore solving only three tasks within the two minutes). Removing the outlier leads to a significant interaction ($p = .018$).

The improvement in subtraction is supported by additional evidence: The percentage of training time children spent with subtraction tasks. In fact, 62.5 % (78.4 % if considering key skills only) of arithmetical tasks consist of subtractions. As children had especially difficulties in subtractions with ten crossings (section "Key Skills") the improvement might stem from a better understanding of subtracting with carry or from higher automation and thus lower working memory load. However, subtraction is also considered the main indicator for numerical understanding (Dehaene 2011). Consistently with this, improved number line representation is directly measurable from the recorded input data. Input data used in the analysis consists of all landing tasks (denoted as samples with index i) solved by each child. Each sample can be characterized by a variable x_i which denotes the index of the sample, i.e. the normalized measurement point of that sample ($x_i \in [0, 1]$), and a dependent variable y_i denoting the deviance from the correct position. The analysis of the accuracy was performed using a non-linear mixed effect model (NLME) (Pinheiro and Bates 1995):

$$y_i \sim \text{Poisson}(\lambda_i) \text{ with } \lambda_i = e^{b_0 + b_1 \cdot x_i + u_i} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \quad (3)$$

where u_i denotes the noise term. Fitting was performed using one group per user. Over time, children achieved greater accuracy when giving the position of a number on a number line (Fig. 5, top). The intercept (coefficient b_0) was significant only for the number range 0-100, indicating that children started at different accuracy levels



Number range	Coefficient	Estimate (SD)	sig.	95% ci
0-10	b_0	-0.08 (0.06)	0.21	[-0.20 0.04]
	b_1	-0.97 (0.07)	$<$ 1e-4	[-1.11 -0.84]
0-100	b_0	2.25 (0.04)	$<$ 1e-4	[2.17 2.34]
	b_1	-0.54 (0.02)	$<$ 1e-4	[-0.57 -0.51]

Fig. 5 Landing accuracy in the number range 0–10 (*left*) and 0–100 (*right*) increases over time. The x-axis denotes the normalized sample indices (Time[x]), while the y-axis displays the deviance from the correct position. Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci)

in the number range 0-10. The significance of b_1 in both number ranges demonstrates the significant improvement in accuracy (Fig. 5, bottom).

The lines of points (at 10 %, 20 % and 30 % in Fig. 5 (left)) arise from the nature of the LANDING game (Fig. 1a): The children need to indicate the position of a given number by steering a falling cone with the joystick. If nothing is done, the cone will always land at the position of the five (in the number range 0–10), which leads to deviations of exactly 10 % (if the given number was 4 or 6), 20 % (if the given number was 3 or 7) or 30 % (if the given number was 2 or 8).

The improvement measured in subtraction, number representation and partly addition is promising and builds the basis for further extensions of the training environment and control structure. One possible future feature could be the incorporation of answer times into the task assessment. In the number range from 0-10, fact retrieval is very important and can only be tested by taking answer times into account. Another addition could be to teach and assess the strategies used to perform arithmetic operations.

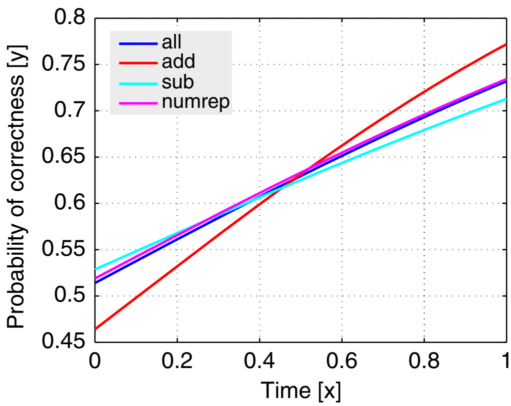
System-internal Improvement Analysis

To quantify improvement, the learning rate over $\mathcal{K}_U$ was measured from all available samples (both if the participant mastered them during training or not). One sample x_i denotes exactly one task of one user. The variable y_i denotes the result (correct or wrong solution) of that task. Therefore, the analysis includes all the tasks that the children solved at their respective key skills. The improvement over time $I([t_{\mathcal{K}_U}, t_{\text{end}}])$ was computed using a NLME model employing one group per user and key skill:

$$y_i \sim \text{Binomial}(1, p_i) \text{ with } p_i = \frac{1}{1 + e^{-(b_0 + b_1 \cdot x_i + u_i)}} \text{ and } u_i \sim \mathcal{N}(0, \sigma^2) \quad (4)$$

where u_i denotes the noise term. The sample indices x_i have been normalized ($x_i \in [0, 1]$).

The resulting model (Fig. 6) for all skills exhibits an estimated mean improvement of 21.8 % (95 % confidence interval = [0.21 0.23]). Interestingly, subtraction exhibits a lower improvement than addition. Given the external training effects (section "Training Effects"), we would expect the opposite. However, children have a lot more subtraction key skills than addition key skills (section "Key Skills"). Therefore, despite the average improvement per skill being higher for addition, the total



Category	Coefficient	Estimate (SD)	sig.	95% ci
All	b_0	0.06 (0.05)	0.22	[-0.03 0.14]
	b_1	0.95 (0.04)	$<$ 1e-4	[0.87 1.03]
Addition	b_0	-0.14 (0.09)	0.13	[-0.33 0.04]
	b_1	1.36 (0.11)	$<$ 1e-4	[1.14 1.59]
Subtraction	b_0	0.11 (0.08)	0.15	[-0.04 0.27]
	b_1	0.79 (0.07)	$<$ 1e-4	[0.66 0.93]
Number representation	b_0	0.08 (0.07)	0.26	[-0.06 0.21]
	b_1	0.94 (0.06)	$<$ 1e-4	[0.82 1.06]

Fig. 6 The percentage of correctly solved tasks (of key skills) increases over the training period by 21.8 % for all skills (*top*). The normalized sample indices x_i (Time[x]) are displayed on the x-axis, while the y-axis shows the ratio of correct solutions. Improvements for addition (add), subtraction (sub) and number representation (numrep) are in the same range. Exact coefficients of NLME along with standard deviation (in brackets) are plotted by respective significance (sig.) and confidence intervals (ci) (*bottom*)

improvement is still higher for subtraction. Furthermore, the higher number of key skills in subtraction leads to more practice in subtraction skills. The conducted analysis measures the learning progress within the system and demonstrates that children managed to improve their abilities in areas that were difficult for them.

Key Skills

Although the key skills vary a lot over the children, some skills seem to be difficult for most of the children and thus more likely to be key skills: Nine skills were key skills for more than one third of the children. Of these skills, five were subtraction skills, four number representation skills and one an addition skill. Even more than 50 % of the children had problems with the top three key skills: Indicating the position of a number on a number line from 0-100 ($Arabic \rightarrow Numberline$ in Fig. 2) was difficult for 52 % of the children. This result is in line with previous work, which observed deficits of mental number representation in children with DD (Kucian et al. 2006; Mussolin et al. 2010; Price et al. 2007). More than 50 % of the children also had problems in subtraction in the number range from 0–100, when a ten crossing was involved (Subtraction 2,1 TC and Subtraction 2,2 TC in Fig. 2). This result again confirms the link between subtraction and spatial number representation (Dehaene 2011).

The mathematical performance of the users, i.e. their mathematical knowledge can also be assessed by their number of key skills. The normalized number of key skills is computed as the number of key skills divided by the number of totally played skills. On average, the normalized number of key skills per user was 0.27 (SD 0.14). This number can be interpreted as follows: On average, the children had difficulties with 27 % of the skills that they played. When breaking this number down into the different categories (number representation, addition and subtraction) it can be seen that most problems arose in subtraction. The normalized number of key skills was 0.26 (SD 0.19) in number representation, 0.17 (SD 0.2) in addition and 0.37 (SD 0.15) in subtraction. The distribution over the normalized key skill numbers in the different categories are displayed in Fig. 7. Interestingly, we observe that the normalized key skill numbers in addition and number representation skills follow an exponential distribution. The long tail of the distribution demonstrates that most children did

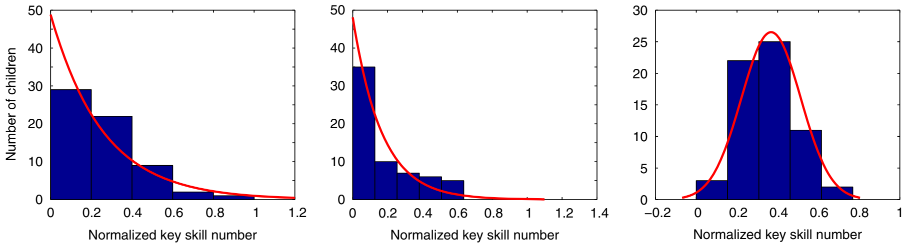


Fig. 7 Distribution over normalized number of key skills for number representation skills (*left*), addition skills (*center*) and subtraction skills (*right*)

not have difficulties in these categories. Rather, only few children had strong difficulties in these categories. On the other hand, the normalized number of key skills in subtraction is significantly higher than in the two other categories (indicated by a two-sided t-test: $p < .001$ for both categories). The key skill analysis once more also shows the heterogeneity of the children: The number of key skills as well as the key skill set itself varied a lot over the children. Nevertheless, we can observe an accumulation in subtraction. Having more input data available, a more detailed analysis of key skills will be conducted in future work.

Controller Design

Further analysis demonstrates that the possibility to go back to easier (played or unplayed) skills yields a significant beneficial effect. The user not only immediately starts reducing the rate of mistakes, but also learns faster. The log files recorded 973 individual cases of going back. On average, 20.6 cases (SD 12.1) of going back are recorded per user. From Fig. 8 it can be seen that the number of going back cases varies a lot among the users, i.e. the users exhibit very different levels of mathematical knowledge. All cases in which users play a certain skill (samples x_b), go back to one or several easier skills, and finally pass them to come back to the current skill (samples x_a) are incorporated in the analysis. The variable x_b therefore denotes all tasks before going back, while x_a stands for the tasks solved after going back. Per each case k the correct rate over time $c_{a,k}$ ($c_{b,k}$) is estimated separately for x_a and x_b. Fitting is performed via logistic regression using bootstrap aggregation (Breiman 1996) with resampling ($B = 200$). The direct improvement d_k is the difference between the initial correct rate $c_{a,k}$ (at $x_a = 0$) and the achieved correct rate $c_{b,k}$ (at $x_b = 1$). The improvement in learning rate r_k is the difference in learning rate over $c_{a,k}$ and $c_{b,k}$. The distributions over $\bar{d}$ (mean over d_k) and $\bar{r}$ (mean over r_k) are well approximated by a normal distribution (Fig. 9 top) with means greater than 0. The rate of correct tasks $\bar{d}$ is increased by 0.14 while the learning rate $\bar{r}$ is even increased by 0.36 after going back. Both measurements are positive on average and a two-sided t-test indicates their statistically significant difference from 0 (Fig. 9

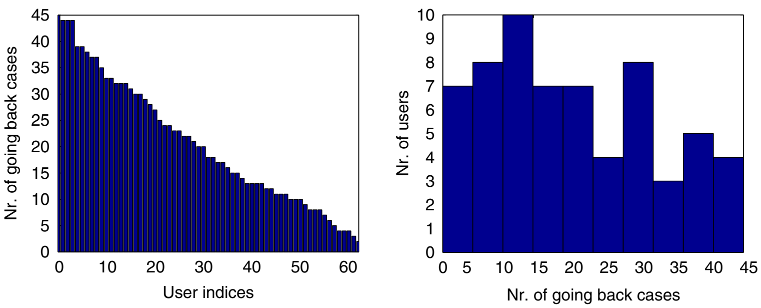
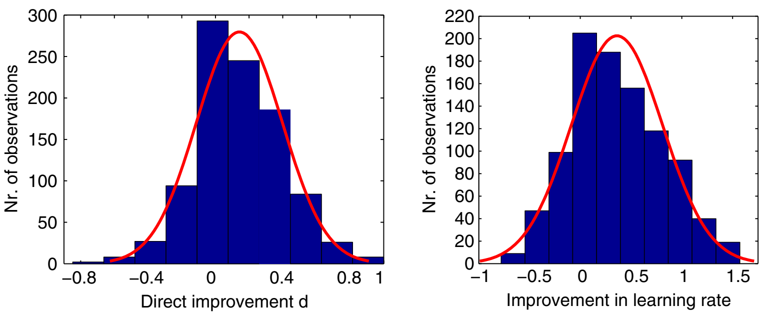


Fig. 8 Number of going back times per user sorted in ascending order (*left*) and distribution over number of going back cases (*right*). The equal distribution of going back numbers demonstrates the heterogeneity of mathematical knowledge of the children



	Mean μ	sig.	99% ci of μ	SD σ	99% ci of σ
$\bar{\mathbf{d}}$	0.1411	<1e-3	[0.1248 0.1574]	0.2570	[0.2459 0.2690]
$\bar{\mathbf{r}}$	0.3618	<1e-3	[0.3330 0.3907]	0.4543	[0.4348 0.4756]

Fig. 9 Distributions over direct improvement $\bar{d}$ (*top left*) and improvement in learning rate $\bar{r}$ (*top right*). Statistics for the improvement after going back (*bottom*): Mean improvement μ, significance of mean (sig.), standard deviation (SD), and confidence intervals (ci)

bottom). To summarize, children reduce the rate of mistakes immediately after going back (demonstrated by the significantly positive $\bar{d}$) and exhibit a higher learning rate (demonstrated by the significantly positive $\bar{r}$).

Controller Adaptability

During the study, all participants started the training at the lowest (easiest) skill of the net. The adaptation time $[t_0, t_{\mathcal{K}_U}]$ is defined as the period between the start t_0 of the training and the first time the user hits one of his key skills $t_{\mathcal{K}_U}$. On average, the participants reached their $t_{\mathcal{K}_U}$ after solving 148.3 tasks (SD 122.6, min 17, max 534). The number of complete sessions played up to this point was 2.1 (SD 1.97, min 0.2, max 10.92). These results show that the model rapidly adjusts to the state of knowledge of the user. The fast adaptability is also confirmed by the fact that 52.4 % of the children hit their first key skill already in the number range 0–10, 38.1 % of children in the number range 0–100 and only 9.5 % of the children in the number range 0–1000. The fast adaptation to the the child's knowledge ensures that each child trains at the optimal difficulty level already after a few days of training.

Conclusion

This study presents a model of the cognitive processes of mathematical development and an automatic control algorithm acting on it. The student model is represented by a dynamic Bayesian network which incorporates domain knowledge. The introduced control algorithm is decision-based and enables the optimization of the learning process through targeted cognitive stimulation. The reported data demonstrate a significant increase in mathematical performance, measured by external effectiveness tests as well as from input logs. The large-scale input data analysis also proved the efficiency and adaptability of the student model and the control algorithm. In particular, the possibility to go back to easier skills significantly (and rapidly) reduces the error rate and yields an overall increased learning rate. The student model has the potential to be further refined by incorporating additional available experimental data.

Acknowledgments The present work was funded by the CTI-grant 11006.1 and the BMBF-grant 01GJ1011.

References

Badian, N.A. (1983). Arithmetic and nonverbal learning. In H.R. Myklebust (Ed.), *Progress in learning disabilities* (pp. 235–264).

Baschera, G.M., & Gross, M. (2010). Poisson-based inference for perturbation models in adaptive spelling training. *International Journal of AIED*, *20*(4), 333–360.

Baschera, G.M., Busetto, A.G., Klingler, S., Buhmann, J., Gross, M. (2011). Modeling engagement dynamics in spelling learning. In *Proceedings AIED* (pp. 31–38).

Breiman, L. (1996). *Bagging predictors* (pp. 123–140).

Busetto, A.G., & Buhmann, J.M. (2009). Structure identification by optimized interventions. *Journal of Machine Learning Research*, *5*, 57–64.

Butterworth, B. (2005a). The development of arithmetical abilities. *Journal of Child Psycholgy and Psychiatry*, *46*, 3–18.

Butterworth, B. (2005b). Developmental dyscalculia. In J.D. Campell (Ed.), *The handbook of mathematical cognition* (pp. 455–469).

Cattell, R., Weiss, R., Osterland, J. (1997). *Grundintelligenztest Skala 1 (CFT)*. Göttingen: Hogrefe Verlag.

Charniak, E. (1991). Bayesian networks without tears. *AI Magazine*, *12*(4), 50–63.

Cohen Kadosh, R., Cohen Kadosh, K., Schuhmann, T., Kaas, A., Goebel, R., Henik, A., Sack, A.T. (2007). Virtual dyscalculia induced by parietal-lobe TMs impairs automatic magnitude processing. *Current Biology*, *17*, 689–693.

Corbett, A.T., & Anderson, J.R. (1994). Knowledge tracing: modeling the acquisition of procedural knowledge. *UMUAI*, *4*, 253–278.

Dehaene, S. (1995). Varieties of numerical abilities. *Cognition*, *44*, 1–42.

Dehaene, S. (2011). *The number sense: how the mind creates mathematics*. New York: Oxford University Press.

Desmarais, M.C., Maluf, A., Liu, J. (1995). User-expertise modeling with empirically derived probabilistic implication networks. *User Modeling and User-Adapted Interaction*, *5*(3–4), 283–315.

Dowker, A. (2001). Intervention in numeracy: the development of a numeracy recovery project for young children with arithmetical difficulties. *Support for Learning*, *16*, 6–10.

Esser, G., Wyschkon, A., Ballaschk, K. (2008). *BUEGA: Basisdiagnostik Umschriebener Entwicklungsstörungen im Grundschulalter*. Göttingen: Hogrefe Verlag.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., Johannesen, L. (1990). Introduction to knowledge spaces: how to build, test, and search them. *Psycholgical Review*, *197*(2), 201–224.

Friedman, N., Murphy, K., Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Proceedings UAI* (pp. 139–147).

Garcia, C.E., Prett, D.M., Morari, M. (1989). Model predictive control: theory and practice—a survey. *Automatica*, *25*(3), 335–348.

Geary, D., Bow-Thomas, C., Yao, Y. (1992). Counting knowledge and skill in cognitive addition: a comparison of normal and mathematically disabled children. *Journal of Experimental Child Psychology*, *54*, 372–391.

Griffin, S., Case, R., Siegler, R. (1994). Rightstart: providing the central conceptual prerequisites for the first formal learning of arithmetic to students at risk for school failure. In K. McGilly (Ed.)*, Classroom lessons: integrating cognitive theory and classroom practice*. Cambridge: MIT Press.

Haffner, J., Baro, K., Parzer, P., Resch, F. (2005). *Erfassung mathematischer Basiskomptenzen im Grundschulalter*. Göttingen: Hogrefe Verlag.

Käser, T., Busetto, A.G., Baschera, G.-M., Kohn, J., Kucian, K., von Aster, M., Gross, M. (2012). Modelling and optimizing the process of learning mathematics. In *Proceedings ITS* (pp. 389–398).

Kaufmann, L., Handl, P., Thöny, B. (2003). Evaluation of a numeracy intervention program focusing on basic numerical knowledge and conceptual knowledge: a pilot study. *Journal of Learning Disabilities*, *35*, 564–573.

Koedinger, K.R., Anderson, J.R., Hadley, W.H., Mark, M.A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of AIED*, *8*(1), 30–43.

Kucian, K., Grond, U., Rotzer, S., Henzi, B., Schönmann, C., Plangger, F., Gälli, M., Martin, E., von Aster, M. (2011). Mental number line training in children with developmental dyscalculia. *NeuroImage*, *57*(3), 782–795.

Kucian, K., & Kaufmann, L. (2009). A developmental model of number representation. *Behavioral and Brain Sciences*, *32*, 313–373.

Kucian, K., Loenneker, T., Dietrich, T., Dosch, M., Martin, E., von Aster, M. (2006). Impaired neural networks for approximate calculation in dyscalculic children: a functional MRI study. *Behavioral and Brain Functions*, *2*, 31.

Landerl, K., Bevan, A., Butterworth, B. (2004). Developmental dyscalculia and basic numerical capacities: a study of 8-9-year-old students. *Cognition*, *93*, 99–125.

Lenhard, A., Lenhard, W., Schug, M., Kowalski, A. (2011). Computerbasierte Mathematikförderung mit den Rechenspielen mit Elfe und Mathis I. Vorstellung und Evaluation eines Computerprogramms für Erst- bis Drittklässler. *Zeitschrift fr Entwicklungspsychologie und Pädagogische Psychologie*, *43*(2), 79–88.

Lewis, C., Hitch, G.J., Walker, P. (1994). The prevalence of specific arithmetic difficulties and specific reading difficulties in 9- to 10-year old boys and girls. *Journal of Child Psychology and Psychiatry*, *35*, 283–292.

Mislevy, R.J., Almond, R.G., Yan, D., Steinberg, L.S. (1999). Bayes nets in educational assessment: where the numbers come from. In *Proceedings UAI* (p. 518).

Mooij, J.M. (2010). libDAI: a free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, *11*, 2169–2173.

Mussolin, C., De Volder, A., Grandin, C., Schlogel, X., Nassogne, M.C., Noel, M. (2010). Neural correlates of symbolic number comparison in developmental dyscalculia. *Journal of Cognitive Neuroscience*, *22*(5), 860–874.

Ostad, S.A. (1997). Developmental differences in addition strategies: a comparison of mathematically disabled and mathematically normal children. *British Journal of Education Psychology*, *67*, 345–357.

Ostad, S.A. (1999). Developmental progression of subtraction strategies: a comparison of mathematically normal and mathematically disabled children. *European Journal of Special Needs Education*, *14*, 21–36.

Pavlik, P.I., Cen, H., Koedinger, K.R. (2009a). Learning factors transfer analysis: using learning curve analysis to automatically generate domain models. In *Proceedings EDM* (pp. 121–130).

Pavlik, P.I., Cen, H., Koedinger, K.R. (2009b). Performance factors analysis—a new alternative to knowledge tracing. In *Proceedings AIED* (pp. 531–538).

Pinheiro, J.C., & Bates, D.M. (1995). Approximations to the loglikelihood function in the nonlinear mixed effects model. *Journal of Computational and Graphical Statistics*, *4*, 12–35.

Price, G.R., Holloway, I., Räsänen, P., Vesterinen, M., Ansari, D. (2007). Impaired parietal magnitude processing in developmental dyscalculia. *Current Biology*, *17*, 1012–1013.

Rau, M.A., Aleven, V., Rummel, N. (2009). Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In *Proceedings AIED* (pp. 441–448).

Rubinsten, O., & Henik, A. (2005). Automatic activation of internal magnitudes: a study of developmental dyscalculia. *Neuropsychology*, *19*, 641–648.

Shalev, R., & von Aster, M.G. (2008). Identification, classification, and prevalence of developmental dyscalculia. *Enc. of Language and Literacy Development*, 1–9.

Shalev, R.S. (2004). Developmental dyscalculia: review. *Journal of Child Neurology*, *19*, 765–771.

Van De Rijt, B.A.M., & Van Luit, J.E.H. (1998). Effectiveness of the additional early mathematics program for teaching children early mathematics. *Instructional Science*, *26*(5), 337–358.

von Aster, M.G., & Shalev, R. (2007). Number development and developmental dyscalculia. *Developmental Medicine and Child Neurology*, *49*, 868–873.

Wilson, A.J., Dehaene, S., Pinel, P., Revkin, S.K., Cohen, L., Cohen, D. (2006). Principles underlying the design of The Number Race, an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions*, *2*, 19.

Wright, R.J. (2003). A mathematics recovery: program of intervention in early number learning. *Australian Journal of Learning Disabilities*, *8*(4), 6–11.