IADIS International Journal on WWW/Internet
Vol. 11, No. 3, pp. 1-12
ISSN: 1645-7641

TEACHERS LITTLE HELPER: MULTI-MATH-COACH

Martin Ebner. *Social Learning, Graz University of Technology, Münzgrabenstraße 35a, 8010 Graz*

Martin Schön. *Social Learning, Graz University of Technology, Münzgrabenstraße 35a, 8010 Graz*

Behnam Taraghi. *Social Learning, Graz University of Technology, Münzgrabenstraße 35a, 8010 Graz*

Michael Steyrer. *Social Learning, Graz University of Technology, Münzgrabenstraße 35a, 8010 Graz*

ABSTRACT

Individual learning is out of sync with the elements of a curricula and the daily program of a teacher. At a time when multidigit multiplication methods are taught, many children are not perfectly performing the basic multiplication table. Teachers organize settings for learning and they usually have no time to give an individual feedback to every single student. Furthermore they have no physiological capacity to produce statistics and to investigate the probable causes of errors. Our goal is to give teachers a tool to help them to review and train some typical requirements for written multiplying. We use the ubiquity of the Internet and provide on different devices free applications with written multiplication problems. All data are collected centrally and are used to determine typical errors. These data could also be used to answer further new questions and test new hypothesis. This is one aspect of Learning Analytics. The given tasks are automatically adapted to the skill level of a student. Teachers can see every detail of the solution process of every task. But really useful is that teachers can see the typical problems of each student as well as of the whole class every time at a glance. They need just a connection to the database with a web browser – anywhere, anytime.
Finally the first evaluation of the system pointed out some remarkable results. Children are learning with the system in the intended way according to the shown learning curves.

KEYWORDS

Written multiplication, feedback, learning analytics, degree of competence, algorithm, school children

1. INTRODUCTION

Even before the advent of the first personal computers (PC) in classrooms, educational researchers attempted to realize an intelligent tutoring system (ITS) [13]. Early attempts to model student knowledge were based on a "failure" model for subtraction with more than 100 typical problems described by Brown and Vanlehn [2]. This is based on the idea that computers can “perceive” and evaluate much faster and more detailed information about some aspects of the behavior and the learning process of each individual student in a class as a teacher. Even teachers have always too little time. Therefore in the past the extensive listings of results and protocols have been ignored or rejected by the teachers. The PC with restricted capacity brought some simple drill & practice programs into the classrooms. For our subject “multidigit multiplication” there are actual applications referred in [1] [9] [11] for example. Computer power today is often used for the delivery of content and for the administration. Adaptive skill tests are often used to efficiently estimate a global performance value.

We at the TU Graz are developing some applications (little helpers) which realize the idea to check up and train only a special curricular area. This is based on the experience that teachers refuse any innovations when they tend to intervene heavily in their management of teaching. May be that is a reason that it seems that the ITS-tradition has been forgotten in classrooms. Today, however, we see more and more a confrontation between the general requirements for greater individualization, for more and more precise feedback [8], and this denial of the teacher towards the support of technology.

We agree with Phil Long and George Siemens who stated [10] that “the most dramatic factor shaping the future of (higher) education is something that we can’t actually touch or see: big data and analytics”, the founders of the research field Learning Analytics. And on of the main issues is the question what exactly must be measured to get a deeper understanding of learning processes [4]. Furthermore nowadays a technological shift is happening to more flexible and ubiquitous applications. They are to be used in browsers or even as applications on mobile devices. This new technological environment and a new understanding of Learning Analytics at Graz University of Technology (TU Graz) are realized in applications that can be used by students on many occasions and that could contribute to an automation of basic mathematical operations [12]. Basic multiplication facts based on an automatic level of proficiency are considered to be foundational for further progress in mathematics [7]. Using this software the arithmetic technique should already be practiced in traditional ways. The program is not primarily intended to use to teach. It is a diagnostic and training tool and gives feedback about typical problems in the domain. It can be used with no further guidance or explanation. The program is intended to relieve the teacher, who usually does not have enough time and concentration to look very carefully for the causes of the mistakes. We had a look to the analog episode in the program TutorIT [11] and we interpret its intrusive help in small steps rather as a blockade than an encouragement. Our applications checks in particular, whether the solution is correct, and then we try to investigate the approach. Of course, if we get only an incorrect result, no carries, no partial calculations, we cannot go on with an analysis.

The main idea of our approach is to collect every student’s data in a central database. Tasks are given by a browser based program and by apps for iOS and Android. If a student has completed a multidigit multiplication, the program analyzes different aspects. For example if the solution is not correct, the program looks for evidence on typical failures. Is there a problem with basics, with simple multiplication facts? Is there a lack of overall competence? Are there problems with the correct placement and alignment of the carries or when writing down the partial results? Many individual data are compressed into one compact statement. This application can give teachers a powerful hint for the further promotion of each individual student. We know, that "the mathematicians were variable and versatile in their strategy use, as shown most strikingly by the fact that when 18 of the mathematicians were retested after a few months, they used different strategies on the two occasions for most of the same problems" [3, p.20]. Considering this aspect the correct solution must be recognized in any case and learners should not be forced to specific solution steps.

Finally, it is a major goal, that the results and the evaluation should not overburden teachers with details. Based on the collected data it is possible to generate detailed analyses, almost broken down to the single multiplications which has been shown to solve the task. But we said it already, it must be taken into account that teachers have only a limited time budget for their interpretation. Therefore normally the teachers were "alarmed" for doing some urgent interventions only in a few cases. It can be summarized that our approach is to develop a small web-based application where each single pupil does not get only a random selection of multidigit multiplication tasks. The program estimates a competence level and generates a hierarchy of adequate new problems. The teacher can see just in time the results and analysis of each single task just after the student has finished it.

2. GENERAL CONCEPT

First of all, the program targets school children in classes 4 and 5 - starting from an age of about 10 years. The name and layout of the web-based program was chosen with regard to children: easy and playful; it is called "Multi-Math-Coach" Learning by playing is an important concept in this respect. Knowledge of the basic multiplication tables and the ability of working with an Internet browser are the prerequisites for using the learning application. It is suggested that teachers should always give an initial introduction and overview to children of how to use and deal with the program before using it. For displaying multiplications in the front-end, the standard school method for written multiplications is used [5].

The crucial goals of our application are:

1. An overall learning algorithm that sets the difficulty for multiplications adaptively and independently for each user,
2. The generation of multiplications of a specific category (of difficulty),
3. The analysis/evaluation of the results of the generated and displayed multiplications (= user inputs), typification of the detailed results (multiplication, addition alignment).

It has already been mentioned that the learning application should adapt to the current user's expertise in multidigit multiplication knowledge. To do so, in the very first beginning categories of multiplications that differ in difficulty must be defined. Another main objective of the multiplication generation algorithm is to stay universal and to offer a large variety of examples at the same time. On the basis of [6] eight different problem groups of multidigit multiplication are distinguished. The overall domain of definition includes multiplicands (left factor) with 2 to 4 digits and multipliers (right factor) starting from 1 up to 3 digits in size. All combinations of such factors result in a total of eight problem groups, which are listed in Table 1. A specific problem group only depends on the size of its both factors. Problem groups 1 to 3, the simplest categories of multidigit multiplication, are summed up to so-called Expertise category I which is formed of multipliers with only one digit in size whereas the multiplicand varies in size. As can be seen in Table 1, a total of three overall Expertise categories are classified like this.

Table 1. The domain of definition for the Multi-Math-Coach.

Expertise	Group	Pattern
I	**1**	nn * n
	2	nnn * n
	3	nnnn * n
II	**4**	nn * nn
	5	nnn * nn
	6	nnnn * nn
III	**7**	nnn * nnn
	8	nnnn * nnn

The outcome of Table 1 now lets us differentiate between three universal categories of examples, each including a more fine-grained classification inside each Expertise class, which allows a universal allocation of multiplications in difficulty. Beside these categories of difficulty the concept of written addition and multiplication includes another dimension: the carry. In our application simple multiplications without carries and multiplications including a carry are separated. As a whole the program's domain consists of eight problem groups for simple multiplications and the same eight problem groups for multiplications including one or more carries. The ability of the application to adapt to the particular skills of the students is based on generating or providing appropriate test items according to this 16 hierarchy levels.

2.1 Generation of Multiplications

The algorithm for generating multiplications on the described sixteen levels differs from problem group to problem group. After a couple of serious tests it was realized that much of the thinking and strategies for generating the factors randomly was not effective. The simulations showed that factors with the digits 1 and 0 are more frequent than expected. For the first concept (tasks with no carry) it is very difficult to build multidigit multiplications.

One pre-condition for the algorithm is the absence of identical digits in a factor. Furthermore factors must be generated under the condition that no carry is needed to solve the multiplication.

A bunch of test runs carried out specific frequencies for the occurrence of specific digits for particular problem groups. Larger numbers in the multiplicand require smaller numbers at the multiplier, if a carry must be avoided. See Table 2 as an example where the probabilities for digits for multipliers of (the simplest) problem group 1 are shown for multiplications without carry. After generating the multiplier for a specific problem group, the multiplicand is generated. This (left) factor depends completely on the single-digit value(s) of its according factor (multiplier).

Table 2. Probabilities of occurrence of digits for multipliers of problem group 1.

Digit	Probability of appearance
1	**5%**
2	**32%**
3	**32%**
4	**21%**
5-9	**10%**

The algorithm deals with random numbers and probabilities to generate appropriate digit values for the multiplicand factor. See Table 3 for illustration where the beforehand-generated possible values for a multiplier for problem group 1 are shown with its equivalent domain of definition for the multiplicand. It is mentioned that the outermost digit to the left of a multiplicand is always generated randomly without considering the multiplier's value because this special digit can also include a special form of carry. In this case the carry is not considered for the number of successful attempts.

Table 3. Probabilities of occurrence of digits for multipliers of problem group 1.

Digit	Multiplicand (exclusive of left outermost digit)
1	**Random number out of [3, 9]**
2	**Random number out of [3, 4]**
3	**Random number out of [1, 3]**
4	**2**
5-9	**Random number out of [0, 1]**

Multipliers from expertise category II and III are not generated digit by digit. They were drawn at random from a table of all remaining possible factors.Generating factors for the category "multiplications with carry" is easier and the values are more variant but the generation happens in a similar way as explained above.

2.2 Evaluation of Multiplications

In table 4 all positive (OK) states for the different sorts of calculation processes in multidigit multiplications are shown. In the same way table 5 shows all possible error states which the program is able to identify.

Table 4. All determinable OK states and their possible occurrences in the program.

Occurrence	OK state	Description
Examples without a carry	**OK**	OK (failure-free)
Examples including a carry	**OK** **OK_MENTAL** **OK_NOCARRY** **OK_ERR**	OK (failure-free) OK (mental arithmetic) OK (mental arithmetic with succeeding failure) OK (including consequential failure)
Addition	**OK** **OK_END_ERR**	OK (failure-free) Addition OK (including consequential failure)
Overall state	**OK** **NO_ERROR**	OK (failure-free) No failure

Immediately after a user responds to a multiplication, the algorithm for evaluation of all detailed user inputs is executed. Beside of saving absolutely all user values of each input cell, the algorithm takes note of specific errors and the so-called positive states of the user input. The main objective of the evaluation process is to find out and classify typical failures done by each learner. The evaluation algorithm for multiplications including carry is much more complex than the one for simple multiplications. This is due to the fact that multiplications with carries offer different input values for different strategies. If the result is near to the correct one, a reconstruction of all steps which has been well done is not always possible. Each “OK”- and “Error” state presented in the table5 is totaled in the profile of each single student. These states are the basis for the statistical presentation and hints. For teachers the hints can be extremely helpful because the teachers have no opportunity for such a close look on a single student under normal circumstances.

Table 5. All determinable errors the program identifies.

Occurrence	Error state	Description
Examples without a carry	**ERR_INPUT** **ERR_1x1** **ERR_REV_IN**	Input failure Basics failure Reverse input failure
Examples including a carry	**ERR_INPUT** **ERR_1x1** **ERR_REV_IN** **ERR_NOADD** **ERR_ADD** **ERR_UNKNOWN**	Input failure Basics failure Reverse input failure Carry error (correct basics – no addition) Carry error (despite correct answer) Unknown failure
Addition	**ERR_INPUT** **ERR_END_ADD** **ERR_END_OK**	Input failure Addition failure Addition failure (despite correct answer)
Overall state	**ERR_MISC**	Multiple failure

2.3 Adaptive Learning Algorithm

When the program starts it is a real challenge to estimate the competency level of the actual user in our hierarchy of 16 categories in order to generate an adequate multiplication task that suits the user's current expertise.

For each user the algorithm sets six so-called expertise attributes in the database. These attributes reflect the current user's knowledge in multidigit multiplication. The six values are divided into two groups of three categories – one table for simple multiplications without carry and the other table for multiplications with a carry. The three categories reflect the three classes of expertise in table 1, respectively.

At the beginning of an assessment the current attribute values of a user are loaded from the database. If it is the first time the user is going to practice, all values are initialized to zero. The learning algorithm now chooses randomly a multiplication of the simplest expertise category (out of problem group 1 to 3), as well randomly without or including a carry. The multiplication is then displayed to the user. After solving the multiplication, the evaluation algorithm computes the state of the user input (the result) and displays the user if his or her answer was correct or incorrect. In the background the expertise attribute matching the expertise category of the generated multiplication is increased with 1, if the user's answer was correct, or decreased with 1, if the answer was incorrect. If the attribute value was zero, no decrease happens. The same happens for a value equal to three – no increase takes place. Next, another example is generated from the lowest expertise category. If an expertise attribute's value is equal to three, this means that the user has practiced three consecutive solutions and it is supposed, he or she is now able to solve multiplications from the reflected expertise category. In the following process, no more examples from this category are generated. If all six attributes are set to value three for a specific user, then this user has successfully solved the training mode. These users have the expertise to solve all multidigit multiplications in this domain. At any time during the training a so-called training example is generated and displayed to the user (10% of occurrences). Such a training example is simply a multiplication chosen from the already solved expertise categories. This serves as a refreshment of knowledge and leads to strengthening the training purpose. If no multiplications were solved correctly and a training example is chosen randomly, then an alert for the teacher is generated.

3. TECHNICAL DETAILS

The web-based bilingual application is programmed in PHP 5 using the Zend Framework. For layout and design purposes Cascading Style Sheets (CSS) and jQuery are used. MySQL was chosen as a relational database management system. All mentioned technologies are Open Source and can be freely adapted and distributed.

4. USER INTERFACE

First of all, there is a start page where users are able to log in with a specific username and password. To be able to log in and to play with the multiplication training program, users have at first to register for free on a separate webpage (http://mathe.tugraz.at). After the login process students enter a welcome-page ("Are you ready?") as a preparation for the training mode. Logged-in teachers will see an additional "Statistics" button at the top of the page. When clicking on the green button ("Let's go"), the training mode starts and according to the described algorithm an appropriate example is presented.

Figure 1 shows a correct result of the overall calculation (user inputs) of a multiplication of expertise category II without add carry. All input cells were checked and marked correctly in the evaluation process. The evaluation algorithm therefore computes a statistic on positive result states, which have been saved in correspondent database tables. Additionally, the learning algorithm increases the user's property for expertise category II (compare the two-digit multiplier) by adding 1 to his/her state.

Figure 1. Correctly solved multidigit multiplications.

Figure 2 demonstrates a wrong answer of a filled-in multiplication calculation of the most difficult problem group (8) including carry. Wrong input cells are marked with a red background to advise users exactly where failures occurred. The evaluation algorithm has computed result states for each input cell, as well as an overall result state. The second row's result-states include, for example, correct calculations (OK), a correct mental calculation (OK_MENTAL) as well as the state OK_NOCARRY and an unknown error (ERR_UNKNOWN). The third result row's outcome states include: OK, ERR_NOADD, ERR_UNKNOWN, OK_ERR (see tables above for an explanation of specific result states).

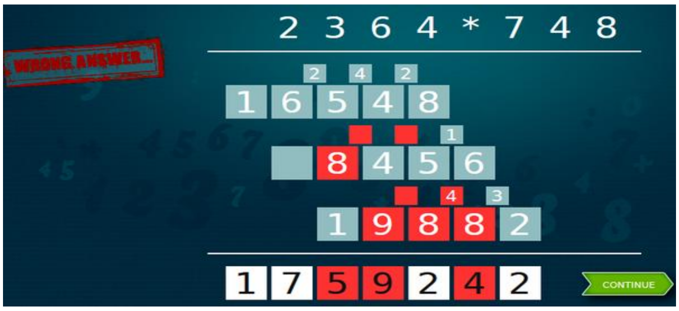

Figure 2. Wrongly answered multidigit multiplications.

Figure 3 points out the statistics of all grades and students belonging to a teacher/class. Each student is listed with its name, total amount of solved examples, total amount of correct solutions, total amount of incorrect solutions, as well as a computed success rate and error burst. A positive success rate (> 70%) is marked with a green background whereas a negative success rate is marked with red background. A link leads to detailed table for a selected student. This figure shows the typically error bursts:

- Basic errors (less competence in handling the multiplication tables)
- Problems with the carry
- Multiple Errors

GRADE saaralt

User name	Overall answers	Correct answers	Incorrect answers	Success rate	Error burst	Details
Basti05 Never	21	14	7	66.67 %	Multiple errors	› HERE ‹
Anton Cary	11	7	4	63.64 %	Carry addition error (correct basics - no addition)	› HERE ‹
no1x1 mining	22	9	13	40.91 %	Basics error	› HERE ‹

Figure 3. List of learners and compressed statistics (Error burst).

3868*48

Basic multiplication	Correct carry to add	Correct Solution	User carry input to add	User answer	New carry	Result status
8*4	OFF	32	OFF	32	3	OK
6*4	3	27	3	27	2	OK
8*4	2	34	2	34	3	OK
3*4	3	15	3	15	OFF	OK

Figure 4. Detailed analysis of one task.

Figure 4 shows the top of the most detailed listing. (This is only optionally shown!) It includes each single input (cell) of a single multiplication problem. It shows exactly the evaluation of the generated problem of figure 2, including the computed positive and erroneous states as explained before.

5. EVALUATION

The application is available for free since January 2013. Teachers as well as children can register by going to the website http://mathe.tugraz.at. The username can be used for all three offered applications – the multiplication trainer [12], the addition and subtraction trainer [15] and the multi-math-coach. Furthermore schools as well as classes can be defined by teachers to get an appropriate overview for the group of learners.

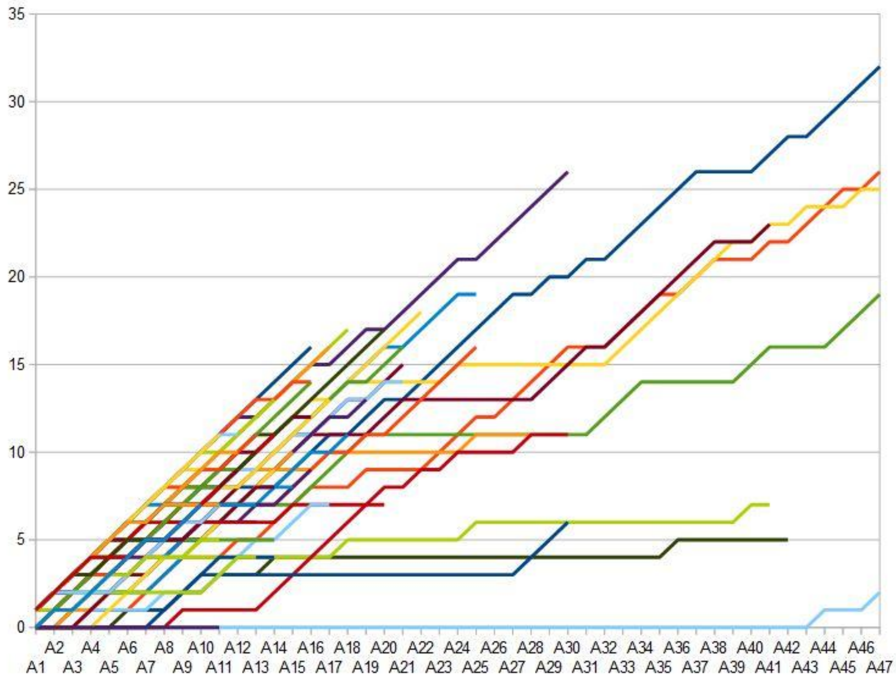

Figure 5. First analyses – learning curves.

After 6 months a first analysis of the data was done, to carry out if the application is working as intended. At this time 936 users were registered. We filtered out of the data only users who have done more than three tasks. Finally 251 users solved altogether 2.575 Tasks. Fig. 5 shows the steadily increasing learning curves of most of the users. A more horizontal line marks a sequence of mistakes and wrong results. This is identified and teachers are noticed of such occurring problems and are able to intervene in a very early stage when they inspect the class statistics. From the application perspective it is not possible to recognize, if the learner has a problem due to missing math-competences for solving the tasks or if there are other motivational problems.

Table 6. Number of Error types.

Error type	Number	[%]
Tasks with errors	1020	
missing digits (ERR_INPUT)	350	34
no identifiable error (ERR_UNKNOWN)	105	10
more than one explanation (ERR_MISC)	219	21
wrong multiplication (ERR_1x1)	188	18
multiplication ok, no addition (ERR_NOADD)	126	12
last addition (ERR_END_ADD)	21	2
transposed digits when entering (ERR_REV_IN)	11	1

In Table 6 the statistic of the investigated group and their errors are shown. The interesting outcome is the fact that we have already carried out interviews with teachers about what they would see as a major cause of errors in the written multiplication. There was broad agreement that the main cause of error is the weaknesses in the mastery of the simple multiplication tables. From investigated 2575 tasks 1555 are right answers (60%). In 169 cases (OK_MENTAL) the pupils didn't use the possibility to enter carries, they solved the problem mentally (7%). The most frequent errors are missing digits (ERR_INPUT) with about 34%. The children didn't enter all necessary digits, but even 98 (4%) still had the right solution. However, these errors are an artifact that results from the work on the screen. In 21% of the solved tasks our application found out more than one typical problem. Perhaps the other perspective on each single learner on the base of more frequent solutions could show a typical problem of him/her. The most remarkable result of our research is that the application identified only 18% errors caused by problems which can be traced back to the multiplication table (ERR_1x1). 12% had definitely no problems with multiplications (ERR_NOADD) but with carries during the addition and 2% had other or further problems with the addition (ERR_NOADD). 1% struggles with entering the digits in the right sequence (ERR_REV_IN). This could be a language problem – for example "fourteen" is written by numbers as "14" but the word suggests a sequence 4 and then 1. Overall, it can be stated that the application is able to provide the necessary training for written multiplication. Additionally diagnostic information about different problems are gathered which helps to give learners differentiated and targeted instructions in a very personalized way.

6. CONCLUSION

The ubiquity of laptops, netbooks, tablet computers, and smartphones among young people led to a complete new way of dealing with learning materials as well as watching learning processes. Research in the field of Learning Analytics seems to be a big task in the future. For example, many teachers express their fears about being replaced or to be less or not necessary anymore. Scandura [11] spread in his article exactly this issue and answered immediately: "What TutorIT can do better than a human and Why: Now and in the Future". Nevertheless, our research team is not sharing this opinion. The role of tomorrow´s teachers is much more like to facilitators and coaches who assist learners according to their individual needs. Many teachers are offering other hints and additional material. Our application should have the role of such an additional content. We don't dictate the students the individual solution steps. If anyone prefers only mentally computing, the right solution is accepted. If someone does not use the prepared input fields for the carries, this will be reported.

Furthermore our application gathers more details about learning multidigit multiplication. Resulting from many personal discussions with teachers about common failures our team got the impression that teachers almost exclusively see only one cause of problems: multiplication facts. Due to the fact that teachers usually have no time to take a closer look or to figure out systematically what might have caused each single failure this application should close this gap. The program tests not only for competence in multiplication facts. Additionally it takes also a look on other fundamental skills. Finally, if teachers use the advice of the program, they can try to help the students in targeted manner. In some cases, the students will probably have other physiological and psychological problems which were presented in detail in in Dowker [3].

For the future of the program also the issue "processing time" is a challenge. This is corresponding to the goal of Hasselbring [7]: an "automatic level of proficiency". This is a category that would be easy to implement in a diagnostic system. In a close cooperation to further interested institutions this issue as well as a huge field study in different classrooms will be carried out in future. It can be concluded that Learning Analytics in combination with the possibilities of innovative technologies and innovative teaching and learning strategies is a powerful field to enhance learning processes of each individual learner.

The apps and the user account for this and other little helpers are available for free on http://mathe.tugraz.at . Choose "Registrieren" and then "English". With a single e-mail address you can get an account to test the functionality. To get accounts for schools and classes please have contact with martin.ebner@tugraz.at.

REFERENCES

[1] Brothersoft.de: Programm Schriftliche Multiplikation. http://www.brothersoft.de/Schriftliche-Multiplikation-download-152702.html.

[2] Brown, J.S. and Vanlehn, K, 1980. Repair Theory: A Generative Theory of Bugs in Procedural Skills. *Cognitive Science*, Vol. 4, pp 379-426.

[3] Dowker, A. Individual Differences in Arithmetic. http://zung.zetamu.net/Library/Education/Education_Neuroscience/Dowker_IndividualDifferencesInArithmeticalAbilities__2005.pdf

[4] Duval, E. 2010. Attention Please! Learning Analytics for Visualization and Recommendation. *Proceedings of LAK11: 1st International Conference on Learning Analytics and Knowledge 2011*.

[5] Eigenwillig, A. and Mehlhorn, K., 2008. Multiplikation langer Zahlen (schneller als in der Schule). Springer, eXamen.press. *Taschenbuch der Algorithmen*. Chap. 11, 109-118.

[6] Gerber, M. M. et al, 1994. Computer-Based Dynamic Assessment of Multidigit Multiplication. *Exceptional Children 61*. 114-125.

[7] Hasselbring, T. et al, 1988. Developing math automaticity in learning handicapped children: The role of computerized drill and practice. *Focus on Exceptional Children*, 20(6), 1-7.

[8] Hattie, J. Visible Learning, Tomorrow's Schools, The Mindsets that make the difference in Education. http://www.treasury.govt.nz/publications/media-speeches/guestlectures/pdfs/tgls-hattie.pdf

[9] Landesbildungsserver Baden Würtemberg. Programm Schriftliche Multiplikation: http://www.schule-bw.de/schularten/grundschule/2gsunterricht/4medien/schmulti/multiprog3.htm

[10] Long, P., Siemens, G., 2011. Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review Magazine*. Volume 46. 5. p. 31-40

[11] Scandura, J.M., 2011. What TutorIT can do better than a Human, and Why - Now and in the Future. *TICL*, 2011, 8, 175-227. Publication version of Chairman's Address TICL SIG, AERA, May 2, 2010 (http://www.tutoritweb.com/)

[12] Schön, M. et al, 2012. It's Just About Learning the Multiplication Table, *In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12)*, Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 73-81

[13] Sleeman D. and J. S. Brown, 1982. *Introduction: Intelligent Tutoring Systems. In Intelligent Tutoring Systems*. Orlando, Florida: Academic Press, Inc. 1-10

[14] Wong, M., Evans, D., 2007. Improving Basic Multiplication Fact Recall for Primary School Students. *Mathematics Education Research Journal* 19 No.1

[15] Ebner, M., Schön, M. (2013) Why Learning Analytics in Primary Education Matters!, Bulletin of the Technical Committee on Learning Technology, Karagiannidis, C. & Graf, S (Ed.), Volume 15, Issue 2, April 2013, p. 14-17