Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 176 (2015) 876 – 883

Procedia
Social and Behavioral Sciences

IETC 2014

Measuring the level of algorithmic skills at the end of secondary education in Hungary

Piroska Biró[a1], Mária Csernoch[a], János Máth[b], Kálmán Abari[b]

[a]*University of Debrecen, Faculty of Informatics, Kassai út 26., Debrecen, 4028, Hungary*
[b]*University of Debrecen, Faculty of Arts and Humanities, Komlóssy út 45-47., Debrecen, 4032, Hungary*

Abstract

Students starting their tertiary studies in Informatics are found to have a low level of algorithmic skills and understanding of programming, which leads to the high number of drop out students and failed semesters during their studies. The students' low level of programming skills contrasts with their excellent results in the school leaving exams. To find out the reasons for this we have launched the TAaAS project (Testing Algorithmic and Application Skills), which focuses on the students' algorithmic skills and programming ability in traditional and non-traditional programming environments. Our analyses proved that school leaving exams are not able to measure these abilities of the students, and beyond that, are not able to distinguish between the different levels of the students. Students are accepted into the universities and start their studies based on the misleading results of the school leaving exams.

Peer-review under responsibility of the Sakarya University.

Keywords: level of digital thinking, algorithmic skills, school leaving exams in Informatics and Mathematics

1. Introduction

As early as 1995 a new National Curriculum was introduced in Hungary including formal Computer Sciences/Informatics (CSI) education, which promised to deliver well-developed algorithmic skills. Since then digital competency and computational thinking is regarded as one of a child's key analytical abilities and is intended to be developed both in formal CSI studies and in traditional school subjects. To support this concept, the structure of the traditional subjects was changed, formal CSI studies were introduced, and the connections between this new subject and the others were clearly stated. In addition to these fundamental changes, in 2005 a new system of school

1 * Corresponding author. Tel.: +3652512900/75227
E-mail address: biro.piroska@inf.unideb.hu

Peer-review under responsibility of the Sakarya University.
doi:10.1016/j.sbspro.2015.01.553

leaving exams was launched with a double-purpose: serving both as the closing event of the primary and secondary studies and as the entrance exam to tertiary education.

Based on these changes in the education system in support of the development of computational thinking (Wing, 2006), the straightforward consequence would be that our students achieve a high level of algorithmic skills, especially those students who choose CSI as their major in tertiary education. However, the pattern is not this clear. Institutes in CSI tertiary studies are faced with the problem that the subjects and the level of the entrance exams are quite contradictory in terms of the requirements of CSI studies. The contradiction is further proved by the high number of drop out students and failed semesters.

To clear up this misunderstanding we have launched a project entitled Testing Algorithmic and Application Skills (TAaAS), which focuses on the students' algorithmic skills in comparison to their results in the school leaving exams (Biró & Csernoch, 2013a, 2013b, 2013c, 2014; Biró et al, 2014a, 2014b; Csernoch & Biró, 2013a, 2013b, 2013c, 2014a, 2014b, 2014c; Csernoch et al, 2014).

2. Sample

2.1. Participating students

The TAaAS project was launched in the 2011/2012 academic year, at the Faculty of Informatics at the University of Debrecen, Hungary, and has been running since then. In the following year the testing was repeated with students newly arrived at our faculty, while in the 2013/2014 academic year the project was widened, and three more Hungarian institutes joined (Eötvös Loránd University (ELTE, Budapest), Eszterházy Károly College (EKF, Eger), and the College of Nyíregyháza (NYF, Nyíregyháza) (Biró et al, 2014a, 2014b; Csernoch & Biró, 2013).

In the following three years at the University of Debrecen (DE) three majors of Informatics were tested including a total of 950 students: Software Engineering (SOE), System Engineering (SYE), and Business Information Management (BIM) (Table 1).

Table 1. The number of students at the Faculty of Informatics at the University of Debrecen, Hungary participating in the TAaAS project

	SOE	SYE	BIM	Sum
2011/2012	115	86	109	310
2012/2013	108	111	101	320
2013/2014	115	115	90	320
Sum	338	312	300	950

2.2. The structure of the school leaving exams

The school leaving exams (School leaving exams, 2014) serve both as the closing event of the primary and secondary studies and as the entrance exam for tertiary studies, and run at intermediate and advanced levels. There are four compulsory subjects – Mathematics, Hungarian, History, and a foreign language –, and at least one more, which is chosen by the students. It is the students' choice at which level(s) they take the exams; however, tertiary education institutes indicate the required subjects and their levels. In order to start a tertiary course in CSI the result of the compulsory Mathematics school leaving exam must be taken into consideration. However, it is not compulsory to take the school leaving exam in Informatics, and this can be substituted with Physics in SOE and SYE, and with other subjects in Sciences in BIM. Furthermore, even the results of exams in Hungarian, History, and foreign languages are considered when calculating the results of the entrance exams for CSI studies. The Informatics school leaving exam runs at both levels, and they consist of a computational and an oral session. At intermediate level in the computational session only application problems have to be solved, while at advanced level there are more complex application tasks and one programming task.

The following question naturally arises in this context: how well is this system of school leaving exams able to measure the students' algorithmic skills, their level of computational thinking, and their correct usage of terminology in CSI, all of which are requirements for tertiary CSI institutes? In sum, we can ask how well students are prepared for higher level studies in CSI.

2.3. The tasks of the TAaAS project

The TAaAS project was launched to test the students' algorithmic skills in traditional and non-traditional programming environments, to reveal how the students approach programming problems in the different environments, and how they would relate the computer related problems to algorithms (Soloway, 1993; Warren, 2004; Sestoft, 2010). We have selected three different environments to test the students' algorithmic skills, two of which are rather traditional (Tasks 1 and 2), while one is a new programming environment (Task 3).

2.3.1. Logical operators

Task 1 is a traditional program code of a multilevel IF structure to test the students' ability to recognize logical operators. The possible output of the program for the pairs of inputs is one of four whole numbers: 3, 2, 1 and 0. The source code is accompanied by a table of nine pairs of inputs and nine empty cells, where the selected output numbers have to be written. The presentation of the problem with the limited number of possible answers and the output table made the task quite easy (Fig. 1).

You draw two cards (X, Y) from two packs. In both packs you can find cards with the letter A, cards with the letter B and cards with the number zero. Give the points in the last column of the table according to the algorithm given below.

```
V:=X=”A” or Y=”A”
W:=X=”B” or Y=”B”
If V and W then Point:=3
else If V then Point:=2
else If W then Point:=1
else Point:=0
```

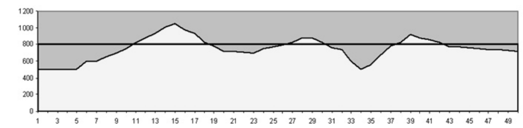
X	Y	Point
A	A	
A	B	
A	0	
B	A	
B	B	
B	0	
0	A	
0	B	
0	0	

Fig. 1. The text, the program code and the output table of Task 1.

2.3.2. Decoding pseudo codes

The other traditional programming task of the test (Task 2) contains three pseudo codes which the students have to decode, and the results should be presented in semantically correct natural language sentences. Compared to Task 1, these problems are a lot more demanding. However, we have to note here that both tasks are borrowed from the Nemes Tihamér Hungarian national programming contest for 5–8th graders (Nemes, 2013).

What do the following programs do? What is stored in variable DB and M?



N=50, measured values:
500,500,500,500,500,600,600,650,700,750,820,880,930,1010,1050,980,930,830,780,720,720,710,700,750,770,790,820,880,880,820,760,740,600,500,560,670,770,770,760,750,740,740,730,720.

```
Task 2.1   DB:=0
           Loop from i=1 to N
             If X(i)>800 then DB:=DB+1
           End loop
Task 2.2   DB:=0
           Loop from i=2 to N−1
             If X(i)<X(i−1) and X(i)<X(i+1) then DB:=DB+1
           End loop
Task 2.3   M:=0
           Loop from i=2 to N
             If X(i)−X(i−1)>M then M:=X(i)−X(i-1)
           End loop
```

Fig. 2. The context and the pseudo codes of Task 2.

2.3.3. Programming in spreadsheet

Spreadsheet is not usually considered a programming environment, but a user-friendly interface which even computer illiterate end-users would be able to use. However, it has been proved that this is not so (Warren, 2004; Sestoft, 2010; Csernoch, 2012). The high number of spreadsheet documents carrying errors (Panko, 2010; Tort, Blondel & Bruillard, 2008) and the extremely long time required for the preparation of these documents (Van Deursen & Van Dijk, 2012) has proved that spreadsheet should be taken more seriously. Spreadsheet is a functional language, and this could serve as an introductory language due to its simplicity and the problems related to it. We chose these problems to reveal how students handle them, and beyond that they are closely related to the other two more traditional programming tasks, and so allow us to obtain comparable results. Tasks 3.1–3.3 should be answered with complete spreadsheet formulas. Task 3.4 is a decoding task with a double purpose: on the one hand, the results should be expressed in a natural language sentence, similar to Task 2; on the other hand, this complete formula would provide guidelines for solving Tasks 3.1–3.3. We wanted to see whether the students recognize the connection between these problems or not.

	A	B	C	D	E
1	Country	Continent	Capital	Area	Population (thousand)
2	Afghanistan	Asia	Kabul	647500	27756
3	Albania	Europe	Tirana	28748	3545
4	Algeria	Africa	Algiers	2381740	32278
5	American Samoa	Oceania	Pago Pago	199	69
6	Andorra	Europe	Andorra la Vella	468	68
7	Angola	Africa	Luanda	1246700	10593
8	Anguilla	America	The Valley	102	12
233	Yemen	Asia	Sanaa	527970	18701
234	Yugoslavia	Europe	Belgrade	102350	10657
235	Zambia	Africa	Lusaka	752614	9959
236	Zimbabwe	Africa	Harare	390580	11377

Task 3.1	How many African countries are in the table?
Task 3.2	What is the average population of those countries whose surface area is smaller than G1?
Task 3.3	How many countries have a surface area greater than G1?
Task 3.4	What is the result of the following formula? {=SUM(IF(B2:B236="Europe",IF(LEFT(A2:A236)="A",1)))}

Fig. 3. Presenting Task 3, with the context, the questions to answer with spreadsheet formulas, and the code for decoding.

2.4. The evaluation of the tasks of the TAaAS project

The correction of Task 1 was unambiguous; the number of the correct answers had to be counted. However, both Tasks 2 and 3 required more serious consideration. To find out how the students approach these tasks, and how their algorithmic skills are developed we adapted the categories of understanding of the SOLO taxonomy to both the traditional and the non-traditional programming tasks. Considering the special environment of the tasks and the preliminary corrections, the following five levels of understanding are included: Ignored (1), Prestructural (2), Unistructural (3), Multistructural (4), Relational (5) (Biggs & Collins, 1982; Lister et al, 2006, Clear et al, 2008, Sheard et al, 2008, Tan & Venables, 2010).

3. Hypotheses

H1: The students' selection of major indicates that their results are higher in the Informatics than in the Mathematics school leaving exams.

H2: There is a connection between the students' results in the test focusing on algorithmic skills and in their results in the school leaving exams.

4. Results

4.1. The results of the school leaving exams

The comparison of the number of the participating students shows that most of the students take the Informatics school leaving exam, even though it is not compulsory (Table 2). The difference between the number of participating students and those who take the exam in Mathematics is due to students arriving from foreign countries and to uncompleted questionnaires (Table 1 and 2). However, the low number of the advanced level exams is remarkable, both in Mathematics and Informatics. The only exception is the SOE students' Informatics exam at advanced level.

Table 2. The number of students and their results in the school leaving exams in Informatics and Mathematics

	SOE	SYE	BIM	Average
school leaving exam (SLE) – intermediate level				
Informatics	84.1 (N=175)	82.2 (N=237)	80.8 (N=193)	82.3 (N=605)
Mathematics	74.1 (N=277)	71.4 (N=265)	74.4 (N=252)	73.3 (N=794)
school leaving exam (SLE) – advanced level				
Informatics	72.5 (N=127)	66.4 (N=37)	55.7 (N=16)	69.7 (N=180)
Mathematics	68.1 (N=22)	70.3 (N=17)	68.9 (N=24)	69.0 (N=63)

The comparison of the results of school leaving exams shows no significant differences between the majors, except in Informatics at advanced level. The comparisons of the pairs with the Kruskal-Wallis probes only found a difference between the results of the SOE and the BIM students in Informatics, at both levels. We can conclude that students start their tertiary CSI studies at the Faculty of Informatics with similar knowledge, both in Informatics and Mathematics, regardless of their majors.

The results of the school leaving exams at intermediate level in Informatics and Mathematics show significant differences (Wilcoxon signed rank test SOE, SYE, BIM: $V=1487$, $p<0.001$, $V=3386$, $p<0.001$, $V=3522$, $p<0.001$, respectively). Consequently, the results in Informatics are significantly higher than in Mathematics, which proves our H1 hypothesis. Based on the results of the school leaving exams only two clusters were distinguishable: C1SLE and C2SLE with 56.4% and 43.6% of the students, respectively. Their results in Informatics are 87.74% and 74.84%, in Mathematics 81.91% and 60.45%, respectively. Both the majors as categories and the clusters indicate that the school leaving exams are not able to distinguish between the students (Chi-square test: $\chi^2(2)=3.82$, $p=0.148$).

4.2. Knowledge-based clusters

Since the results of the school leaving exam do not show differences between the majors, we selected Task 1 to create knowledge-based clusters to see whether we could differentiate the students or not. With Task 1 we found four knowledge-based clusters: C1L–C4L, from the best group to the worst, respectively. With these clusters we can tell the students apart based on their results in the logical task (Fig. 4). The students who ignored or did not finish Task 1 belong to C4L, the best students to C1L, while C2L and C3L fall between these two. The question was how the clusters would differentiate between the students. The performance of C2L was found somewhat arbitrary; we could not find any characteristic feature for this cluster. However, C3L proved to have some limited knowledge. Those who belong to this cluster calculated the output almost as perfectly as those in C1L with A and B pairs of inputs; however, when one of the inputs was 0 they provided a 0 output, without considering the algorithm.

In Task 2 the high number of Level 1 results (Ignored) is remarkable; however there are significant differences between the clusters; with the increase in the number of the clusters the students' results are significantly lower (Jonckheere-Terpstra Test: $p<0.001$ in all the three codes). The Mann-Whitney test proved that in all the three pseudo codes the adjacent clusters differ significantly ($p<0.05$), with the exception of C2L and C3L in Task 2.1, but even in this case the direction of the difference is the same as with the other pairs. Further analyses of the clusters in Task 2 revealed a similar pattern as that found in Task 1. C4L achieved the lowest level; most of these students stop at Level 1. The most frequent result for C1L in Task 2.1 is Level 5. Even in this group, both in Task 2.2 and 2.3, the number of students ignoring the tasks is higher than in any of the other levels. However, in Task 2.2 the second most frequent level is 5, while in Task 2.3 it is 4. In C2L and C3L the second most frequent level is 3. However, in Tasks 2.1 and 2.2, which are the easier tasks, more C2L than C3L students reach Level 5, while in Task 2.3, which is the most difficult, the results of C2L decrease with the increasing level of understanding, and from Level 3 the frequency of C2L is below C3L. This result is a further proof of the limited knowledge of C2L.

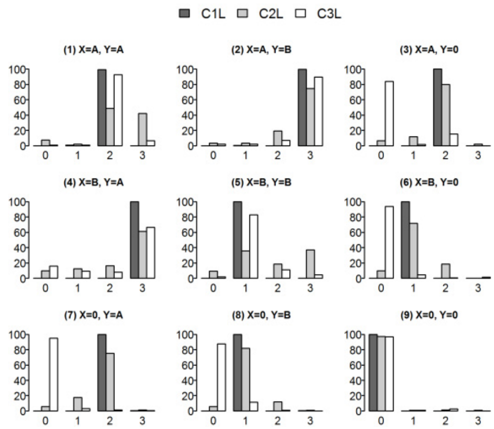


Fig. 4. The results of the clusters in Task 1.

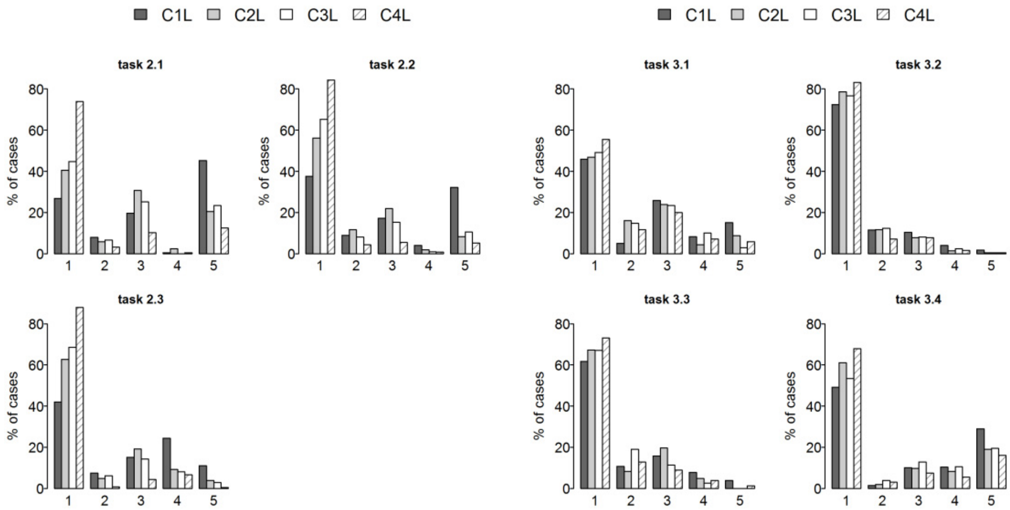


Fig. 5. The results of the clusters in Tasks 2 and 3.

In general, the result of Task 3 is the lowest among the three programming tasks. The high number of the ignored problems in this task is remarkable in all the clusters, especially for Tasks 3.2 and 3.3, which are the generalizations of Task 3.1. Significant differences between the clusters were only found in Tasks 3.1 and 3.4 (Kruskal-Wallis probes, Tasks 3.1, 3.2, 3.3 and 3.4: $\chi^2(3)=19.27$, $p<0.001$; $\chi^2(3)=8.67$, $p=0.034$; $\chi^2(3)=3.07$, $p=0.38$; $\chi^2(3)=25$, $p<0.001$). However, a comparison of the pairs show only differences in Task 3.1 between C1L and C3L-C4L, and in Task 3.4 between C1L and C4L. Level 5 is achieved with the highest frequency only in Task 3.4, which is the decoding problem. In this problem C1L performs better than the other clusters, while there are hardly any differences between the other clusters. In the other tasks the most frequent levels are 2 and 3, and C1L is slightly better in Task 3.1 than the others. We can conclude that even the knowledge based clusters which worked well in the traditional programming environment are not able to distinguish between the different weaknesses of the students in spreadsheet. Based on the comparison of the results in the school leaving exams and the test, we have found that while our knowledge based clusters are able to distinguish the students' levels of algorithmic skills, the school leaving exams do not have this feature. Consequently, we have to reject our H2 hypothesis.

5. Conclusions

In the framework of the TAaAS (Testing Algorithmic and Application Skills) project we have tested the first year students of the Faculty of Informatics at the University of Debrecen, Hungary, focusing on the students' algorithmic skills, and their level of understanding in traditional and non-traditional programming environments. We have realized that the students arriving at our faculty do not match the requirements set by our majors, in spite of the students' high results in the school leaving exams. This contradiction leads to the high number of drop out students and failed semesters.

The present article focuses on a comparison of the first year students' results in the school leaving exams and in the TAaAS project. First of all, the low number of students taking the advanced level school leaving exams is remarkable (except with the SOE students in Informatics), consequently, we rely mainly on the results at intermediate level. The analysis proved that the results of the school leaving exams in Informatics are significantly higher than in Mathematics. This would explain the students' low results in the different subjects in Mathematics in tertiary CSI education. It was also found that the majors do not show significant differences in their results, either in Informatics or Mathematics. Consequently, based on the school leaving exams only two clusters are recognizable, i.e. the school leaving exams seem an unsatisfactory to distinguish between the different levels of the students. Furthermore, the good results in Informatics suggest that the students start their studies in tertiary education with a high level of computational thinking.

To find the sources of the contradiction between the entrance exam results of the students and their real performance in tertiary education we selected algorithm driven tasks in the TAaAS project. The students' results in these tasks revealed that they have a low level of algorithmic skills and understanding of programming.

The most successful task in the test, Task 1, was selected to create knowledge based clusters, and four well distinguishable clusters were recognizable; C1L–C4L, moving from the best to the worst. The best cluster is proved to be the best, while the worst is the worst in all the three tasks. Between them two clusters are also clearly distinguishable. C2L's results seem rather arbitrary, while C3L is found to have limited knowledge. The characteristic feature of this type of knowledge is that until they reach their limit they achieve similarly good results to C1L, while with more difficult tasks they perform at the lowest level of understanding.

Tasks focusing on the programming skills are able to distinguish the different levels of algorithmic skills and levels of understanding in programming. On the one hand, we can conclude that the system which relies heavily on the results of the school leaving exams when selecting students for Informatics courses does not serve its purpose. On the other hand, neither the Mathematics, and most unfortunately, nor the Informatics school leaving exam at intermediate level is able to measure the level of the students' algorithmic skills.

Acknowledgements

The research was supported by the TÁMOP-4.2.2.C- 11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

The research was partly supported by the Hungarian Scientific Research Fund under Grant No. OTKA K-105262.

References

Biggs, J. B. & Collis, K. E. (1982). Evaluating the Quality of Learning: The SOLO Taxonomy. New York, Academic Press.

Biró, P. & Csernoch, M. (2013a). An Algorithmic Approach to Spreadsheets, in Hungarian, Kiss Árpád Emlékkonferencia, Debrecen, accepted.

Biró, P. & Csernoch, M. (2013b). Deep and surface structural metacognitive abilities of the first year students of Informatics. 4th IEEE International Conference on Cognitive Info-communications, Proceedings, Budapest, pp. 521–526.

Biró, P. & Csernoch, M. (2013c). Programming skills of the first year students of Informatics. XXIII. International Conference on Computer Science 2013, EMT, in Hungarian, pp. 154–159.

Biró, P. & Csernoch, M. (2014). Deep and surface metacognitive processes in non-traditional programming tasks. 5th IEEE International Conference on Cognitive Info-communications Conference, 4–5. November, 2014, Vietri sul Mare, Italy, submitted.

Biró, P. Csernoch, M., Abari, K. & Máth J. (2014a). First year students' algorithmic skills in tertiary Computer Science education, 9th International Conference on Knowledge, Information and Creativity Support Systems, 6–8. November, 2014, Cyprus, Limassol, submitted.

Biró, P. Csernoch, M., Máth, J. & Abari, K. (2014b). Algorithmic skills transferred from secondary CSI studies into tertiary education, The 6th IEEE International Conference on Technology for Education, 18–21. December, 2014, Kerala, India, submitted.

Clear, T., Whalley, J., Lister, R., Carbone, A., Hu, M., Sheard, J., Simon, B. & Thompson, E. (2008). Reliably Classifying Novice Programmer Exam Responses using the SOLO Taxonomy. 21st Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2008), Auckland, New Zealand. Samuel Mann and Mike Lopez (Eds).

Csernoch, M. & Biró, P. (2013a). Teachers' Assessment and Students' Self-Assessment on The Students' Spreadsheet Knowledge. EDULEARN13 Proceedings of the 5th International Conference on Education and New Learning Technologies, July 1st-3rd, 2013 – Barcelona, Spain. Ed. by L. Gómez Chova, A. López Martínez, I. Candel Torres. International Association of Technology, Education and Development. pp. 949–956.

Csernoch, M. & Biró, P. (2013b). Testing algorithmic and application skills. In Hungarian: Algoritmikus és alkalmazói készségek tesztelése. INFODIDACT Conference 2013, Zamárdi, Hungary, 21–22. November, 2013.

Csernoch, M. & Biró, P. (2013c). The investigation of the effectiveness of Bottom-up techniques in the spreadsheet education of students of Informatics. (Eds): Kozma T. and Perjés I., New Research in Education Studies. ELTE Eötvös Publisher, Budapest. pp. 369–392, in Hungarian.

Csernoch, M. & Biró, P. (2014a). Digital Competency and Digital Literacy is at Stake, ECER 2014 Conference, 1–5. September, 2014, Porto, Portugal, accepted.

Csernoch, M. & Biró, P. (2014b). Spreadsheet misconception, spreadsheet errors. Oktatáskutatás határon innen és túl. HERA Évkönyvek I., ed. Juhász Erika, Kozma Tamás, Publisher: Belvedere Meridionale, Szeged, ISSN 2064-6755, pp. 370-395.

Csernoch, M. & Biró, P. (2014c). The power in digital literacy and algorithmic skill, INTE 2014 Conference, 25–27. June, 2014, Paris, Procedia – Social and Behavioral Sciences, accepted.

Csernoch, M. (2012). Introducing Conditional Array Formulas in Spreadsheet Classes. EDULEARN12 Proceedings. Barcelona, Spain. 2-4 July, 2012. Publisher: IATED, pp. 7270–7279.

Csernoch, M., Biró, P., Máth, J. & Abari, K. (2014). What do I know in Informatics? Mit tudok informatikából? in Hungarian. IF2014 Conference, 27–29 August, 2014, Debrecen, Hungary, accepted.

Lister, R., Simon, B., Thompson, E., Whalley, J. L. & Prasad, C. (2006). Not seeing the forest for the trees: novice programmers and the SOLO taxonomy, in Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, New York, NY, USA, pp. 118–122.

Nemes (2013). Nemes Tihamér National Competition of Informatics 2008-2009 round I, 5-8th grade. Exercices (Nemes Tihamér Országos Informatikai Verseny 2008-2009 I. forduló, I. korcsoport, 5-8. osztály. Feladatok) in Hungarian. Retrieved May 17, 2013 from http://nemes.inf.elte.hu/2009/nt09-1f1.doc.

Panko, R. and Aurigemma, S. (2010). Revising the Panko-Halverson taxonomy of spreadsheet errors. Decision Support Systems 49, 2 (2010), pp. 235–244.

School leaving exams (2014). School leaving exams in Hungary – Tasks and correcting guides. In Hungarian: Központi írásbeli feladatsorok, javítási-értékelési útmutatók. Retrieved June 15, 2014 from http://www.oktatas.hu/kozneveles/erettsegi/feladatsorok.

Sestoft, P. (2010). Spreadsheet technology. Version 0.12 of 2012-01-31. IT University Technical Report ITU-TR-2011-142. IT University of Copenhagen, December 2011.

Sheard, J., Carbone, A., Lister, R., Simon, B., Thompson, E. & Whalley, J. L. (2008). Going SOLO to assess novice programmers, SIGCSE Bull., vol 40, 3, pp. 209–213.

Soloway, E. (1993). Should we teach students to program? Communications of the ACM. October 1993/Vol.36, No.10, pp. 21–24.

Tan, G. & Venables, A. (2010). Wearing the Assessment 'BRACElet'. Journal of Information Technology Education: Innovations in Practice. Vol. 9.

Tort, F. Blondel F. & Bruillard, É. (2008). Spreadsheet Knowledge and Skills of French Secondary School Students. R.T. Mittermeir and M.M. Sysło (Eds.): ISSEP 2008, LNCS 5090, 2008, pp. 305–316. Springer-Verlag Berlin Heidelberg.

Van Deursen A. & Van Dijk J. (2012). CTRL ALT DELETE. Lost productivity due to IT problems and inadequate computer skills in the workplace. Enschede: Universiteit Twente. Retrieved 02. June 2014. from http://www.ecdl.org/media/ControlAltDelete_LostProductivityLackofICTSkills_UniverstiyofTwente1.pdf.

Warren, P. (2004). Learning to program: spreadsheets, scripting and HCI, in Proceedings of the Sixth Australasian Conference on Computing Education – vol. 30, Darlinghurst, Australia, pp. 327–333 (2004).

Wing, J. M. (2006). Computational Thinking. March 2006/Vol. 49, No. 3 Communications of the ACM.