Semi-Unplugged Tools for Building Algorithms With Sprego

Piroska BIRÓ
Faculty of Informatics
University of Debrecen
Hungary
biro.piroska@inf.unideb.hu

Mária CSERNOCH
Faculty of Informatics
University of Debrecen
Hungary
csernoch.maria@inf.unideb.hu

ABSTRACT

Sprego is a programming approach in spreadsheet environments, which is firmly supported by unplugged and semi-unplugged tools for better comprehension. In the present paper we provide details of how authentic webpage-tables support real world problem solving and serve as semi-unplugged tools.

The "Computer Science Unplugged" method is generally accepted in communicating the great ideas of computer science. However, we have found that the variation and the right proportion of unplugged and semi-unplugged tools can serve these great ideas also satisfy students' eagerness for digital contents and tools. In this context we must clarify the misconception that both programming and end-user activities start when the computer is switched on. Beyond this, we provide details of our experience on how data management and end-user programming in spreadsheet environments can be introduced and supported with semi-unplugged tools.

Along with previously suggested ideas, we have found that teaching algorithms and programming in spreadsheet environments are both equally possible, but less demanding than 'classical' programming. To carry out coding in spreadsheets, we developed the Sprego – Spreadsheet Lego – high mathability programming approach and language. The semi-unplugged tools, which we rely on heavily in Sprego, are authentic data arranged in tables – primarily downloaded from the Internet and/or collected by students. The authentic tables allow both teachers and students to make a deep analysis of the content and structure of the data, to recognize data types, convert files, formulate tasks and problems based on the data, and finally to discuss and debug the problem solving processes and solutions. Most of these processes can be supported by providing the tables in printed form – either on paper or on digital devices –, prior to or alongside the coding process. One further feature of Sprego is the extensive use of composite functions. With this other semi-unplugged approach the discussion and debugging of solutions are strongly supported and it allows students to share the great ideas of computer sciences, and to develop computational thinking and algorithmic skills.

INTRODUCTION

> "Spreadsheets are everywhere in modern business, from the smallest organisations to the largest and most complex."
> (ICAEW [SCF], 2016)

It is wildly accepted that spreadsheets are pervasive, particularly among finance users (ICAEW [SCF], 2016); however, recent research in education has proved that (1) spreadsheets can be used in much wider contexts (Kadijevic, 2013; Angeli, 2013; Csernoch, 2017), (2) education and training can start from early ages in school (Csernoch, 2016; Biró & Csernoch, 2017a), and (3) they are programming interfaces (Hubwieser, 2004; Schneider, 2004, 2005; Zsakó, 2006; Elliott, 2007; Wakeling, 2007; Sestoft, 2011; Csernoch, 2014, 2017; Csernoch & Biró, 2015, 2017b).

Beyond the opportunities contained within spreadsheets, we are faced with several problems related to both their use and the way they are taught. The consequences of inappropriate spreadsheet use in finance is wildly researched and analyzed, due to the serious losses originating from erroneous spreadsheet documents ([Horror Stories], 2017; Panko, 2013, 2015; Panko & Port, 2013). One of the reasons for this failure is clearly stated in the Spreadsheet Competency Framework (SCF): "Spreadsheet skills are often learned ad hoc – almost two-thirds of Excel Community users are self-taught – and many users are unaware of their own true competency. Novices are generally overconfident; experts tend to sell themselves short." (ICAEW [SCF], 2016). Among several other reasons it has been shown (Panko, 2013, 2015; Panko & Port, 2013; Kadijevic 2013; Kruger & Dunning, 1999) that spreadsheet education is almost invisible to ICT professionals and computer sciences (CS) and/or thought to involve boring routine tasks in general educational contexts (Gove, 2012, 2014). These misconceptions would

explain the poorly developed and utilized spreadsheet documents and their negatives consequences. Recognizing the lack of spreadsheet competences, again in a financial context, the ICAEW have published the "Twenty principles for good spreadsheet practice", the second edition of it (ICAEW [20 Principles], 2016), and the "Spreadsheet competency framework. A structure for classifying spreadsheet ability in finance professionals." (ICAEW [SCF], 2016). These documents are to be warmly welcomed since they provide the bases for further research in spreadsheet competences.

However, we have found that the ICAEW documents support neither the programming aspect of spreadsheets nor the development of computational thinking (Wing, 2006). They rather focus on the tools within these frameworks and support the idea that typical spreadsheet users just want to use a power tool, with the craft coming later. This is the "classical" and wildly accepted approach to spreadsheets, where software is primarily taught with a technocentric and decontextualized focus (Angeli, 2013). In view of all these misconceptions, we have published the Edu-Edition of the Spreadsheet Competency Framework (E^2SCF, Csernoch & Biró, 2017c), where the focus is on the development of computational thinking, high-mathability computer problem solving, and knowledge transfer – in general, the future competences of the digital world (Davies et al., 2011).

Within the framework of the E^2SCF we and our colleagues have developed several unplugged (Csernoch, 2016; Csernoch & Biró, 2017a) and semi-unplugged tools (Csapó & Sebestyén, 2015) to introduce spreadsheet programming in schools. In the present paper we focus on several semi-unplugged tools to support real world problem solving in spreadsheet programming. These are primarily based on authentic tables whose sources are webpages containing tables or imitations of tables and/or the private collections of students/teachers.

AUTHENTIC TABLES FOR SPREADSHEET PROGRAMMING

As mentioned in the previous section, one of the main reasons for spreadsheet failures is a lack of motivation. Students are not interested in mindless and decontextualized data; they prefer contexts which are interesting and sources of real information retrieval. In the following we present webpages, their converted spreadsheet versions, tasks and games developed on these tables, and examples of the webpage→spreadsheet conversion (W→S).

One further advantage of using authentic tables as a data source is that this approach extensively supports knowledge transfer: (1) the boring act of typing is replaced by handling files of various formats, which is one of the crucial elements of ICT studies, (2) the normalization of the tables leads to basic database management knowledge, (3) the conversion process requires algorithmic skills, which are the bases for further studies in ICT and computer sciences, especially in programming, and (4) through the contexts of these tables a close link is built with other sciences and subjects.

TABLE EXAMPLES

One of students' favorite subjects is digital media, and within this framework they are interested in movies, music, and video games. We have found that they frequently use the IMDB (Internet Movie Database) charts, so we introduced one of these tables into our spreadsheet classes ([IMDB], 2013). However, the recently available webpage contains one fewer column (Figure 1, left), so we use an older version, downloaded in 2013 which is available in our collection of Sprego tables ([Sprego tables], 2017; Figure 1, right; Figure 8).

	A	B	C	D
1	Rank	Rating	Title	Votes
2	1	9,2	The Shawshank Redemption (1994)	790425
3	2	9,2	The Godfather (1972)	585045
4	3	9	The Godfather: Part II (1974)	371343
5	4	8,9	Pulp Fiction (1994)	621174
6	5	8,9	The Good, the Bad and the Ugly (1966)	244920
7	6	8,9	12 Angry Men (1957)	194680
8	7	8,9	Schindler's List (1993)	414133
9	8	8,8	The Dark Knight (2008)	736027
249	248	8	The Untouchables (1987)	120523
250	249	8	Papillon (1973)	42632
251	250	8	Castle in the Sky (1986)	38305

Figure 1: The IMDB Top Rated Movies webpage (retrieved in 2017, left) and its spreadsheet version (retrieved in 2013, right).

In recent years students have become interested in the subject of healthy and nutritious foods, weight gain and obesity, and so they like to see calorie tables (Figure 2, Figure 9). Another favorite source of data and information are local food webpages (Figure 14 – Figure 16).

	B	C
1	Categories/Foods	Calories
2	Apéritifs (pour 100 g)	
3	Apéricube La Vache qui rit (à l'unité)	16 calories
4	Bretzel (à l'unité 15 g)	80 calories
5	Cacahuètes (à l'unité)	10 calories
6	Chipster salé	460 calories
7	Chips Vico à l'ancienne saveur moutarde	549 calories
20	Rillettes d'oie	464 calories
21	Rillettes de saumon	320 calories
22	Boissons	
23	Bière canette = 33 cl	178 calories
701	Végétaline	900 calories
702	Féculents & Boulangerie (pour 100 g)	
703	Baguette (unité)	510 calories
772	Tarte (pommes)	300 calories
773	Triscottes	390 calories

Figure 2: The Les calories webpage (left) and its normalized spreadsheet version (right).

The most frequently searched and eagerly received contents include tables dealing with countries, and their statistical extremes (the largest, the highest, the smallest, etc.), data which is not obvious and/or is unique (complete lists of various subjects, war losses, expenses, life expectancy rates, etc.), as well as lottery tables, and various board games ([Sprego tables], 2017). In the present paper the "Countries of the World" and the "Word Heritage Sites" tables are introduced. The Countries table is retrievable from the Hungarian school leaving exams ([Countries], 2004; [Sprego tables], 2017; Figure 3), primarily prepared as a source for database management but serving our purposes just as well. Originally, it contains the name, the continent, the area, and the population (divided by thousand) of each country. In the modified version an index field was added to the table to clearly demonstrate the difference between the spreadsheet row numbers and the indexes of items of vectors. In our practice we refer to this semi-unplugged version as "countries_printed" – printed on screen, not on paper – (Figure 3), since we use it for manual data retrieval, building and checking algorithms, and debugging.

	A	B	C	D	E	F
1	index	Country	Continent	Capital	Area	Population (1000)
2	1	Afghanistan	Asia	Kabul	647500	27756
3	2	Albania	Europe	Tirana	28748	3545
4	3	Algeria	Africa	Algiers	2381740	32278
5	4	American Samoa	Oceania	Pago Pago	199	69
6	5	Andorra	Europe	Andorra la Vella	468	68
7	6	Angola	Africa	Luanda	1246700	10593
8	7	Anguilla	Amerika	The Valley	102	12
234	233	Yugoslavia	Europe	Belgrade	102350	10657
235	234	Zambia	Africa	Lusaka	752614	9959
236	235	Zimbabwe	Africa	Harare	390580	11377

Figure 3: The "countries_printed" semi-unplugged version of the Countries table.

In close connection with the Countries table, one of the most interesting tables is the World Heritage List, which contains data of heritage sites (Figure 4) in seven fields of various data types. Beyond its valuable content, the process of converting the table involves various elements of ICT (Figure 12), and as such supports knowledge transfer, which is one of our main concerns in relation to students' future competences (Davies et al., 2011).

Year	Name of the property	Country	Type	Region	Property (ha)	ID
1978	Aachen Cathedral	DE	C	EUR	0	3
1978	City of Quito	EC	C	LAC	320	2
1978	Galápagos Islands	EC	N	LAC	14,066,514	1
1978	Historic Centre of Kraków	PL	C	EUR	150	29
1978	Island of Gorée	SN	C	AFR	0	26
1978	L'Anse aux Meadows National Historic Site	CA	C	EUR	8,056	4
1978	Mesa Verde National Park	US	C	EUR	21,043	27
1978	Nahanni National Park #	CA	N	EUR	476,560	24
1978	Rock-Hewn Churches, Lalibela	ET	C	AFR	0	18
1978	Simien National Park	ET	N	AFR	13,600	9
1978	Wieliczka and Bochnia Royal Salt Mines	PL	C	EUR	1,105	32

	B	C	D	E	F	G
1	Name of the property	Country	Type	Region	Property (ha)	ID
2	Aachen Cathedral	DE	C	EUR	0	3
3	City of Quito	EC	C	LAC	320	2
4	Galápagos Islands	EC	N	LAC	14066514	1
5	Historic Centre of Kraków	PL	C	EUR	150	29
6	Island of Gorée	SN	C	AFR	0	26
7	L'Anse aux Meadows Nationa	CA	C	EUR	8056	4
8	Mesa Verde National Park	US	C	EUR	21043	27
9	Nahanni National Park #	CA	N	EUR	476560	24
10	Rock-Hewn Churches, Lalibela	ET	C	AFR	0	18
11	Simien National Park	ET	N	AFR	13600	9
12	Wieliczka and Bochnia Royal S	PL	C	EUR	1105	32
13	Yellowstone National Park	US	N	EUR	898349	28
14	Abu Mena	EG	C	ARB	183	90
1116	Western Tien-Shan *	UZ	N	APA	0	1490
1117	Zuojiang Huashan Rock Art Cu	CN	C	APA	6622	1508

Figure 4: The table of the World Heritage Site webpage (left) and its converted spreadsheet version (right).

Sports tables are also great favorites. For the present paper we selected an ATP table ([ATP], 2015; Figure 5 and Figure 6), which is generally accepted by both girls and boys. The content of the table offers a wide range of information retrieval tasks, deals with various data types, and the conversion process is extremely challenging and leads to webpage development and design. The details of the conversion are beyond the scope of the present paper; however, similar to the Heritage table, the question of the automated data type recognition in spreadsheets is mentioned in the corresponding section (Figure 13).

TOURNAMENT	DRAW	SURFACE	TOTAL FINANCIAL COMMITMENT	WINNER	RESULTS
Brisbane Brisbane, Australia **2015.01.04**	SGL 28 DBL 16	Outdoor Hard	$494,310	SGL: Roger Federer DBL: Jamie Murray , John Peers	RESULTS
Chennai Chennai, India **2015.01.05**	SGL 28 DBL 16	Outdoor Hard	$458,400	SGL: Stan Wawrinka DBL: Yen-Hsun Lu , Jonathan Marray	RESULTS

Figure 5: The table of the ATP Scores & Stats 2015 webpage.

	A	B	C	D	E	F	G	H	I	J	K
1	Title	Location	Date	Date	SGL	DBL	Surface1	Surface2	Commitment	Winner SGL	Winner DBL
2	Brisbane	Brisbane	Australia	2015.01.04	28	16	Outdoor	Hard	$494,310	Roger Federer	Jamie Murray John Peers
3	Chennai	Chennai	India	2015.01.05	28	16	Outdoor	Hard	$458,400	Stan Wawrinka	Yen-Hsun Lu Jonathan Marray
4	Doha	Doha	Qatar	2015.01.05	32	16	Outdoor	Hard	$1,221,320	David Ferrer	Juan Monaco Rafael Nadal
5	Sydney	Sydney	Australia	2015.01.11	28	16	Outdoor	Hard	$494,310	Viktor Troicki	Rohan Bopanna Daniel Nestor
6	Auckland	Auckland	New Zealand	2015.01.12	28	16	Outdoor	Hard	$519,395	Jiri Vesely	Raven Klaasen Leander Paes
7	Australian Open	Melbourne	Australia	2015.01.19	128	64	Outdoor	Hard	A$17,748,600	Novak Djokovic	Simone Bolelli Fabio Fognini
8	Montpellier	Montpellier	France	2015.02.02	28	16	Indoor	Hard	€494,310	Richard Gasquet	Marcus Daniell Artem Sitak
9	Quito	Quito	Ecuador	2015.02.02	28	16	Outdoor	Clay	$494,310	Victor Estrella Burgos	Gero Kretschmer Alexander Satschko
12	Rotterdam	Rotterdam	Netherlands	2015.02.09	32	16	Indoor	Hard	€1,600,855	Stan Wawrinka	Jean-Julien Rojer Horia Tecau
15	Marseille	Marseille	France	2015.02.16	28	16	Indoor	Hard	€632,840	Gilles Simon	Marin Draganja Henri Kontinen
16	Rio de Janeiro	Rio de Janeiro	Brazil	2015.02.16	32	16	Outdoor	Clay	$1,548,755	David Ferrer	Martin Klizan Philipp Oswald
64	Basel	Basel	Switzerland	2015.10.26	32	16	Indoor	Hard	€2,022,300	Roger Federer	Alexander Peya Bruno Soares
65	Valencia	Valencia	Spain	2015.10.26	28	16	Indoor	Hard	€604,155	Joao Sousa	Eric Butorac Scott Lipsky
66	ATP World Tour Masters	Paris	France	2015.11.02	48	24	Indoor	Hard	€3,830,295	Novak Djokovic	Ivan Dodig Marcelo Melo
67	ATP Finals	London	Great Britain	2015.11.15	8	8	Indoor	Hard	$7,000,000	Novak Djokovic	Jean-Julien Rojer Horia Tecau

Figure 6: The converted version of the ATP Scores & Stats 2015 table.

As mentioned above, local food outlets are popular among our students. The selected webpage is one of the leading pastry shops in our hometown so everyone in our close environment knows it ([Cakes], 2017). The webpage turned out to be one of our best resources since it is convertible either with spreadsheets or word processors and fundamental algorithms can be taught alongside the conversion processes (the conversion process is detailed in the "SEMI-AUTOMATED W→S CONVERSION" section of the present paper).

	A	B
1	Torta	Ár
2	Áfonya Hercegnő cukormentes tortája - KT057	3 950 Ft
3	Őrség zöld aranya torta ország tortája 2016 - KT058	3 950 Ft
4	Az ország tortája 2015 - KT050	3 950 Ft
5	Oreo keksz torta - KT052	3 950 Ft
6	Paleo Sacher torta - KT053	3 950 Ft
55	Francia vanília torta - KT055	4 200 Ft
56	Málnácska torta - KT056	3 800 Ft
57	Lekváros kardinális torta - KT044	3 500 Ft
58	Karamellás csoki mousse torta	3 800 Ft

Figure 7: The Cakes (Torták) webpage of a local pastry shop (left) and its converted version in a spreadsheet table (right).

TASKS AND GAMES

Our novel method for computer supported real world problem solving in spreadsheets is based on the effective and efficient methodologies used in other subjects, especially mathematics and programming, further developed to create an introductory end-user programming environment and approach, and was introduced as Sprego programming – Spreadsheet Lego – in 2014 (Csernoch, 2014). Consequently, in Sprego problem solving starts with data analysis. Since young students are much more competent with strings than numbers we have found that introductory tasks should be based on strings rather than on numerical problems, which is the "classical" approach to spreadsheets (ICAEW [SCF], 2016; Kalchman & Koedinger, 2005; Blanton & Kaput, 2011; Section TEACHING MATERIALS).

All the selected tables contain text fields but among them the IMDB serves as the best introductory table. In the IMDB table it is obvious that in the converted version of column C contains two data of different types, so they have to be separated and displayed in two columns. (The details of the solution are beyond the scope of the present paper; however, it is available in one of our previously published papers (Csernoch & Biró, 2016).) Similar tasks can be formulated either in the IMDB or in the other data tables. In the Cakes table (Figure 7) the Torta field has two types of data – the name and the product code –, although in some cases the code is missing. In the ATP table (Figure 6) the names of the doubles winners can be separated, while handling the currencies is a challenging problem, suitable for advanced users.

	A	B	C	D	E	F	G
1	Rank	Rating	Title	Votes			
2	1	9,2	The Shawshank Redemption (1994)	790425	1994	The Shawshank Redemption	2000
3	2	9,2	The Godfather (1972)	585045	1972	The Godfather	1994
4	3	9	The Godfather: Part II (1974)	371343	1974	The Godfather: Part II	
5	4	8,9	Pulp Fiction (1994)	621174	1994	Pulp Fiction	
6	5	8,9	The Good, the Bad and the Ugly (1966)	244920	1966	The Good, the Bad and the Ugly	
7	6	8,9	12 Angry Men (1957)	194680	1957	12 Angry Men	
8	7	8,9	Schindler's List (1993)	414133	1993	Schindler's List	
9	8	8,8	The Dark Knight (2008)	736027	2008	The Dark Knight	
249	248	8	The Untouchables (1987)	120523	1987	The Untouchables	
250	249	8	Papillon (1973)	42632	1973	Papillon	
251	250	8	Castle in the Sky (1986)	38305	1986	Castle in the Sky	
252							
253	The number of movies released between 1994 and 2000 is 36.				The Untouchables		
254	The Untouchables was released in 1987.						

Figure 8: The normalized IMDB table and two tasks based on the new data table. Task 1: The number of movies released between two years. The years are the input values of the task, provided by the user in cells G2 and G3 and the output is a sentence with the two years and the calculated number. Task 2: A title is selected from the combobox and the formula calculates the year and displays it in a whole sentence.

Once these data are separated – the titles and the years in the IMDB table –, in addition to string problems, all the

converted tables can be used for counting and searching problems (Csernoch & Biró, 2016). Two of these tasks are introduced in Figure 8 in rows 253 and 254. These two problems are rather for intermediate users; consequently, for novices simpler, and for advanced users, more complex tasks can be formulated.

The Calories table – retrieved from the Les calories webpage – holds two columns of data. In the first column the categories and the foods are listed, while in the second we have an empty string and calories, respectively (Figure 2, categories are highlighted in green). With these data we can create a spreadsheet calories calculator (Figure 9). The calculator uses categories as major items. Each category contains a combobox and a slide bar. A combobox lists all the foods in the respective category derived from the data table. The slide bar is an additional object to each category, where we can select the amount of food. Based on these data the calculator displays the calories for each selected food, adds them, and displays the sumproduct in cell I774. One further feature of the calculator is the ability to set a limit. If the sumproduct, based on the calories of the selected items and the amount, exceeds the limit, the cell changes its color.

	E	F	H	I	J	M	N
773							
774				1809		Limit:	1500
775		Apéritifs (pour 100 g)	185	0	Mousse de poisson	0	
776		Boissons	71	284	Orangina verre = 15 cl	4	
777		Desserts et sucreries (pour 100 g)	526	526	Bonbons M&M'S	1	
778		Coquillages et crustacés (pour 100 g)	103	0	Crabe (conserve)	0	
779		Poissons (pour 100 g)	77	0	Daurade	0	
780		Viandes (pour 100 g)	193	386	Agneau (langue)	2	
781		Volailles (pour 100 g)	350	0	Oie	0	
782		Légumes (pour 100 g)	38	228	Carotte	6	
783		Fruits	77	385	Cerises	5	
784		Produits laitiers (pour 100 g)	353	0	Gouda	0	
785		Matières grasses (pour 100 g)	900	0	Olives (huile)	0	
786		Féculents & Boulangerie (pour 100 g)	230	0	Pain complet	0	

Figure 9: A spreadsheet calories calculator, based on the table of the Les-calories webpage.

The "country-capital quiz" can be used in geography classes but its creation requires several fundamental programming algorithms which are better handled in ICT and/or CS classes. Based on the original "searching in vector problem" several different tasks can be formulated and introduced in classes. In the present paper one of the most advanced versions is presented, where various spreadsheet tools are applied: a combobox for selecting the country, an IF() function for varying the output text in cell H4, a formula to calculate the capital of the selected country (various versions of the MATCH(), INDEX(), and INDEX(MATCH()) functions), a concatenation operator or function for creating the sentence of strings and calculated values, and conditional formatting to color the content of cell H4 according to the user's answer in H6.

	H
1	
2	Germany
3	
4	
5	
6	

	H	I
1		
2	Germany	
3		
4	The capital city of Germany is Berlin.	
5		
6	Berlin	

	H
1	
2	Germany
3	
4	Ajjaj!
5	
6	Budapest

Figure 10: A "country-capital quiz" based on the Countries table. The country selected from the combobox (left), text (capital city) is typed in H6. If the answer is correct a whole sentence is displayed in green; if not, an error message in red is displayed in cell H4; if H6 is empty, H4 is also left empty.

SEMI-AUTOMATED W→S CONVERSION

The conversion process of the webpages might seem a one-step process since spreadsheets open webpages. However, we have experienced in several cases that these tables are manipulated: neither the table nor the paragraph structure is clear. Beyond these hidden errors, it is obvious that the differences in the syntax and in the language setup of spreadsheets and operating systems, along with the automated data recognition, might result in serious data losses (e.g. see Figure 12 and Figure 13).

A version independent conversion process officially requires nothing more than the opening of the webpage in a spreadsheet and then that it is saved in spreadsheet format. However, this method only works in theory, since the additional objects, formats, and the automated data type recognition of spreadsheets might result in a chaotic table and/or data losses, respectively (Figure 12 and Figure 13). To get rid of the additional objects and formats a text file would serve best. However, in several cases we have to apply more software which can create the text file but does not change the content (Figure 11). Word processing programs were found to be the most convenient for our purposes. Considering all these points, the major steps of the conversion process are mapped in Figure 11. We are aware of the "user friendly" but limited or less-limited conversion options of the different spreadsheets versions, but our goal is to present a less software-dependent, general purpose algorithm of the process.

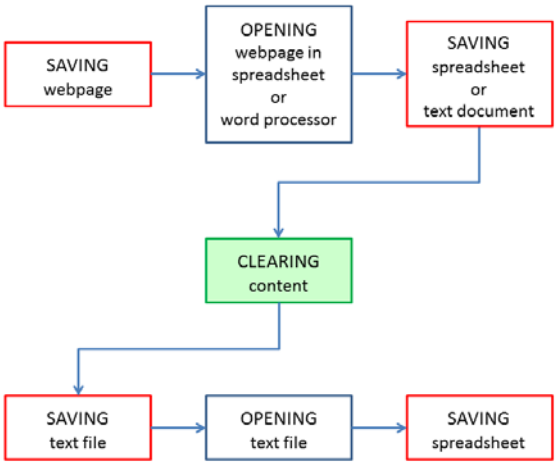


Figure 11: The major steps of the webpage→spreadsheet (W→S) conversion process.

As is shown in Figure 11, the algorithm of the conversion process is primarily concerned with handling different file types with the Open and the Save As commands. However, the "Clearing content" step is unique to each document and in most cases requires well developed algorithmic skills. The complexity of the document and the algorithm on which the webpage is built formulate the algorithm. In several cases the clearing phase is broken down into smaller units, and can be inserted between any of the [opening, saving] blocks and repeated various times.

This method of the conversion is semi-automated in the sense that the major commands are carried out manually, while each step is automated and as such can be applied to a clearly defined section of the data. In word processors the clearing is primarily carried out with the Replace command (Figure 12) and/or with the table→tabulator→table conversions (Figure 15 and Figure 16), while in spreadsheets formulas have to be created to handle the data conversion (for further details see the "*Conversion with spreadsheet formulas"* section of the present paper; Csernoch, 2014).

Handling thousand separator and decimal characters

F	G
Property (ha)	ID
0	3
320	2
14,066,514	1
150	29
0	26
8,056	4
21,043	27
476,560	24
0	18
13,600	9
1,105	32
898,349	28
183	90
0	1490
6,622	1508

F	G	H
Property (ha)	ID	
0	3	0
320	2	320000
14,066,514	1	#ÉRTÉK!
150	29	150000
0	26	0
8,056	4	8056
21,043	27	21043
476,56	24	476560
0	18	0
13,6	9	13600
1,105	32	1105
898,349	28	898349
183	90	183000
0	1490	0
6,622	1508	6622

F	G
Property (ha)	ID
0	3
320	2
14066514	1
150	29
0	26
8056	4
21043	27
476560	24
0	18
13600	9
1105	32
898349	28
183	90
0	1490
6622	1508

Figure 12: The World Heritage page is opened in English (left), opened/imported in European (middle) spreadsheets without intervention, and opened and cleared in a word processor – the commas were removed with a Replace command (right).

In Figure 12 in an English spreadsheet all the small numbers – less than 1000 – are recognized as numbers, while the others with commas are converted to strings but no data are lost. In the European version the numbers with one comma are divided by 1000 while the others are converted to strings, and the small numbers are converted properly to numbers. The English version can be handled in spreadsheets since the commas can be removed either by the Replace command or the SUBSTITUTE() function and then the data can be converted into numbers (Csernoch, 2014; Csernoch & Biró, 2016). Due to the loss of data, the European version cannot be handled in spreadsheets without first creating the text file in a word processor. By opening the table in a word processor in the Property column the commas can be replaced with the empty string. Following the replacement the table has to be converted into a text file, which can be opened in spreadsheets. While creating the text file we have to take into account the fact that the table contains several non-ASCII characters; consequently, the encoding has to be changed to Unicode.

	A	B	C	D	E	F	G	H	I	J	K
1	Title	Location	Date	Date	SGL	DBL	Surface1	Surface2	Commitment	Winner SGL	Winner DBL
2	Brisbane	Brisbane	Australia	2015.01.04	28	16	Outdoor	Hard	$494,310	Roger Federer	Jamie Murray John Peers
3	Chennai	Chennai	India	2015.01.05	28	16	Outdoor	Hard	$458,400	Stan Wawrinka	Yen-Hsun Lu Jonathan Marray
4	Doha	Doha	Qatar	2015.01.05	32	16	Outdoor	Hard	$1,221,320	David Ferrer	Juan Monaco Rafael Nadal
5	Sydney	Sydney	Australia	2015.01.11	28	16	Outdoor	Hard	$494,310	Viktor Troicki	Rohan Bopanna Daniel Nestor
6	Auckland	Auckland	New Zealand	2015.01.12	28	16	Outdoor	Hard	$519,395	Jiri Vesely	Raven Klaasen Leander Paes
7	Australian Open	Melbourne	Australia	2015.01.19	128	64	Outdoor	Hard	A$17,748,600	Novak Djokovic	Simone Bolelli Fabio Fognini
8	Montpellier	Montpellier	France	2015.02.02	28	16	Indoor	Hard	€ 494,31	Richard Gasquet	Marcus Daniell Artem Sitak
9	Quito	Quito	Ecuador	2015.02.02	28	16	Outdoor	Clay	$494,310	Victor Estrella Burgos	Gero Kretschmer Alexander Satschko
12	Rotterdam	Rotterdam	Netherlands	2015.02.09	32	16	Indoor	Hard	€1,600,855	Stan Wawrinka	Jean-Julien Rojer Horia Tecau
15	Marseille	Marseille	France	2015.02.16	28	16	Indoor	Hard	€ 632,84	Gilles Simon	Marin Draganja Henri Kontinen
16	Rio de Janeiro	Rio de Janeiro	Brazil	2015.02.16	32	16	Outdoor	Clay	$1,548,755	David Ferrer	Martin Klizan Philipp Oswald
64	Basel	Basel	Switzerland	2015.10.26	32	16	Indoor	Hard	€2,022,300	Roger Federer	Alexander Peya Bruno Soares
65	Valencia	Valencia	Spain	2015.10.26	28	16	Indoor	Hard	€ 604,16	Joao Sousa	Eric Butorac Scott Lipsky
66	ATP World Tour Masters	Paris	France	2015.11.02	48	24	Indoor	Hard	€3,830,295	Novak Djokovic	Ivan Dodig Marcelo Melo
67	ATP Finals	London	Great Britain	2015.11.15	8	8	Indoor	Hard	$7,000,000	Novak Djokovic	Jean-Julien Rojer Horia Tecau

Figure 13: The ATP text file opened in a spreadsheet allows automated data type recognition. In column I the conversion of the currency leads to a loss of data. In cells I8, I15, and I65 the original values are replaced by their 1000th; beyond this, in I65 the value is rounded.

To stop the automated data type conversion we have to open the text file in a spreadsheet, and in the "Text Import Wizard" we can change the General data type to Text (Step 3). With this option all the data in the selected column is converted to text (Figure 6).

Conversion with spreadsheet formulas

The conversion of the Cakes webpage ([Cakes], 2017; Figure 14) can be carried out both in spreadsheets and word processors. Both of them have their own algorithms, and the choice made is based on the pedagogical aims of the task.

	A
1	Áfonya Hercegnő cukormentes tortája - KT057
2	3.950 Ft
3	A termék adatai »
4	RENDELHETŐ
5	
6	Őrség zöld aranya torta ország tortája 2016 - KT058
7	3.950 Ft
8	A termék adatai »
9	RENDELHETŐ
10	
11	Az ország tortája 2015 - KT050
12	3.950 Ft
13	A termék adatai »
14	RENDELHETŐ
15	

	B	C	D
1	Cake	Price1	Price2
2	Áfonya Hercegnő cukormentes tortája - KT057	3.950 Ft	3 950 Ft
3	Őrség zöld aranya torta ország tortája 2016 - KT058	3.950 Ft	3 950 Ft
4	Az ország tortája 2015 - KT050	3.950 Ft	3 950 Ft
5	Oreo keksz torta - KT052	3.950 Ft	3 950 Ft
6	Paleo Sacher torta - KT053	3.950 Ft	3 950 Ft
7	Paleo Bounty torta - KT054	3.950 Ft	3 950 Ft
8	Nosztalgia krémes torta - KT055	3.800 Ft	3 800 Ft
9	Kinder mousse torta csoki kellékekkel - KT056	5.000 Ft	5 000 Ft
10	Vaníliás eper mousse torta	3.800 Ft	3 800 Ft
11	Narancsos karamell mousse torta	3.800 Ft	3 800 Ft
12	Szeretlek torta, epres brownie - KT004	3.800 Ft	3 800 Ft
13	Írkávé mousse torta - KT005	3.800 Ft	3 800 Ft
57	Lekváros kardinális torta - KT044	3.500 Ft	3 500 Ft
58	Karamellás csoki mousse torta	3.800 Ft	3 800 Ft

Figure 14: The Cakes webpage opened and cleared in a word processor (left) and converted into a normalized table with spreadsheet functions (right). The version shown has three columns: column B: cake and code (optional) – data type: string –, column C: price – data type: string –, column D: price – data type: number –, and the non-significant data are removed – strings: "A termék adatai »", "RENDELHETŐ", and the empty string.

All the conversions start with data analysis, as follows (Figure 14).

The features of the spreadsheet table:

- each cake occupies 5 rows
- the 1st row of the block holds the name of the cake (for advanced users: occasionally, the code of the cake), data type: string
- the 2nd row of the block holds the price of the cake, data type: originally string but should be converted to number (currency)
- the other rows can be ignored
- the original position of the cakes and prices can be calculated from the row number of the new position
- all the prices contain a dot, a "SpaceFt" string, and a Space character at the end of the string
- at the end of the names there is a Space character

The algorithm of the spreadsheet conversion:

- calculating the row number of the new position: n
- calculating the original position of the cakes/prices: $(n–2)\times5+1$ or $(n–2)\times5+2$
- displaying the cakes/prices
- converting the prices to numbers
  - calculating the length of the string
  - calculating the length of the number
  - displaying the numbers from the left of the string
  - substituting the dot with the empty string
  - converting the string to number
- formatting the currency

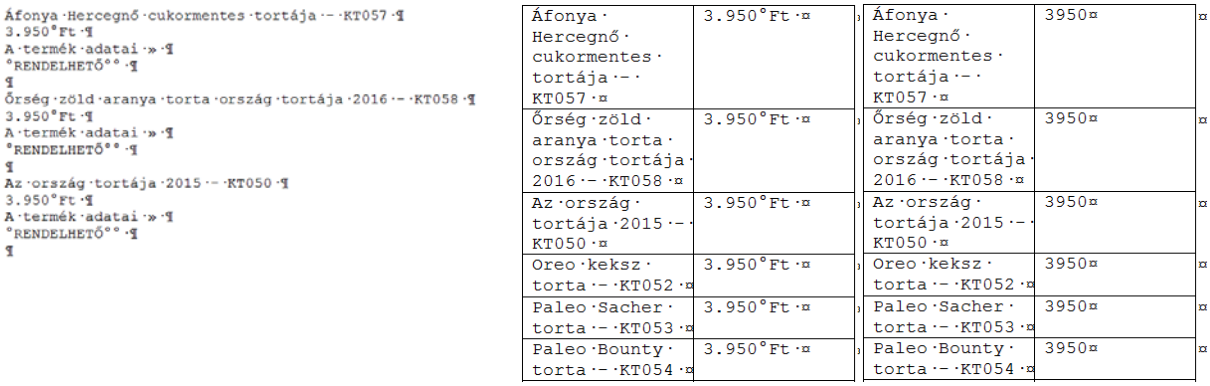
The coding of the spreadsheet conversion with Sprego:

- {=ROW()}
- {=(ROW()-2)*5+1}
- {=INDEX(A2:A284,(ROW()-2)*5+1)}

The coding of the string→number conversion with Sprego:

- {=LEN(C2:C58)}
- {=LEN(C2:C58)-4}
- {=LEFT(C2:C58,LEN(C2:C58)-4)}
- {=SUBSTITUTE(LEFT(C2:C58,LEN(C2:C58)-4),".","")}
- {=SUBSTITUTE(LEFT(C2:C58,LEN(C2:C58)-4),".","")*1}

Conversion with a word processor



Áfonya·Hercegnő·cukormentes·tortája·–·KT057·¶
3.950°Ft·¶
A·termék·adatai·»·¶
°RENDELHETŐ°°·¶
¶
Őrség·zöld·aranya·torta·ország·tortája·2016·–·KT058·¶
3.950°Ft·¶
A·termék·adatai·»·¶
°RENDELHETŐ°°·¶
¶
Az·ország·tortája·2015·–·KT050·¶
3.950°Ft·¶
A·termék·adatai·»·¶
°RENDELHETŐ°°·¶
¶

Áfonya·Hercegnő·cukormentes·tortája·–·KT057·¤	3.950°Ft·¤	Áfonya·Hercegnő·cukormentes·tortája·–·KT057·¤	3950¤
Őrség·zöld·aranya·torta·ország·tortája·2016·–·KT058·¤	3.950°Ft·¤	Őrség·zöld·aranya·torta·ország·tortája·2016·–·KT058·¤	3950¤
Az·ország·tortája·2015·–·KT050·¤	3.950°Ft·¤	Az·ország·tortája·2015·–·KT050·¤	3950¤
Oreo·keksz·torta·–·KT052·¤	3.950°Ft·¤	Oreo·keksz·torta·–·KT052·¤	3950¤
Paleo·Sacher·torta·–·KT053·¤	3.950°Ft·¤	Paleo·Sacher·torta·–·KT053·¤	3950¤
Paleo·Bounty·torta·–·KT054·¤	3.950°Ft·¤	Paleo·Bounty·torta·–·KT054·¤	3950¤

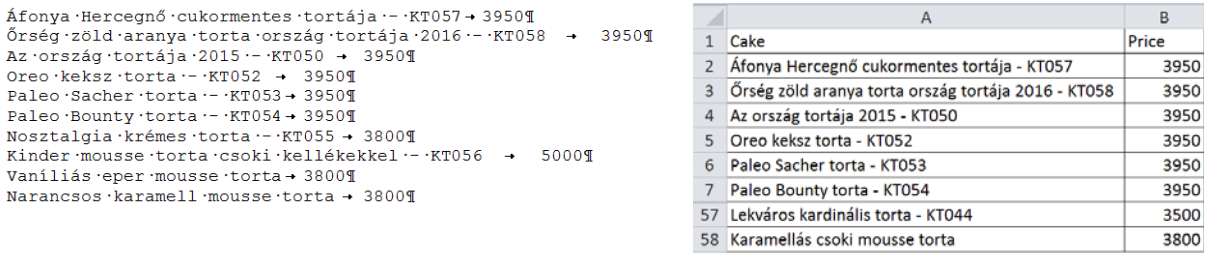
Figure 15: The Cakes table opened in a word processor (left), the 5-line blocks converted into a 5-column table, in which 3 columns are deleted (middle), and the characters removed from the numbers (right)

The algorithm of the word processor conversion:

- the text (Figure 15, left) is converted into a 5-column table
- 3 columns with the unnecessary data are deleted (Figure 15, middle)
- dots are removed from the numbers
- the "nonbrakingSpaceFtSpace" string is removed (Figure 15, right)

The coding of the string→number conversion in word processor:

- From Text to Table, the number of columns set to 5, Separator character: ¶
- 3 columns are deleted (Figure 15, middle)
- the column of numbers is selected and dots are replaced with nothing
- the column of numbers is selected and the "non-braking-SpaceFtSpace" strings are replaced with nothing (Figure 15, right)
- From Table to Text, the separator character is Tab
- the "SpaceTab" string in converted to Tab (Figure 16, left)
- Save as text file, code: Unicode (UTF-8)



Áfonya·Hercegnő·cukormentes·tortája·–·KT057→3950¶
Őrség·zöld·aranya·torta·ország·tortája·2016·–·KT058 → 3950¶
Az·ország·tortája·2015·–·KT050 → 3950¶
Oreo·keksz·torta·–·KT052 → 3950¶
Paleo·Sacher·torta·–·KT053→3950¶
Paleo·Bounty·torta·–·KT054→3950¶
Nosztalgia·krémes·torta·–·KT055→3800¶
Kinder·mousse·torta·csoki·kellékekkel·–·KT056 → 5000¶
Vaníliás·eper·mousse·torta→3800¶
Narancsos·karamell·mousse·torta→3800¶

	A	B
1	Cake	Price
2	Áfonya Hercegnő cukormentes tortája - KT057	3950
3	Őrség zöld aranya torta ország tortája 2016 - KT058	3950
4	Az ország tortája 2015 - KT050	3950
5	Oreo keksz torta - KT052	3950
6	Paleo Sacher torta - KT053	3950
7	Paleo Bounty torta - KT054	3950
57	Lekváros kardinális torta - KT044	3500
58	Karamellás csoki mousse torta	3800

Figure 16: The two data of the Cakes table separated with a Tab character (left); the converted table opened in a spreadsheet (right)

CONCLUSIONS

Our present paper offers and refers to various semi-unplugged tools introduced in our classroom practice to support Sprego programming. Sprego – Spreadsheet Lego – is a programming approach in spreadsheet environments, using those tools available in user-friendly interfaces which support programming, algorithm building, the development of computational thinking, computer supported problem solving, and knowledge transfer, all of which are the competences of the future.

One of the primary features of Sprego is that it focuses on real world problem solving and as such requires various authentic contents organized into tables. The main source of the tables is the Internet or the private collections of students and/or teachers. In the present paper we have provided examples of tables retrieved from webpages which were found to be interesting and motivating for students. The most popular tables contain data relating to extremes, local food outlets, board games, and sports, but any special data table would be appropriate for Sprego if it contains real world data and students are interested in it.

In addition to presenting the tables, we dealt with their conversion, the advantages and disadvantages of the

automated data type recognition of spreadsheets, and how we can handle these problems. A conversion of a webpage-table is detailed, where both the spreadsheet and the word processor solutions presented. However, we must emphasize that the form in which a table is presented in the class plays a fundamental role in teaching end-user programming and algorithms. It is always part of a teacher's competence to select the version of the table which best suits the students' interests and the pedagogical aims of the classes.

REFERENCES

Angeli, C. (2013). Teaching Spreadsheets: A TPCK Perspective. In *Improving Computer Science Education.* (Eds.) D. M. Kadijevich, C. Angeli, and C. Schulte. Routledge.

EuSpRIG Horror Stories (2017). Retrieved 09.07.2017 from http://eusprig.org/horror-stories.htm.

Blanton, M. L. & Kaput, J. J. (2011). *Functional Thinking as a Route Into Algebra in the Elementary Grades.* Early Algebraization, Advances in Mathematics Education, J. Cai, E. Knuth (eds.), DOI 10.1007/978-3-642-17735-4_2, Springer-Verlag Berlin Heidelberg.

Csapó, G. & Sebestyén, K. (2015). *Oktatóprogram a Sprego táblázatkezelő módszerhez*. In: Szlávi Péter, Zsakó László (eds.) INFODIDACT 2015. Zamárdi, Webdidaktika Alapítvány, Budapest 2015. pp. 1-18.

Csernoch, M. (2014). *Programming with Spreadsheet Functions: Sprego*. In Hungarian: Programozás táblázatkezelő függvényekkel – Sprego. Budapest, Műszaki Könyvkiadó.

Csernoch, M. (2016). *Sprego – funkcionális programozás oktatása általános és középiskolában*. Kihívások az oktatásban: demográfia, digitalizáció, tartalomfejlesztés, munkavállalói kompetenciák. XVIII. Országos Közoktatási Szakértői Konferencia. HAJDÚSZOBOSZLÓ, 11-13 October 2016, pp. 377–394.

Csernoch, M. (2017). *Thinking Fast and Slow in Computer Problem Solving*, Journal of Software Engineering and Applications. Vol.10 No. 01(2017), Article ID:73749, 30 pages 10.4236/jsea.2017.101002

Csernoch, M. & Balogh, L. (2010). *Algorithms and Spreadsheet-management – Talent Support In Education In The Field Of Informatics*. In Hungarian: Algoritmusok és táblázatkezelés – Tehetséggondozás a közoktatásban az informatika területén. Association of Hungarian Talent Support Organizations, in Hungarian, Magyar Tehetségsegítő Szervezetek Szövetsége, Budapest. ISSN: 2062-5936 http://tehetseg.hu/sites/default/files/16_kotet_net_color.pdf, accessed 15-July-2014.

Csernoch, M. & Biró, P. (2015). *Sprego Programming, Spreadsheets in Education* (eJSiE), 8(1). Retrieved July 21, 2016, from http://epublications.bond.edu.au/ejsie/vol8/iss1/4.

Csernoch, M. & Biró, P. (2016). *Introduction to Classroom Sprego*. ACTA DIDACTICA NAPOCENSIA 9:(1) pp. 1-14. (2016).

Csernoch, M. & Biró, P. (2017a). *Unplugged tools for building algorithms with Sprego*. In: Mafalda Carmo (szerk.) Education and New Developments 2017. 401 p. Lisbon, Portugal, InScience Press, 2017. pp. 401-405.

Csernoch, M. & Biró, P. (2017b). *Sprego, End-user Programming in Spreadsheets*, PPIG 2017 - 28th Annual Workshop. 1 Jul 2017 to 3 Jul 2017. Delft, Netherlands (accepted)

Csernoch, M. & Biró, P. (2017c). *Teaching methods are erroneous: approaches which lead to erroneous end-user computing.* London, Five Star Printing Ltd., 2017. 14 p. (Proceedings of the EuSpRIG 2016 Conference)

Davies, A., Fidler, D., & Gorbis, M. (2011). *Future Work Skills 2020.* Institute for the Future for the University of Phoenix Research Institute. Palo Alto, CA, USA. Retrieved June 20, 2017 from http://www.iftf.org/uploads/media/SR-1382A_UPRI_future_work_skills_sm.pdf

Elliott, C. M. (2007). *Tangible Functional Programming*. ICFP'07, October 1–3, 2007, Freiburg, Germany. http://conal.net/papers/Eros/eros.pdf, accessed 15-June-2014.

Gove, M. (2012). *Michael Gove speech at the BETT Show 2012*. Published 13 January 2012. https://www.gov.uk/government/speeches/michael-govespeech-at-the-bett-show-2012, accessed 15-June-2014.

Gove, M. (2014). *Michael Gove speaks about computing and education technology*. Published 22 January 2014. https://www.gov.uk/government/speeches/michael-gove-speaks-aboutcomputing-and-education-technology, accessed 15-June-2014.

Horror Stories (2017). *EuSpRIG Horror Stories*. Retrieved July 08, 2017 from http://eusprig.org/horror-stories.htm

Hubwieser, P. (2004). *Functional Modelling in Secondary Schools Using Spreadsheets*, Education and Information Technologies, 9(2), 175–183. DOI= http://doi.org/10.1023/B:EAIT.0000027929.91773.ab.

ICAEW [SCF] (2016). *Spreadsheet competency framework* (2016)

ICAEW [20 Principles] (2016). *Twenty principles for good spreadsheet practice*. Second edition. Retrieved 09.07.2017 from http://www.icaew.com/~/media/corporate/files/technical/information%20technology/excel%20community/166%20twenty%20principles%20for%20good%20spreadsheet%20practice.ashx.

Kadijevich, D. (2013). *Learning about spreadsheet*, In Kadijevich, D., Angeli, C., and Schulte, C. (Eds.). (2013). Improving Computer Science Education. (pp. 19–33) New York: Routledge.

Kalchman, M. & Koedinger, K R. (2005). *Teaching and Learning Functions. How Students Learn: History, Mathematics, and Science in the Classroom*. Committee on How People Learn, A Targeted Report for Teachers. M. Suzanne Donovan and John D. Bransford, Editors. Division of Behavioral and Social Sciences and Education. NATIONAL RESEARCH COUNCIL OF THE NATIONAL ACADEMIES. THE NATIONAL ACADEMIES PRESS. Washington, D.C.

Kruger, J. & Dunning, D. (1999). *Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments*. Journal of Personality and Social Psychology 77 (6): pp. 1121–34.

Panko, R. (2013). *The Cognitive Science of Spreadsheet Errors: Why Thinking is Bad*, Proceedings of the 46th Hawaii International Conference on System Sciences, January 7-10, 2013, Maui, Hawaii.

Panko, R. (2015). *What We Don't Know About Spreadsheet Errors Today: The Facts, Why We Don't Believe Them, and What We Need to Do*, arXiv:1602.02601 [cs]. Retrieved July 21, 2016, from http://arxiv.org/abs/1602.02601.

Panko, R. & Port, D. (2013). *End User Computing: The Dark Matter (and Dark Energy) of Corporate It*, Journal of Organizational and End User Computing, 25(3), pp. 1–19.

Schneider, M. (2004). *An Empirical Study of Introductory Lectures in Informatics Based on Fundamental Concepts*, In Informatics and Students Assessment, Lecture Notes in Informatics 1, pp. 123–133.

Schneider, M. (2005). *A Strategy to Introduce Functional Data Modeling at School Informatics*, In R. T. Mittermeir (Ed.), From Computer Literacy to Informatics Fundamentals (pp. 130–144). Berlin Heidelberg, Germany: Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-31958-0_16.

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

Wakeling, D. (2007). *Spreadsheet functional programming*. JFP 17(1), pp. 131–143, 2007. Cambridge University Press.

Wing, J. M. (2006). *Computational thinking*, Communications of the ACM, 49(3), pp. 33–35. DOI= http://doi.org/10.1145/1118178.1118215.

Zsakó, L. (2006). *Combinatorics – Competition – Excel*. Teaching Mathematics and Computer Science, 4/2 (2006), pp. 427–435.

SOURCES

[ATP] (2015). ATP Scores & Stats 2015. Retrieved June 20, 2017 from http://www.atpworldtour.com/en/scores/results-archive?year=2015

[Cakes] (2017). Torták. Kismandula Cukrászda. Retrieved June 20, 2017 from http://www.kismandula.hu/klasszikus-tortak/tortak-2

[Calories] (2017). Les Calories. Retrieved June 20, 2017 from http://www.les-calories.fr/

[Countries] (2004). The Countries of the World. Retrieved June 20, 2017 from https://www.oktatas.hu/bin/content/dload/erettsegi/probaerettsegi_2004/info_em_flap.zip

[IMDB] (2013). IMDB Top Rated Movies. Retrieved January 20, 2013 from http://www.imdb.com/chart/top.

[Sprego tables] (2017). Retrieved June 20, 2017 from https://arato.inf.unideb.hu/csernoch.maria/Sprego/sprego_tables.xlsx

[Heritage] (2017). World Heritage List. Retrieved June 20, 2017 from http://whc.unesco.org/?cid=31&mode=table

TEACHING MATERIALS

Cours Excel: formules de calculs et fonctions. Retrieved July 21, 2016, from http://www.excel-pratique.com/fr/cours/excel_formules_calculs_fonctions.php.

Dancsó, T. and Korom, P. (2013). Informatics for Grades 9–10. In Hungarian: Informatika 9-10. a gimnáziumok számára. Nemzedékek Tudása Tankönyvkiadó (Oktatáskutató és Fejlesztő Intézet), Budapest.

ECDL / ICDL Sample Test Papers (Windows 7 2010). Retrieved July 21, 2016, from http://www.ecdl.org/programmes/media/ECDL_ICDL_5.0_Sample_Part_Tests_Win_7_20101.zip.

Fishel, S. (2013). Excel 2013 Core. Intro. Retrieved July 21, 2016, from http://nova.saisd.net/getinfo/assets/excel-2013-core-pt-1.pdf.

Florida Gulf Coast University 2013. Excel 2013 Tutorial. Retrieved July 21, 2016, from http://www.fgcu.edu/Support/excel2013.html.

Gross, D., Akaiwa, F., and Nordquist, K. (2014). Succeeding in Business with Microsoft Excel 2013: A Problem-Solving Approach, Cengage Learning, US.

Hunger, L. (2010). Excel 2010 Basis. An Beispielen lernen. Mit Aufgaben üben. Durch Test-fragen Wissen überprüfen. Retrieved July 21, 2016, from http://www.ktn.gv.at.

Walkenbach, J. (2010). Excel 2010 Bible. Retrieved June 15, 2016 from http://www.seu.ac.lk/cedpl/student%20download/Excel%202010%20Bible.pdf.