Mathability in Business Education

Viktor László Takács[1] **(orcid 0000-0001-8433-6115)**,
Katalin Bubnó[2] **(orcid 0000-0002-5445-2374)**

[1] Department of Business Informatics, University of Debrecen, Böszörményi u. 138, 4032 Debrecen, Hungary; takacs.viktor@econ.unideb.hu

[2] Doctoral School of Mathematical and Computational Sciences, University and National Library, University of Debrecen, Egyetem tér 1, 4032 Debrecen, Hungary; kbubno@lib.unideb.hu

Abstract: Data Analyst or Data Scientist is one of the most sought-after professions. In tertiary education, the most common way for someone to become a Data Analyst is to learn Business and, later, if someone has some computational skills they can learn Computer Science as well. With the two degrees, they can solve real-life problems in their job. But there are a lot of necessary things they had never studied before because the basic conceptual methodology and the mindset were not based on any of the two curricula. Our aim is to develop a method in introductory business education to lay down the bases of the right-thinking towards data sciences for undergraduate students in Economics. If they are interested in the topic, they can orient themselves towards data sciences during their further studies. In this paper, we present a methodology to help Economics and Business Students to understand the main business questions, formalize the questions correctly, find computer-based solutions, and discuss/debug the results. Furthermore, students must learn basic data transformations and data enrichment methods as well, which is the primary feature of high-mathability problem-solving approaches. The tool we use is Microsoft Excel which contains OLAP (On-Line Analytical Processes) elements.

Keywords: OLAP; data science; formalization; management questions; computer problem-solving; high-mathability teaching approaches; knowledge transfer; spreadsheets

1 Introduction: Computational Thinking and Data Science

We aimed to develop three critical skills for undergraduate students in economics so that, if they are interested in the topic, they could orient themselves towards data science during their further studies. The three skills are problem-solving, analytical thinking, and system approach. These are all part of computer thinking [9]. The term Computational Thinking was first used by Seymour Paper [5], but it has become an important research area for IT methodology over the past ten years [4]. Many definitions have been created, reinterpreted, and explained, but everyone agrees that computational thinking should not only be a matter for IT professionals, but a key competence for everyone.

Our first and most important skill is problem-solving, without this, the data scientist cannot be good in his profession. This skill is not technology-dependent, problem solving is developed mainly in mathematics classes in public education. Typical tools are mathematical word problems.

In a previous study [7], we discovered some analogies between mathematical problem-solving and the process of computer programming based on Pólya's problem-solving method [6]. We worked out a teaching methodology introducing computer programming. We had early developing courses in a primary school, and later we created an experiment in a secondary school [7, 10], and we presented some cognitive aspects of our experiments and results [26] based on Ambrus's mathematical psychological research [27]. Meanwhile, we start to teach Business Informatics for Economics and Business Students. We recognized the similarities to word problems and we adapted Pólya's method to solve business questions as domain-specific word problems with OLAP (On-Line Analytical Processing) technology.

Based on Pólya's problem-solving method [6] a Data Analyst must go through some steps:

- Understanding the problem,
- Devising a plan,
- Carrying out the plan,
- Discussion.

In this paper, our goal is to show an introductory method and tools to help Economics and Business Students to understand usual business questions, formalize the questions correctly, solve and discuss/debug the result. The truth is that the more serious discussion will be the topic of further courses (i.e. Information systems or Risk analysis).

The second skill to be developed is analytical thinking, while the third is the system approach. Developing the right system approach is difficult because students think that they have to work with prepared data tables and they only have to master the technical processing. We believe that students coming to universities should be taught a completely new approach to informatics, instead of teaching the office user knowledge that they should have acquired in high school, we can base their confident technical knowledge and system approach. Unfortunately, high school data processing skills can even cause misconceptions for students. There are no large amount of data management tasks in high school informatics studies in Hungary, neither in spreadsheet nor database topics. Students need to be faced with large datasets as soon as possible to process data that is impossible to do manually one by one at the university level. To do this, they need to learn effective methods and approaches.

2 Mathability in Business Education

The first time we start the work with students we make it clear that there are two types of information systems, the first is transaction-oriented (OLTP) systems, and the other is analytics-oriented (OLAP) systems. The difference is the aim and the structure of the system. When we want many workers to work in a system simultaneously and input data about the operation of an enterprise, we use transactional (OLTP) systems. When we want to analyse the formerly recorded data we use analytical (OLAP) systems. The structural difference is the topic of a further course (Development and Management of System of Information systems), in this course we focus on the question of when and why we use the two types.

Usual problems for beginner Data Analysts are to create the visualization, and report based on a dataset (aka. unfolded OLAP cube). Microsoft Excel also contains OLAP elements, a fact that has greatly democratized data science, allowing anyone to learn the basics of OLAP through one of the most well-known application software. To achieve the goal they must learn basic data transformation and data enrichment methods, too.

High-level knowledge of spreadsheets is necessary in the economic field and this is more and more the OLAP knowledge [3] when analyzing job advertisements for data scientists. We found and adapted an appropriate technique (Sprego) to achieve our goals in Maria Chernoch and colleagues' Sprego work [1, 11, 12, 13, 14]. The essence of the Sprego method is that even though Excel has more than 600 built-in functions, we use up to a dozen of them regularly [8]. Starting from this basic idea, the Sprego method has defined the basic function group that we can use to solve many tasks in spreadsheets, using them as building blocks.

In the sense of CogInfoCom or mathability the other parts of our topics were investigated in some studies. We mentioned Csernoch and colleagues' spreadsheets research.

In 2012 Baranyi and Csapó provided the finalized definition of CogInfoCom and provided an overview of a number of related fields [15]. In the 2013 CogInfoCom Conference, where the concept of Mathability has been defined Gilányi and Baranyi declared that important goals of mathability are to investigate how artificial mathematical capabilities can be quantified, and to develop a set of methodologies using which human mathematical capabilities can be emulated and enhanced [16]. Gilányi and colleagues presented the practical usage of the mathability concept approach via tertiary education content many times [21, 22, 23, 24, 25]. Furthermore, there were several experiments in secondary education in problem-solving domain emphasizing the strong connection between mathematical and computational thinking [28, 29].

In Data Science – as we call it nowadays business analyses supported by computer programming and artificial intelligence – is a very popular domain in business and industry. Tertiary education can hardly follow the requirements of the market, and there are no suitable teaching methodologies to lay down the basics of the right conceptions of the topic. It is alarming that there are neither teaching nor evaluating methods for selecting the persons who could become really good data scientists.

Further important findings were found regarding constructive learning methods in Chmielewska and Gilányi's papers [17, 18, 19], whose experiments focused on computer-assisted self-study, furthermore, Chmielewska and Matuszak developed a coaching method as well [20].

Obviously, in many areas, it is essential today to lay the foundations for data processing and OLAP technology for example in medical, engineering sciences [30, 31].

3 Teaching Methodology in Introductory Business Informatics Courses

We divided the lessons into five parts:

1) Introduction to Excel usage,
2) Introduction to formulas,
3) Introduction to analysis,
4) Analytics,
5) Analysis with Pivot Tables and dynamic charts.

3.1 Introduction to Excel Usage[1]

Our first problems are the technical studies of excel usage. At first sight, it does not seem to be a mathematical capability. However, Csernoch [1, 11] investigated this problem as well because it can refer to the conscious usage of the spreadsheet tool. She declared the trial-error or bricolage level when a user was obviously just clicking back and forth without a conscious reason.

Chmielewska and colleagues pointed out [17, 18, 19], that uncontrolled computer-assisted self-education can be risky, while a lack of accuracy and sketchy solutions are characteristic for the young generation. Furthermore, they showed how important it is to reflect on the results obtained while learning by trial and error because the lack of feedback can lead to misunderstandings and misconceptions.

[1] The exercise location: http://takacs-viktor.hu/pages/excel/Xcelpract2016_TVL.xlsx Worksheet: 'Basics'

We agreed with this, and we begin the course by showing students what their former knowledge (including technical management of Excel) is about when we started to develop the computational tool consciously.

We believe the usage could be taught as seriously as a ‘real’ mathematical problem because it depends on analytical thinking. Instead of clicking back and forth, and instead of guessing what the program might know, the user has to click consciously and discover the functionalities of the application software. For that, the user has to recognize the same scheme as when they start to discover a computer tool.

In this domain, common problems are manipulations with various excel objects. There are no strict differences between the plans in these example areas. So this ’tool’ is good for introducing the method.

We have a prepared excel practice worksheet in which we teach the basic introductory manipulation examples with.

The tasks of the first exercise:

1) Unhide the hidden columns and rows,
2) Delete the columns with headers ’ab1’, ’acc’, ’bb1’, ’bcc’,
3) Search the cell which has content ’xxxxxxxxxx’,
4) Format the columns A, D, E to the same width,
5) Copy this worksheet and rename it to ’Data’ and delete the columns A-G,
6) Insert a new (empty) first row,
7) Insert a new column B and hide it.

Even in the introduction to excel and formalization, we follow Pólya’s steps.

In our former research study, we detected some analogies between Pólya’s steps and the steps of algorithmization. We used it in novice programming environments to teach the basics of computer programming, but as Csernoch and colleagues pointed out, the language of spreadsheets are functional programming languages [1], so we can transfer our methods into spreadsheets as well in this introductory course that we can see in Figure 1. [7]

Word problems in mathematics	Computer programming
1. Comprehension, understanding the problem	
2. Devising a plan	
Determinantion of unknown variable	Declaration of variables
Mathematical model (plan)	Algorithms and their implementation (plan)
3. Carrying the plan	
Solving equations	Compiling and running
Answering to the questions of the starting problem	
4. Discussion (Looking back)	
changing initial conditions	testing
generalizing the problem	debugging
looking for other methods for solution	optimizing code

Figure 1

Analogies between Pólya's steps, algorithmization, and common spreadsheet technical manipulations

3.1.1 Pólya's First Three Steps

The first step is understanding the problem, followed by detecting necessary data and declaring variables.

In the problems (tasks) described above, we can recognize four types of variables: function, object (Excel), condition, and parameter. It can be useful if we form all the tasks in a table, like in Table 2:

Table 1

The variable recognition in the first exercise

function	object (excel)	condition	parameter
unhide	column	visible = false (hidden)	width = …
hide	row	content = 'xxxxxxxxxx'	number format = …
delete	cell	id = A	height = …
copy	worksheet	id = 1	
insert	workbook	name = 'Data'	
select	selection	header = 'ab1'	
search	null		
format			

We have a special object named as selection, which is a conditional object, a subset of an excel object, and three special functions:

- select usually based on a condition, the result is a selection,
- search is a specialized 'select' based on a condition also, the result is a special selection, in this case a range of cells,

- the format usually has parameters (set an attribute of an object to a value) and the result is empty.

In this domain condition is a selection criterion, it contains an attribute of an object, a value, and a relation (usually the equal relation).

Formerly, in solving mathematical word problems we created a mathematical model and made the algorithm and their implementation in the concrete programming environment when we taught it as a novice computer programming task. Here, when we create the generalized steps (carriable logical plan) to solve the domain-specific 'word problems', similar to database design methodology [2], the following three steps are essential in the design of data warehouses and executive information systems, which are enhancements of Pólya's 'Devising a plan' step:

- Conceptual planning,
- Logical planning,
- Physical planning.

Conceptual planning means that 'I know what to do with which object and by what condition or parameter':

1) select an excel object,
2) apply a function on the selection,
3) set the possible parameters.

Logical planning means the formalizing of the concrete tasks one by one and translating the logical plans to physical plans. Carrying out the plan in these cases gives immediate results as soon as we do the manipulation on the object.

In the first task, we can see that we must divide the problem into two plans, because we realize that we cannot select hidden columns and rows, so we must select column and row ranges where the selection borders are visible columns and rows.

1) unhide(select(column, visible=false))
2) unhide(select(row, visible=false))

When we translate the logical plan to a physical plan we simplify the problems.

1) unhide(select(column, id ϵ {B,..,F})),
2) unhide(select(row, id ϵ {5,..,20})).

Logical plans of the second task:

1) delete(select(column, search(cell, content='ab1'))),
2) delete(select(column, search(cell, content='acc'))),
3) delete(select(column, search(cell, content='bb1'))),
4) delete(select(column, search(cell, content='bcc'))).

In this case we have A, B, C, D conditions which define four selections that could be described with one complex condition E = A or B or C or D = A + B + C + D, the result will be the logical plan:

delete(select(column, search(cell, content='ab1' or content='acc' or content='bb1' or content='bcc'))).

The physical plans for each logical plan are as follows:

1) delete(select(column, id=G)),
2) delete(select(column, id=H)),
3) delete(select(column, id=I)),
4) delete(select(column, id=j)).

or delete(select(column, id ϵ {G,..,J})).

From tasks 3 to 7 the plans are:

1) search(worksheet, content = 'xxxxxxxxxx'),
2) format(select(column, id ϵ {A,D,E}, width=a), where 'a' is a parameter value,
3) this is a complex problem again:
    a. copy(worksheet, this),
    b. rename(worksheet, new, name='Data'),
    c. delete(select(columns, id ϵ {A,..,G})),
4) insert(row, 1, 1).
5) another complex problem:
    a. insert(select(column, id=B), 1),
    b. hide(select(column, id=B)).

3.1.2 Looking Back

When we discuss all the above, we can create the generalized formal structure: (f,o[,c][,p])->o, basically f(o[,c][,p])->o

object=function(object[,condition][,parameter]).

We can develop an excel object hierarchy also: cell -> column and row -> worksheet -> workbook.

The Excel object hierarchy is important when we specify an Excel object, every definition contains every mother level of an excel object, when we leave any hierarchy level 'unspecified', that means 'actual' or 'this' worksheet or workbook of the cell, row or column. This object hierarchy is very important when cell references occur in later physical plans.

We extend the introductory manipulation in several ways, and always refer to the discussed introductory formal structures e.g. create function, table, pivot table, chart, slicer objects, with the relevant task, problems also, e.g. create a pivot table from the dataset, create a column chart based on a pivot table, set pivot table connection to a slicer, etc.

3.2 Introduction to Formulas[2]

From this point, we use the fully extended Pólya's steps of planning.

The identified variables of formulas as a problem-solving domain are result and formula in the following structure: result = formula, where the equal symbol must be written directly into the result selection, and formula must be replaced with the logically planned solution.

Basic introductory formula exercises:

1) Calculate Total Cost and format as Forint (Ft)
   a. hourly cost: 1,000 Ft/h
   b. Nr of hours: {23 h, 34 h, 45 h, 56 h, 78 h}
2) Fill the multiplication table!
   a. a: {1, 2, 3, …, 10}
   b. b: {1, 2, 3, …, 10}

3.2.1 Understanding the Problem

As you can see in the above examples we have constants, arrays, and matrices.

3.2.2 Devising a Plan

Conceptual plans: result = formula.

Logical plans:

1) There are two possible logical plans:
   a. A natural logical plan: $\overline{TC} = C * \overline{H}$, where TC is a Total Cost result array, C is the Cost constant, and H is the number of hours array.
   b. An alternative logical plan: $TC_j = C * H_j, 1 \le j \le 5$, where j is the jth element of a related array.
2) There are two possible logical plans:

2 The exercise location: http://takacs-viktor.hu/pages/excel/Xcelpract2016_TVL.xlsx Worksheet: 'Formulas'

a. A natural logical plan: $\overline{M} = \overline{a} \times \overline{b}$, where M is the result multiplication matrix, a and b are arrays with n elements.

b. An alternative logical plan: $M_{ij} = a_i * b_j, 1 \le i, j \le 10$, where i and j are the ith and jth elements of a related array and/or related matrix.

Physical plans:

In the process of transforming the logical plans to physical plans we define the excel reference objects (cell or cells) in both the result and formula parts. Generally, in our examples, the alternative logical plans are easier to implement, but the natural logical plans are more sophisticated solutions, and the plan will be closer to the natural thinking method.

In our examples the i translated to row identifier (numbers), j translated to column identifier (letters).

1) In this example we translate the logical plans to physical plans in two ways:

   a. Cell range based: $\overline{TC} = \overline{C} \times \overline{H}$ →B7:F7{=B5*B6:F6} the result planned into a cell range, the formula must be accepted with [CTRL]+[SHIFT]+[ENTER] key combination.

   b. Cell-based: $TC_j = C * H_j$ →B8=B5*B$6, (j=1)

      $1 \le j \le 5$ →copy(cell,B8,B8-F8)

      the result panned into a cell, we should fix the row part of the cell references, and the column part must remain in relative format, and we must copy the cell B8 (formula) from the result cell B8 until the cell F8.

2) In this example we translate the logical plans in two ways:

   a. Cell range based: $\overline{M} = \overline{a} \times \overline{b}$ →C14:L23{=B14:B23*C13:L13} the result planned into a cell range, the formula must be accepted with CTRL+SHIFT+ENTER key combination.

   b. Cell-based: $M_{ij} = a_i * b_j$ →C14=$B14*C$13, (i,j=1)

      $1 \le i, j \le 10$ →copy(copy(cell,C14,C14:L14),C14:L14-C23:L23)

      the result planned into a cell, we must use partly relative formula according to the **a** and **b** arrays in the logical plan, and we must copy the cell (formula) both between the columns and the rows.

3.2.3 Carrying Out the Plan

Generally, when we carry out the physical plan, we do the following steps:

1) select the result object (cell or cell range), green part of the plan,
2) write the formula, the black part of the plan, starting with =,
3) accept the formula with ENTER or CTRL+SHIFT+ENTER,
4) copy the result if you must.

We have to emphasize many times that as we have more complex problems, the number of logical plans could be increased. There is no single good solution. There are many possible ways of thinking and many possible logical plans. We must choose from them which one to solve. In the sense of mathability, when we examine humans and the computer as a whole system, when we want to develop computational thinking we have to highlight points when human thinking or decision is essential and cannot be substituted with artificial, emulated skills (or at least not at this level).

3.3 Introduction to Analysis[3]

In this domain, we are working with unfolded OLAP cubes (aka datasets), where OLAP means the classical On-Line Analytical Processes, and our database are organized to serve analytical needs. A dataset contains only two types of data: Measure and Attribute. Measures are always numbers with a unit, Attributes are everything else.

3.3.1 Understanding the Problem

The most basic task is to show a projected attribute or aggregated measure on a selected subset of an unfolded OLAP cube. This is the step when we recognize relevant data and gather them as variables.

Table 2

Analytical variable recognition

Recognized variable	Type of the variable		Relational algebra analogy
aggregate function	result		aggregation
Attribute xor Measure	result		projection
Condition	condition		selection

We must gather the variable values from the problems, choose a formula and substitute the values in the formula. Depending on the students' knowledge and interest we can introduce the basic concepts of relational algebra to lay down the foundations for future database management as we can see it in Table 2.

3.3.2 Devising a Plan

Conceptual plans in this domain are grouped in two basic structures:

[3] The exercise location: http://takacs-viktor.hu/pages/excel/Xcelpract2016_TVL.xlsx Worksheet: 'AnalysisIntro'

1) Conditional aggregation (1), means an aggregation of a measure where C condition is TRUE.

$$result\{= af\left(IF\left(C,M\right)\right)\} \tag{1}$$

2) Conditional select (2), the result is an attribute or measure value, where the condition is first time TRUE

$$result = INDEX\left({}^{A}\!\!/_{M}, MATCH\left(C\right)\right) \tag{2}$$

For these two basic structures, we must define the elements:

- ***result*** is the projection of an attribute or an aggregated measure,
- ***aggregate function*** is an element of the set of basic functions {SUM(), MIN(), MAX(), AVERAGE(), and COUNT()}. This set can be extended with special LARGE() and SMALL() and statistical aggregations too. note: COUNT() usually planned and implemented with a SUM() function of a measure 1,
- ***condition*** is a three-tuple consisting set (3), an attribute or measure is in a relation with a value.

$$c \in C, {}^{A}\!\!/_{M} \begin{matrix} < & \geq \\ > & \leq \\ = & \neq \end{matrix} V \tag{3}$$

On a logical level, we refer to the cubes' measures and attributes as a systematic arrangement of values, noted by the name of the attribute or the measure.

When we translate logical plans to physical plans, we must be aware of some considerations related to datasets. Every column of a dataset is a measure or an attribute. The first row must contain the name of the measures and the attributes, and values from the second row to the last. It must not be an empty column and/or row in the dataset. Dataset is a set of column-based cell ranges.

In the process of transforming the logical plans to physical plans, we define the excel reference objects (cell or cells) in both the result and formula parts. Usually, it must not but should be a good practice to fix the data source references of measures and attributes.

Additionally, we can define named tables with attribute and measure columns over a cell range as a column-based dataset. With these additions, our physical plans get closer to the logical plan.

To achieve this, we must follow the plan:

1) select(cell, A3:I102),
2) format(selection, table, header=true),
3) rename(table, name='dataset'),

4) select(cell, L18:N23),
5) format(selection, table, header=true),
6) rename(table, name='BMI').

When we plan complex problems, we follow the top-down planning method to break down a complex problem into simple sub-problems.

3.3.3 Carrying Out the Plan

Generally, when we carry out the physical plan, we build up our solution formulas strictly following the bottom-up concept, starting with the formulas related to the independent simple sub-problems.

3.3.4 Basic Introductory Analysis Examples

What is the pulse of the 51-year-old patient? - we have no defined aggregation in the task. In this case we, can use either the conditional aggregation or the selection conceptual formulas.

1) Conditional aggregation, we can use any aggregation from the four basic ones, because we get back a one-element-array of Pulse as a result of the condition.

Table 3
Variable recognition of conditional aggregation

Understanding the problem	af	any from {MIN, MAX, SUM, AVERAGE}
	A/M	Pulse
	C	Age = 51
Conceptual plan		$result\{= af\left(IF\left(C, M\right)\right)\}$
Logical plan		$result\left\{= SUM\left(IF\left(\overline{Age} = 51, \overline{Pulse}\right)\right)\right\}$
Physical plan		K5={SUM(IF(B4:B105=51,I4:I102))}
Physical plan based on tables		K5{=SUM(IF(dataset[Age]=51,datase[Pulse]))}

2) Conditional select, where 51 is the value from the condition, Age is the attribute part, and the last 0 is the equal relation.

Table 4
Variable recognition of conditional search

Understanding the problem	af	-
	A/M	Pulse
	C	Age = 51
Conceptual plan		$result = INDEX\left(A/M, MATCH\left(C\right)\right)$
Logical plan		$result = INDEX\left(\overline{Pulse}, MATCH\left(51, \overline{Age}, 0\right)\right)$
Physical plan		K5=INDEX(I4:I102,MATCH(51,B4:B102,0))
Physical plan based on tables		K5=INDEX(dataset[Pulse],MATCH(51,dataset[Age],0))

3.4 Analytics[4]

We are working with OLAP technology, where the data is stored in OLAP cubes with one or more fact tables and dimensions. Fact tables contain aggregated or raw measures and special attributes as dimension keys, dimensions have unique identifiers as dimension keys and dimensional attributes and possible dimension hierarchies. The fact tables and dimensions are connected to each other through the dimension keys.

The usual conceptual problems in this domain are:

- Data enrichment and calculated measures,
- zero to two-dimensional details of an aggregated measure,
- unfold dimensional attribute and/or measure into the cube.

Analysis examples

1) Data enrichment:
   a. Extend the date with temporal dimension attributes.
   b. Calculate the Quarters or Percentiles of the observation.
2) Simple answers:
   a. Calculate the sum of the ordered units.
   b. Which is the largest number of units in the orders?
   c. How many representatives got gifts?
   d. How many gifts did Quebec region get?
3) One dimensional aggregation:
   a. How many orders are there by regions?
4) Two dimensional aggregation:
   a. How many pieces did the representatives buy per Item?
   b. Compare the Cost (HUF) detailed by regions, by two freely selectable values of Item.
5) Unfold dimensional attribute:
   a. Calculate the Unit Price ($) for every order based on the item’s monthly price list.
   b. Calculate the Cost (HUF) based on daily exchange rates.
   c. Specify currency for every region in the orders.
6) Calculated Measure:

[4] The exercise location: http://takacs-viktor.hu/pages/excel/Xcelpract2016_TVL.xlsx Worksheets: ‘Orders’, ‘Prices’ and ‘Rates’

a. Calculate Cost ($) in every order.

b. Calculate the Delivery time.

3.4.1 Devising a Plan

Conceptual plans in this domain are grouped into two basic structures.

1) Conditional aggregation with zero to n details (4), this means an aggregation of a measure where C condition is TRUE.

$$M([A_1]..[,A_n])\{=af(IF(C,M))\} \tag{4}$$

2) Unfold dimensional attribute or measure into the fact table (5).

$$factA/_{factM}=INDEX\left(\dim.A/_{\dim.M}, MATCH(fact.DK_i,\dim.DK,0)\right) \tag{5}$$

$$1\le i\le n, n=|fact.DK|$$

The result is an unfolded attribute or measure value, where the fact table dimension key is found in the dimension unique key, based on the many-select problem.

Condition is a three-tuple consisting set, an attribute or measure is in a relation with a value.

We use boolean simple logic structures (6), translated to physical plan.

$$a,b,c,d\in C, c=a\wedge b=(a)\times(b), d=a\vee b=(a)+(b) \tag{6}$$

3.4.2 Examples

1) Simple, introductionary problems

Table 5
Variable recognition of simple problems

Problem		Calculate the sum of the ordered units!	Which is the largest number of units in the orders?
Understanding the problem	af	SUM	MAX
	A/M	Unit	Unit
	C	-	-
Conceptual plan		$result\{=af(IF(C,M))\}$	
Logical plan		result=SUM(Unit)	result=MAX(Unit)
Physical plan		X1=SUM($N$2:$N$53)	X6=MAX($N$2:$N$53)

2) How many representatives got gifts?

Table 6

Variable recognition of a conditional simple problem

Understanding the problem	af	SUM	
	A/M	1	
	C	Gifts<>””	Gifts>0
Conceptual plan		$result\{= af(IF(C,M))\}$	

Logical plans		result{=SUM(IF(Gifts<>””,1))}
		result{=SUM(IF(Gifts>0,1))}	

Physical plans		X11{=SUM(IF($Q$2:$Q$53<>””,1))}
		X11{=SUM(IF($Q$2:$Q$53>1,1))}	

This question is a bit confusing. How many is a simple sum of 1, but what does getting a gift mean? The natural translation should be gift = ‘anything’, where we have a value ‘anything’, but this is confusing as a value, we can’t write ‘anything’ in a cell. One possible solution could be to double negate the logical expression, which means not equal as opposite of equal, and ‘nothing’ as the opposite of ‘anything’.

3) How many gifts did Quebec region get?

Table 7

Variable recognition of a conditional simple problem

Understanding the problem	af	SUM
	A/M	Gifts
	C	Region=”Quebec”
Conceptual plan		$result\{= af(IF(C,M))\}$
Logical plans		result{=SUM(IF(Region=”Quebec”,Gifts))}
Physical plans		X18{=SUM(IF($J$2:$J$53=”Quebec”,$Q$2:$Q$53))}

4) How many orders are there by regions?

Table 8

Variable recognition of a conditional simple problem

Understanding the problem	af	SUM
	A/M	1
	C	Region=actRegion
Conceptual plan		$result\{= af(IF(C,M))\}$
Logical plans		result{=SUM(IF(Region=actRegion,1))} ,where actRegion = Region$_i$ ϵ Region, $1 \leq i \leq n$, n=|Region|

Physical plans		X18{=SUM(IF($J$2:$J$53=$V18,1))}
		copy(cell,X18,X18-X20)

5) How many pieces did the representatives buy per Item?

Table 9
Variable recognition of a conditional simple problem

Understanding the problem	af	SUM
	A/M	Units

	C	Representative=actRepresentative
		AND
		Item=actItem

Conceptual plan		$result\{=af(IF(C,M))\}$
Logical plans		result{=SUM(IF((Representative=actRepresentative)*(Item=actItem), Units))}, where actRepresentative = Representative_i ϵ Representative, $1 \leq i \leq n$, n=|Representative|, actItem = Item_j ϵ Item, $1 \leq j \leq m$, m=|Item|
Physical plans		AB24{=SUM(IF(($L$2:$L$53=$AA24)*($M$2:$M$53=AB$23), $N$2:$N$53))} copy(copy(cell,AB24,AB24-AF24),AB24:AF24-AB54:AF54)

3.5 Analysis with Pivot Tables and Dynamic Charts[5]

We are working with OLAP technology, where the data is stored in unfolded OLAP cubes with one or more fact columns and dimensional columns. Fact columns contain aggregated or raw measures, dimensional columns have only dimensional attributes and possible dimension hierarchies.

We collect and formalize management questions. The question and description are defined in a textual and formal way. The 'manager' is the person for whom the system provides information. As a result, she expects to see a data visualization report appearing on different dashboards.

We 'analyse' the management question based on the following considerations in Table 10:

What is the indicator? In which aggregation? What is the unit? Which visualization do we want to see? In Which detail(s) is there a slicer?

Table 10
Management question analysis

Indicator	$I^{\{u[,u]\}}_{\{af[,af]\}}$	the indicator ***I*** to be produced with *u* unit(s) in the upper right index and af aggregate function(s) in the bottom right index,
unit(s)		
aggregate function(s)		

[5] The exercise location: http://takacs-viktor.hu/pages/excel/Xcelpract2016_TVL.xlsx Worksheet: 'PivotAndCharts'

visualization	$\left(\begin{matrix} vt \\ \left[\begin{Bmatrix} s \\ s \end{Bmatrix}\right]\end{matrix}\right)^{v}$	the *v* visualization with the type *vt* (table, line diagram, bar graph, etc. ...) and optional *s* slicers (values can be ***D***$_{\{a\}}$ dimensional attribute, ***D***$_{\{v\}}$ subset of concrete values, or a ***D***$_{\{a\}}$ dimensional attribute in the *d* detail of another ***I*** indicator on the same dashboard)
slicer(s)		
detail(s)	$\left[\left(\begin{matrix} D_{\sum\{a\}} \\ \left[D_{\sum\{a\}}\right] \end{matrix}\right)^{\{d\}}\right]$	*d* details with D$_{\{a\}}$dimensional attribute(s), with optional $\sum\{a\}$ aggregation. *d* values e.g.: *row*, *col*umn, *cat*egory, *y* indicator

Formally (7):

$$\begin{matrix} I^{\{u[,u]\}}_{\{af[,af]\}} \\ \left[I^{\{u[,u]\}}_{\{af[,af]\}}\right]\end{matrix} \left(\begin{matrix} vt \\ \left[\begin{Bmatrix} s \\ s \end{Bmatrix}\right]\end{matrix}\right)^{v} \left[\left(\begin{matrix} D_{\sum\{a\}} \\ \left[D_{\sum\{a\}}\right]\end{matrix}\right)^{\{d\}}\right] \tag{7}$$

Physically we must create a Pivot Table visualization when we only need to create the definition, and a Chart if needed with slicers.

Table 11
Structure of the physical definition

Filter [Filter]	Column detail [Column detail]
Row detail [Row detail]	aggregate function(Measure) [aggregate function(Measure)]

Aggregate function and measure are defined before. Filters, Slicers and Details are based on Conditions.

Conclusions

Summarizing, formerly, with using Blockly Code and our analogy-based introductory computer programming teaching method we successfully taught and automatized a problem-solving strategy that pupils could store in their long-term memory and it can help them to analyse, formalize and generalize the problem successfully. In this paper, we presented that this concept could be adopted in tertiary education also in a more serious way and higher level. In tertiary business education, the aim was to make the problem of solving management questions independent from the computational tool we use. In traditional computer science teaching (at any level) we show the tool, and we show what it is for. We use examples to present the knowledge and the borders of the computational tool we teach. But now, we taught problem-solving, and we gave the students a way of thinking that could be the successful whatever computational tool we use for solving the problem. We gave them structures or schemas for formalizing management questions when they just create pivot analyses or even develop them into data warehouses. Furthermore, we showed that we can also teach technical manipulations as they would be 'problems' in a conscious way and we gave students transferable knowledge with it because the model that shows how to think about the problem is the same. And this is what we call computational thinking.

As we mentioned above there are neither teaching nor evaluating methods for selecting the people who could become really good data scientists. In our future research, we plan to create an evaluation system to detect data scientist talents. We believe that the presented method could be a very good base for this aim from the very beginning of the course.

Acknowledgement

Katalin Bubnó was supported by the EFOP-3.6.1-16-2016-00022 project.

References

[1] M. Csernoch: Sprego programming. Spreadsheets in Education, 8:3, 2015, pp. 1-35

[2] B. Halassy: Adatmodellezés. 2000 (In Hungarian)

[3] ICAEW: Spreadsheet Competency Framework. 2016

[4] R. B. Millwood: Review of literature on computational thinking. 2018

[5] S. Papert: Mindstorms. Children, computers and powerful ideas. Basic Books, 1980

[6] G. Pólya: How to solve it. 2nd ed., Doubleday, 1957

[7] K. Takácsné Bubnó and V. Takács: Solving word problems by computer programming. Proceedings of Problem Solving in Mathematics Education 2013 Conference (ProMath), A. Ambrus and É. Vásárhelyi, (eds.), Budapest, 2014, pp. 193-208

[8] J. Walkenbach: Excel 2010 Bible. Indianapolis: Wiley, 2010

[9] J. M. Wing: Computational thinking. Communications of the ACM, 49:3, 2006, pp. 33-35

[10] K. Bubnó and V. Takács: The mathability of word problems as initial computer programming exercises. Proceedings of the 8th IEEE International Conference on Cognitive Infocommunications (CoginfoCom), IEEE, 2017, pp. 39-44

[11] P. Biró and M. Csernoch: The mathability of spreadsheet tools. 2015 Proceedings of the 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2015, pp. 105-110

[12] P. Biró and M. Csernoch: Deep and surface structural metacognitive abilities of the first year students of Informatics. 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2013, pp. 521-526

[13] M. Csernoch, P. Biró, J. Máth and K. Abari: Testing algorithmic skills in traditional and non-traditional programming environments. Inf. Educ. 14, 2015, pp. 175-197

[14] G. Csapó: Sprego virtual collaboration space. 2017 IEEE 8th International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2017, pp. 137-142

[15] Baranyi P and Csapo A., Definition and synergies of cognitive infocommunication, Acta Polytechnica Hungarica, 9:1, 2012, pp. 67-83

[16] P. Baranyi and A. Gilányi: Mathability: Emulating and enhancing human mathematical capabilities. Proceedings of the 2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2013, pp. 555-558

[17] K. Chmielewska, A. Gilányi and A. Łukasiewicz: Mathability and mathematical cognition. Proceedings of the 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2016, pp. 245-250

[18] K. Chmielewska and A. Gilányi: Mathability and computer aided mathematical education. 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2015, pp. 473-477

[19] K. Chmielewska and A. Gilányi: Educational context of mathability. Acta Polytechnica Hungarica, 15:5, 2018, pp. 223-237

[20] K. Chmielewska, D. Matuszak: Mathability and Coaching. 8th IEEE Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2017, pp. 427-431

[21] A. Gilányi, N. Merentes and R. Quintero: Mathability and an animation related to a convex-like property. 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2016, pp. 227-232

[22] A. Gilányi, N. Merentes and R. Quintero: Presentation of an animation of the m-convex hull of sets. 7th IEEE Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2016, pp. 307-308

[23] G. Gy. Borus and A. Gilányi: On a computer program for solving systems of functional equations. 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2013, p. 939

[24] G. Gy. Borus and A. Gilányi: Solving systems of linear functional equations with computer. 4th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), IEEE, 2013, pp. 559-562

[25] G. Gy. Borus and A. Gilányi: Computer assisted solution of systems of two variable linear functional equations. Aequationes Math. 94, 2020, pp. 723-736

[26] K. Bubnó and V. Takács: Cognitive aspects of mathematics-aided computer science teaching. Acta Polytechnica Hungarica, 16:6, 2019, pp. 73-93

[27] A. Ambrus: Some cognitive psychological issues in mathematics education. Gradus 2, 2015, pp. 63-73 (In Hungarian)

[28] A. Kővári and M. Rajcsányi-Molnár: Mathability and creative problem solving in the MaTech Math Competition. Acta Polytechnica Hungarica, 17:2, 2020, pp. 147-161

[29] A. Kővári: Study of algorithm problem-solving and executive function. Acta Polytechnica Hungarica, to appear, 2020

[30] N. Yusupova, G. Shakhmametova and R. Zulkarneev: Complex analysis of medical data with data mining usage. Acta Polytechnica Hungarica, 17:8, 2020, pp. 75-93

[31] G. Kulikov, V. Antonov, A. Fahrullina, L. Rodionova, A. Abdulnagimov and M. Shilina: Formal method of structural-logical identification of functional model of subject area polycubic data matrix. Acta Polytechnica Hungarica, 17:8, 2020, pp. 41-59