IS (ICT) and CS in Civil Engineering Curricula: Case Study

R. Robert Gajewski
Warsaw University of Technology,
Faculty of Civil Engineering
Armii Ludowej 16, 00-637
Warszawa, Poland
Email: rg@il.pw.edu.pl

Lech Własak
Warsaw University of Technology,
Faculty of Civil Engineering
Armii Ludowej 16, 00-637
Warszawa, Poland
Email: lw@il.pw.edu.pl

Marcin Jaczewski
Warsaw University of Technology,
Faculty of Civil Engineering
Armii Ludowej 16, 00-637
Warszawa, Poland
Email: mjacz@il.pw.edu.pl

Abstract—The paper presents case study – Information Systems and Computer Science in Civil Engineering curricula. Introduction gives historical background of present role and position of IS, Information & Communication Technologies (ICT) and CS. Later details of the course of Information Technologies (IT) are presented in which elements of IS (ICT) and CS are combined. The use of spreadsheet and its Solver as well as Computer Algebra System (CAS) are stressed. Special attention is put on lectures which give necessary theoretical background for classes. Additionally there are presented some remarks about the subject Computing in Civil Engineering (CCE) which is natural successor of IT course. The paper is illustrated by the results of questionnaires performed in the beginning and in the end of semester. The main purpose of the article is to present changes and convergence between mentioned above subjects.

I. INTRODUCTION

THE place and role of Information Systems (IS), which can be treated in narrow sense as a term referring mainly to ICT, and Computer Sciences (CS) in curricula of Civil Engineering (CE) studies changed a lot in last few decades. In '70 and '80 mainframe computers were used and students were taught Algol and FORTRAN as programming languages. In '90 first PC laboratories were created so Algol and FORTRAN were replaced by Turbo Pascal. Students were also taught software like spreadsheets and text processors. Moreover for the purpose of Numerical Methods Computer Algebra System (CAS) MathCAD was introduced. In XXI century Windows environment and its applications dominates in curricula of studies. Additionally to these changes IS and CS are more separated than before. They are taught in different subjects namely Information Technologies and Informatics because Visual Basic for Applications (VBA) as sample programming language was in 2005 replaced by C++. The place and role of IS and CS in curricula of CE studies are subject of never ending discussions. The most orthodox engineers see no room for such subjects. In their opinion students' knowledge gained from secondary school on the level of European Computer Driving License (ECDL) described in [1] is absolutely satisfactory for future civil engineers. Their opponents see growing role of IS and CS in all engineering fields. Existing curricula is to some extend a kind of compromise between these two opposite opinions. In the rest of the paper laboratories and lectures in the field of IT, curricula of CCE and attempts to flip the education will be presented.

II. INFORMATION TECHNOLOGIES: LABORATORIES

Because in common opinion C++ taught on 2nd semester of BSc studies in subject named Informatics is not the best idea curricula of the subject Information Technologies (IT) taught on 1st semester is a trial to make a reasonable combination of IS and CS. Curricula of IT subject is based on compromise between the level of students' knowledge and foreseen needs of other subjects. While first point can be easily measured by questionnaires second one is hard to tackle with because of mentioned above conservative attitude of many teachers to the role of IS and CS in engineering.

A. Results of questionnaires
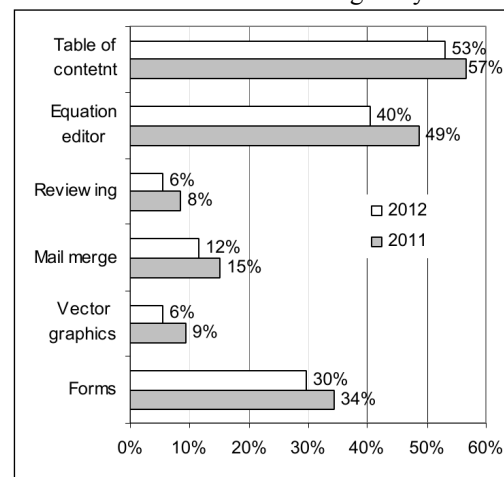
Questionnaires has been conducted regularly since 2011.

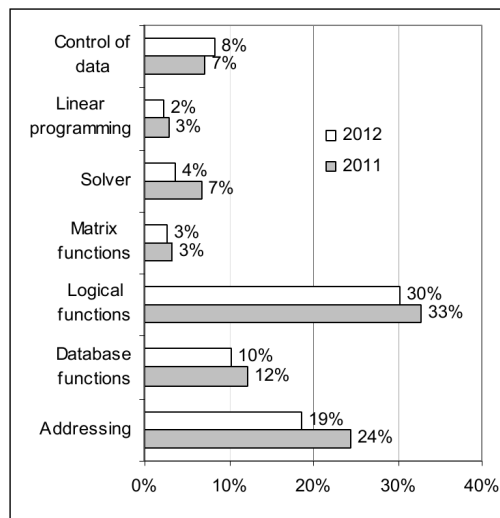Fig. 1 Sample questions concerning text editor

Fig. 2 Sample questions concerning spreadsheet

Their results are worse than expected. Students know how to run software like text editor or spreadsheet but they do not know how to use it in order to solve particular problem in effective way. Both Fig. 1 and Fig. 2 show that the knowledge of more advanced functions in text editor and in spreadsheet decreases. This means that material from IT on the level of ECDL stil should be present in curricula of studies.

B. Block 1: First things first…

First block consisting of three classes can be named first things first. Students are definitely very skilled mainly in dragging, dropping and tapping, so topics like file systems, file transfer and rights are quite new for some of them. Similar situation happens in the field of presentation graphics. Ten slides and 20 MB files are common problems because terms like resolution and appropriate graphic file format are in students' opinion not necessary for them. Last but not least text editors are also used in rather ineffective way. Styles, table of content, bibliography tools as well as mail merge are used very rarely. Last part of this short block consists of elements of Hyper Text Markup Language (HTML) and Cascade Style Sheet (CSS). People who are against this say that nowadays nobody creates web pages in pure HTML and CSS using only text editor. The purpose of these classes is different. First of all this is one of the simplest examples in which it is possible to presents to students how something from the field of IT works. Moreover in contradiction to clicking, dragging, dropping and tapping HTML and CSS require precise thinking which can be treated as an example of algorithmic thinking necessary in programming. Last on the list of pros is the fact that due to existence of numerous validators students can easily check results of their work.

C. Block 2: Spreadsheet - in between…

This block of classes is also three weeks long. There are three major points in this block: logical functions and conditional statements, database functions and their usage and

Solver described precisely in [2]. Solving engineering problems especially during design process means using conditions. Simple spreadsheet IF function combined with OR, AND and NOT functions are excellent introduction to programming. Table databases are not real relative databases but they can at least give to students a flavor of database systems and give to them an opportunity to learn how to formulate queries using specialized database functions. Solver opens the opportunity to solve more complicated problems like optimization. Students are taught two things: how to solve linear and nonlinear maximization and minimization problems and how to create appropriate mathematical model of a given problem. Fig. 3 shows Solver window for linear programming problem.
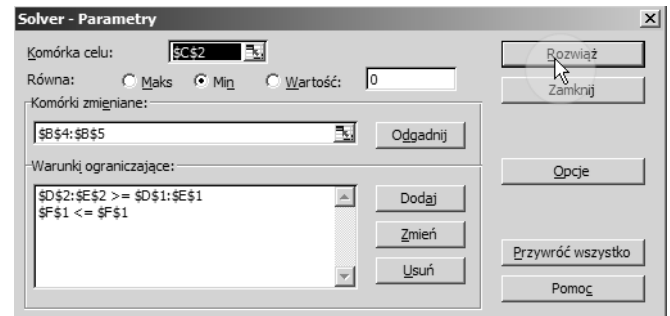

Fig. 3 Solver parameters for linear problem

D. Block 3: CAS – towards algorithmics and programming

The last and the biggest block of classes is devoted to CAS namely to MathCAD which is described in many books like: [3], [4] and [5]. Its presence in curricula of studies is a source of never ending discussions. In the opinion of many teachers students overuse MathCAD while preparing their design homework using it. It is enough that one person creates a file and all remaining can simply enter only data.

First part of MathCAD classes is devoted to solving classical mathematical problems:

- Symbolic calculations
- Definition of variables and functions
- Calculus: integrals, derivatives, limits.
- Matrix and vector operators and functions.
- Solving problems: linear and nonlinear equations, minimization and maximization.

Second part is devoted to programming. In the first part basic instructions (if, for, while) and control statements (return, continue, break) are introduced. The idea of this subject was inspired by the book [6]. List of algorithms is based on two Polish books from this field: [7] and [8]:

- Numerical algorithms (bisection, regula falsi, Newton method)
- Classical algorithms (Euclid's algorithm for greater common divider, Fibonacci numbers)
- Sorting algorithms (insertion, selection, bubble)

These algorithmic problems are solved together with classical programming problems including:

- Matrix and vector operations and
- Sums of series.

Fig. 4 presents sample MathCAD code of insertion sorting algorithm.

$$
\text{ins(w)} := \begin{vmatrix}
n \leftarrow \text{length(w)} \\
\text{for } j \in n-1, n-2..1 \\
\quad \begin{vmatrix}
\text{rob} \leftarrow w_j \\
i \leftarrow j+1 \\
\text{while } i \leq n \wedge \text{rob} > w_i \\
\quad \begin{vmatrix}
w_{i-1} \leftarrow w_i \\
i \leftarrow i+1
\end{vmatrix} \\
w_{i-1} \leftarrow \text{rob}
\end{vmatrix} \\
w
\end{vmatrix}
$$

Fig. 4 Sample MathCAD code

III. INFORMATION TECHNOLOGIES: LECTURES

Lectures are also very controversial part of the subject Information Technologies. There is quite common opinion that they are not necessary because in the second decade of XXI century everybody is a specialist in the field of IS and CS. Lectures are mainly based on four books: [9], [10], [11] and [12]. First one is accompanied by excellent web site enabling students to learn independently. Subjects of subsequent lectures are as follows:

- Introductory remarks: layers of computing system, the history of hardware and software.
- Binary values, number systems, conversions, floating point arithmetic.
- Data representation of text, graphics, audio and video; compression.
- Boole algebra, gates and circuits; von Neumann architecture;
- Elements and parameters of computing system.
- Algorithms and their representation: searching and sorting algorithms, recursive algorithms.
- Programming languages: translation, compilation, interpretation, basic programming structures, programming paradigms and languages.
- Types and data structures: stacks, queues, lists; subprograms.
- Operating system: role, memory and process management.
- File system and directories.
- Information systems and applications: spreadsheets and databases.

- Computer networks and their security: network addresses, cloud computing.
- Internet and the World Wide Web: HTTP, HTML.
- Introduction to artificial intelligence and expert systems.
- Limitations in the field of hardware and software; the untouched problems; questions and answers.

IV. COMPUTING IN CE

This subject plays supplementary role to IS and CS. Its curriculum consists of three blocks. First is devoted to Directs Stiffness Method (DSM) enabling students to solve trusses, beams and frames. Second part is devoted to stationary heat transfer problems – students learn how to solve set of Partial Differential Equations (PDE) using Finite Difference Method (FDE) or Finite Element Method (FEM). The last block devoted to optimal design is mainly based on the book [13]. Student learn from this block how to solve Operation Research (OR) problems as well as structural optimization problems. Tools used in curricula of this subject are mainly known from IT classes – this is spreadsheet and CAS software MathCAD. The stress is put on building appropriate mathematical model. In many cases for simple problems solution can be verified by hand calculations. Fig. 6 presents solver window for transportation problem.
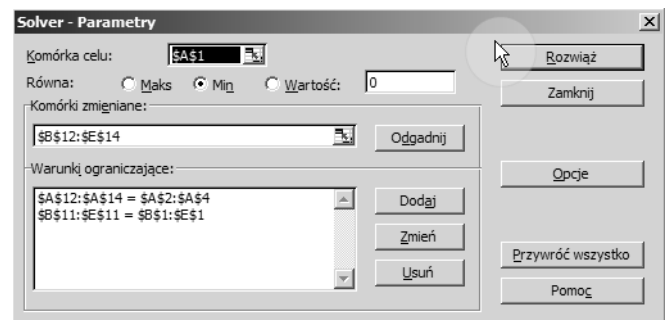

Fig. 5 Solver parameters for transportation problem

V. TOWARDS FLIPPED EDUCATION

For ten years students have been provided with different multimedia materials in the form of podcasts – personal on demand broadcasts. First podcasts prepared in the Division of Information Technologies (DoIT) had the form of screencasts – "digital recordings of computer screen output often containing audio narration". Screencasts contain software animations helping students to learn how to use software. Second kind of podcasts are slidecasts – "audio podcasts combined with slideshow". Slidecasts have the form of knowledge clips – short explanatory presentations of particular problem and its solution. Last kind of multimedia materials prepared by DoIT are webcasts – "media presentations distributed over the Internet using streaming media technology to many simultaneous viewers". In fact webcasts were lecture captures which were recorded and later distributed as podcasts. Tenths of hours of different podcasts stored on ed-
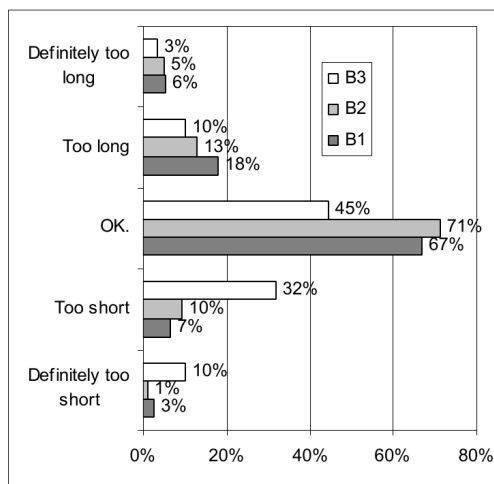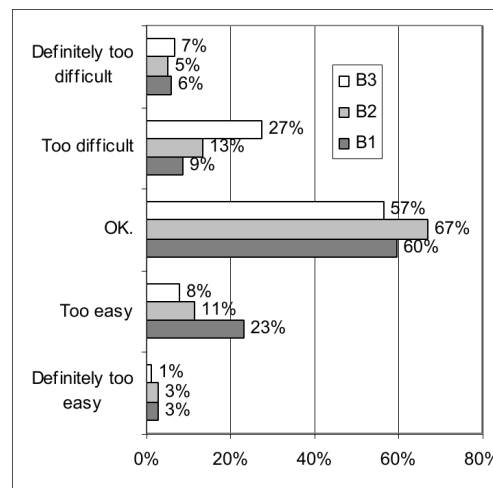
Fig. 6 Opinion about the length of blocks

Fig. 7 Opinion about the difficulty of blocks

ucational portal helped a lot during classes but did not have expected impact on quality of learning process measured in terms of grades obtained by students.

Starting from academic year 2012-2013 in some of the groups podcasts are used in different way. Students are asked to watch podcasts at home. During classes they should be prepared to use software without any problems and to solve using it particular problems. This idea is known as flipped classroom and is described precisely in [14] and [15]. First results of this experiment are to some extend promising - students gain better scores in flipped mode. Students are not very keen to spend time at home watching podcasts. They do prefer to "be taught" during classes. This problem can be easily solved by adding simple point to subject regulations – students should be prepared to computer laboratories and this fact is checked by means of test before the class. According to European Credit Transfer System (ECTS) in fact the same amount of time as at the university average student should spend learning at home. It is much more effective to watch passive in nature screencasts at home and solve problems with tutor in class than the other way round.

VI. CONCLUSIONS

The question of the place and role of IS (ICT) and CS in engineering curricula belongs to the category of ill posed problems. From the point of view of majority of teachers the most important for engineers is their intuition, so they neglect the knowledge from the fields of IS (ICT) and CS. Results from all computational programs should be verified and this is out of the question. One can say that the role of computations is to prove hand calculations. But on the other hand IS and CS can make engineers work more efficient.

From students' perspective IS tools are treated as specific black box. In order to use these tools in proper way at least basic knowledge form CS area is required.

Results of questionnaires from the end of the course presented on Fig. 6 and Fig. 7 show that the idea of sustainable presence of IS and CS in curriculum of civil engineering studies is accepted by students.

Presented paper shows that further convergence between IS and CS in higher education is necessary, because tools used by engineers are more and more complicated. New and innovative teaching methods like podcasts and evaluation of teaching can help to merge from IS to CS and vice versa. Curricula of studies should be continuously improved by adding new elements like for example Geographic Information Systems (GIS) or Building Information Modeling (BIM) software and systems.

REFERENCES

[1] J. Kennedy, *Complete Ecdl 5*, Gill & Macmillan Ltd, 2012.
[2] M. Harmon, *Step-By-Step Optimization With Excel Solver - The Excel Statistical Master*, Excel Master Series, 2012.
[3] B. Maxfield, *Essential Mathcad for Engineering, Science, and Math*, Academic Press, 2009.
[4] P. Pritchard, *Mathcad: A Tool for Engineering Problem Solving*, McGraw-Hill Science/Engineering/Math, 2011.
[5] R. W. Larsen, *Introduction to Mathcad 15*, Prentice Hall, 2010.
[6] D. Harel & Y. Feldman, *Algorithmics: The Spirit of Computing*, Addison-Wesley, 2004.
[7] M. M. Syslo, *Algorytmy*. Warszawa: Wydawnictwa Szkolone i Pedagogiczne, 1997.
[8] M. M. Syslo, *Piramidy, szyszki i inne konstrukcje algorytmiczne*. Warszawa: Wydawnictwa Szkolne i Pedagogiczne, 1998.
[9] N. Dale & J. Lewis, *Computer Science Illuminated*, Jones & Bartlett Learning, 2012.
[10] J. G. Brookshear, *Computer Science: An Overview*, Prentice Hall, 2011.
[11] R. T. Watson, *Information Systems*. CreateSpace Independent Publishing Platform, 2012.
[12] R. Stair & G. Reynolds, *Principles of Information Systems*, Cengage Learning, 2011.
[13] J. Arora, *Introduction to Optimum Design*, Academic Press, 2011.
[14] J. Bergmann & A. Sams, *Flip Your Classroom: Reach Every Student in Every Class Every Day*. International Society for Technology in Education, 2012.
[15] J. Gerstein, *The Flipped Classroom: The Full Picture*, 2012.