Vilnius University
INSTITUTE OF MATHEMATICS AND INFORMATICS
LITHUANIA

INFORMATICS ENGINEERING (07 T)

MODEL OF FUNDAMENTAL INFORMATICS CONCEPTS EDUCATION

Gabrielė Stupurienė

October 2017

Technical report MII-DS-07T-17-02

VU Institute of Mathematics and Informatics, Akademijos str. 4, Vilnius LT-08663, Lithuania

www.mii.lt

Abstract

Bringing informatics subject to schools means preparing young people to be creators of information technology. Introducing informatics concepts in primary, basic and secondary education (K-12) enrich student ability to create many things using information technologies. It is important to find out which informatics concepts are relevant for school curricula. Identification process of informatics concepts is based on process modelling language.

On the other hand, it is important to help students to solve problems by using technology and developing computational thinking in various areas. For this purpose, two-dimensional categorization system is implemented for educational informatics tasks.

Finally, all these components are integrated in model of fundamental informatics concepts education.

Keywords: Informatics concepts, identification of Informatics concepts, concept-based educational task, Informatics education, two-dimensional categorization system.

Content

1 Introduction

While informatics is a well-established discipline in higher education around the world, it is not the case in secondary education, with the exception of a few countries. Generally, what is taught is not informatics as a subject with its own methods, concepts, and principles, but some software tools with the goal that the use is sufficient for students to acquire skills. In addition, an analysis of the current situation reveals that the real competencies of teachers and students in informatics are far weaker than might be expected in secondary education (Hadjerrouit, 2009).

The fact that there is much less published research work on the pedagogy of school informatics has huge consequences for the teaching and learning of the subject. As a result, there still persist strong disagreements about the nature of school informatics, its aims, content, teaching and learning methods, and assessment approaches (Hammond, 2004).

The concept can be understood as extensive information on a particular object, existing in human mind. The content of a concept can vary a lot as it depends on personal experience. Concepts of informatics are tightly related with our intensions: what we would like to teach at school. A concept can be defined as a set of objects having common attributes (Dagiene, Stupuriene, 2016).

It is problematic to decide what we should include in informatics education for primary and secondary schools. Some reasons for that could be as follows:

1. Informatics is relatively young and rapidly evolving science;
2. There is a variety of different practical applications of informatics and that overruns core theoretical and scientific concepts;
3. There is no common agreement (framework) on what should be introduced in school from the theory of informatics, and whether it should be introduced at all.

The findings of Hadjerrouit (2009) show that conceptual understanding is a critical factor of success in teaching and learning informatics.

From other point, we live in 21st century, so student needs to have skills that are required for Information Age.

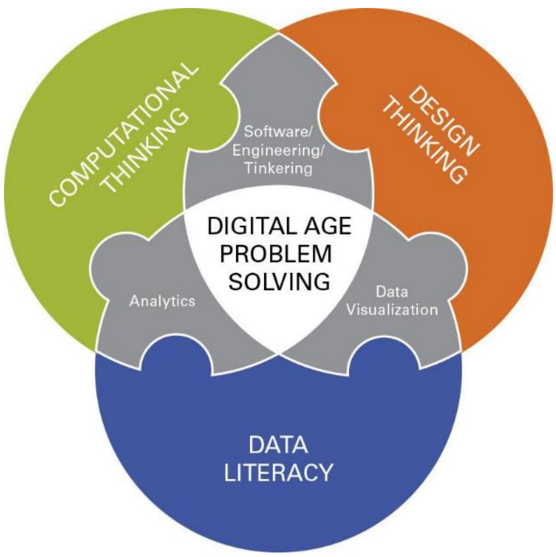

Fig. 1: Digital age problem solving elements (by Mark Samberg, 2017)

Digital-age problem solving is a term to describe the collection of the skills and strategies required for students to be able to identify, frame, and solve problems in the Information Age. In other word it means by using technology and data to help solve problems for people. The term Design Thinking is a strategy to solve complex problems and develop human-centered solutions and differs from traditional problem solving.

2 Identification of informatics concepts

Process-based development of competence models to computer science (informatics) education is provided by Zendler, Seitz and Klaudt (Zendler *et al.*, 2016). The process model (cpm.4.CSE) includes eight subprocesses: A1 - determine competence concept; A2 - determine competence areas; A3 - identify computer science concepts; A4 - assign competence dimensions to computer science; A5 - code competences; A6 - formulate competences; A7 - formulate learning tasks; and A8 - formulate test tasks (Fig. 2).

The processes, activities, and transformations of cpm.4.CSE are documented in Integration Definition for Function Modelling (IFED0), a process modelling language that is standardized, widely used and easy to understand.

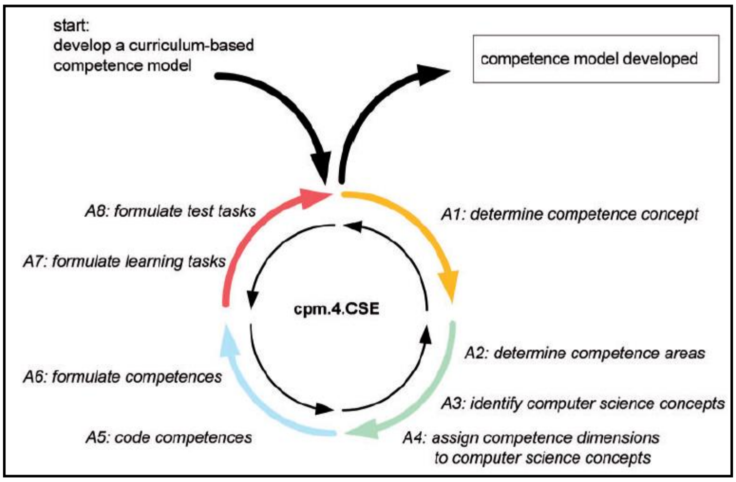

Fig. 2: Process-based competence model for computer science education (Zendler et al., 2016)

After long discussion with experts and teachers from informatics education there was decided to modify this subprocesses of Informatics concepts identification for some reasons:

- process model (*cpm.4.CSE*) is dedicated to higher education because input to subprocess A2 is based on literature and curricular elements from colleges and universities.
- we are interested in informatics concepts identification for primary and secondary education, also high education (K-12), so it is not enough to determine competence area and identify informatics concepts, but also need to provide and keywords. This is very useful for teachers to help them easily find and choose the particular concept-based task. It is important to mention, that

informatics is the only subject that teachers of primary schools have to teach, but never studied themselves.

Sequence of subprocesses related to informatics concepts identification (Fig. 3) should start with determination of competence area (Informatics concepts area) – A2, then identification of informatics concepts subarea – A3(1), and finish with identification of informatics concepts keywords – A3(2).

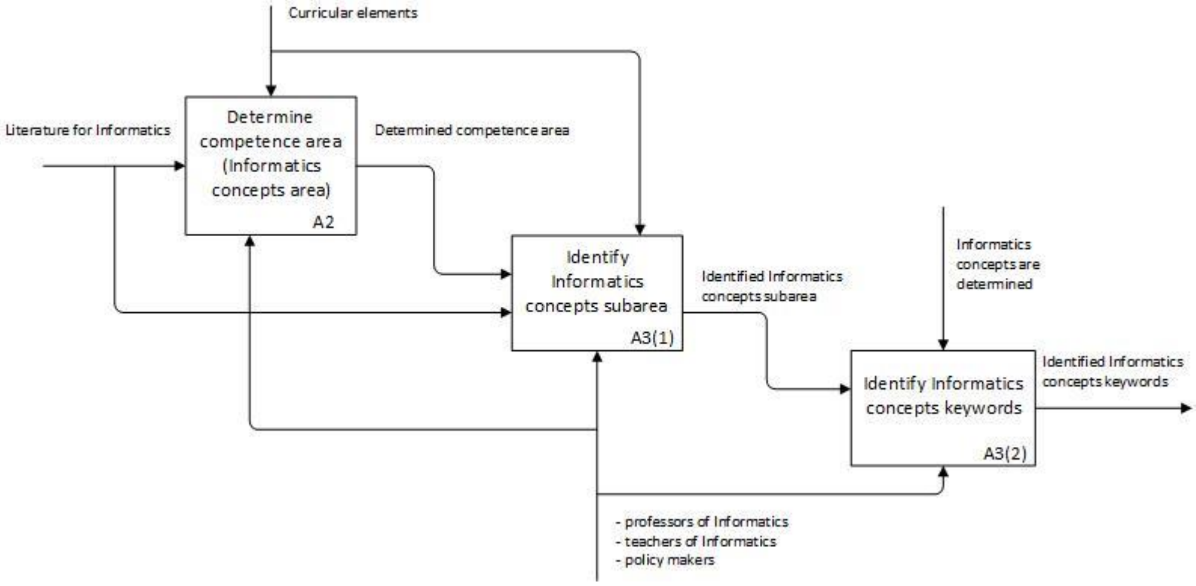

Fig. 3: Extended subprocesses for informatics concepts identification

Input to subprocess A2 (determine Informatics concepts area) are literature for Informatics education in school, e.g., Australian Curriculum: Digital Technologies, v8.3, 2016; The national curriculum in England, 2013; K-12 Computer Science Framework, 2016; CSTA K–12 Computer Science Standards, 2011, and published papers, e.g., Bell et al., 2014; Caspersen, Nowack, 2013; Sysło, Kwiatkowska, 2015; Barendsen, Steenvoorden, 2016; Barendsen et al., 2016.

Control conditions for subprocess A2 are curricular structural elements (for K-12 education), which may be different from country to country.

In *mechanisms* roles are involved teachers, professors of informatics in collaboration with education policy makes, who are responsible for selecting the competence areas.

Output of subprocess A2 are determined informatics concepts areas. We determined five areas (see Table 1): Algorithms and Programming; Data, Data Structures, and Representations; Computer Processes and Hardware; Communication and Networking; Interactions, Systems, and Society.

Input to subprocess A3(1) (identify informatics concepts subarea) are determined informatics concepts areas and the same literature as to subprocess A2. Also the same *control* and *mechanisms* elements.

Output of subprocess A3(1) are determined informatics concepts subareas. List of subareas are provided in Table 1.

Input to subprocess A3(2) (identify informatics concepts keywords) are determined informatics concepts subareas.

Output of subprocess A3(2) are identified informatics concepts keywords. The results are provided in Table 1 as well.

Table 1. Outputs from subprocesses A2, A3(1) and A3(2)

INFORMATICS CONCEPTS AREA	INFORMATICS CONCEPTS SUBAREA	INFORMATICS CONCEPTS KEYWORDS
Data, Data Structures, and Representations	Data	Classification; Databases; Data mining; Information; Metadata; RAID array;
	Data structuring	Array; Biconnected graph; Binary tree; Graph; Hash table; Index; Linked list; List; Queue; Record; Set; Stack; String; Tree;
	Information representation	Binary representations; Bitmap; Character encoding; Color representation; Coordinates; Data compression; Finite-state machine; Graph representation; Hexadecimal representations; Image/Sound representation; Integer; Real numbers representation; Pattern; Vector graphics;
Algorithms and Programming	Algorithms	Binary search; Bubble sort; Breadth-first search; Depth-first search; Dijkstra's algorithm; Kruskal's algorithm; Prim's algorithm; Quick sort; Selection sort;
	Computing problems	Eulerian path; Fractal; Knapsack problem; Maximum flow; Pattern recognition; Searching; Shortest path; Sorting; Scheduling; Traveling salesman problem;
	Design principles	Automaton; Bottom up; Brute-force search; Computational complexity; Divide and conquer; Dynamic programming; Greedy strategy; Heuristic; Invariant; Optimization; Priority; Permutation; Sequencing; State; Top down;
	Programming	Algorithm; Coding; Command; Constants; Constraints; Encapsulation; Flowcharts; Function; IF conditions; Inheritance; Iteration; Loop; Object; Parameter; Procedure; Program; Programming language; Recursion; Pseudocode; Variable;
	Logic	Boolean algebra; Logic circuits; Logic expression; Logic gates; Operations AND, OR, NOT;
Computer Processes and Hardware	Hardware and related software	Assembler; Cloud computing; Computer components; Computer devices; Fetch-execute cycle; Grid computing; Interpreter; Logic gates; Logic circuits; Machine code; Memory; Operating systems; Registers; Translator; Virtualization;
	Processes	Deadlock; Multithreading; Parallel processing; Process scheduling; Semaphore; Turing machine;
Communication and Networking	Cryptology	Authentication; Code; Cryptography; E-signature; Encryption; Parity bit; RSA algorithm; Security;
	Networking	Client/server; Computer network; Protocol; Secure data transmission; Topology; Peer-to-peer; Watchdog; Data transmission; Web services;
Interactions, Systems, and Society	Interaction	Graphical user interface; Interaction; Robotics; Online processing; Batch processing; Input/Output; Webbots; Digital assistant;
	Society	Authentication; Cloud computing; Computing history; Copyright; Digital footprint; E-bullying; E-commerce; Ethics; Hacking; Legal issues; License; Malware; Netiquette; Open Source; Password; Phishing; Self-identity; Social engineering; Social issues; Virus;
	Software design	Agile; Alpha and Beta testing; Black-box testing; Debugging; Localization; Program tracing; Templates; Testing; Waterfall; White-box testing;

3 Two-Dimensional Categorization System for Educational Tasks in Informatics

Conceptualization is formation of concepts (Papaurelytė-Klovienė, 2007). The term that is associated to conceptualization is categorization. Thus it obvious that when we deal with concepts, we cannot forget the importance of conceptualization and categorization. The process of conceptualization allows us to form concepts in our minds. Categorization allows us to categorize them according to some certain features.

Based on review of previous category systems for educational informatics tasks with relation to content, the content of school informatics can be divided into five areas:

1) Data, Data Structures, and Representations;
2) Algorithms and Programming;
3) Computer Processes and Hardware;
4) Communication and Networking;
5) Interactions, Systems, and Society.

For practical use, when developing educational informatics tasks, a precise description of each category is needed. One way of achieving this uses keywords. Keywords are important to assist in the categorization. They will also be important to teachers who wish to find tasks that fit with the topic being taught in the curriculum (Dagiene, Sentance, 2016).

The area of computational thinking covers a range of different skills relating to problem-solving. The issue becomes the need to select a categorization system which is true to the definition of computational thinking whilst encompassing the range of skills that students utilize when solving concept-based educational tasks.

A suggested categorization of computational thinking skills follows the work of Selby and Woollard (2013) and which has been adopted by Computing at School in the UK in developing guidance on computational thinking for teachers (Csizmadia et al., 2015). This describes aspects of computational thinking skills exhibited by learners as falling into the five categories below:

1) Abstraction;
2) Algorithmic thinking;
3) Decomposition;
4) Evaluation;
5) Generalization.

Incorporating both described categorization systems (computational thinking skills and informatics concepts), we can compose a two-dimensional system which can be represented as shown in Table 2. The suggested categorization system is dedicated to classify educational informatics tasks.

Table 2. Two-dimensional categorization system

	Data, Data Structures, and Representations	Algorithms and Programming	Computer Processes and Hardware	Communication and Networking	Interactions, Systems, and Society
Abstraction					
Algorithmic thinking					
Decomposition					
Evaluation					
Generalisation					

The categorization system could be used in addition to encourage the development of educational tasks that use a variety of informatics concepts areas as well as computational thinking skills. On the other hand, this system helps teachers of informatics to choose the content of lesson and helps effectively to select the tasks according to the particular topic.

The matrix presented in Table 2 demonstrates that this schema can be seen as two-dimensional. Although more complex, this new system captures more information about the task in a way that will be accessible and will support both task setter and teacher.

More: *Dagienė, Valentina; Sentance, Sue; Stupurienė, Gabrielė. Developing a two-dimensional categorization system for educational tasks in informatics // Informatica. ISSN 0868-4952. 2017, Vol. 28, no 1, p. 23-44.*

4 Model of fundamental informatics concepts education

There was implemented model of fundamental informatics concepts education based on identification process of informatics concepts and two-dimensional categorization system for educational tasks. UML data model of this model is shown in Fig. 3.

The core point of model is concept-based educational informatics task, which depend on educational stage (grade of student). Every task incorporates informatics concepts keywords as well as computational thinking skills categories.

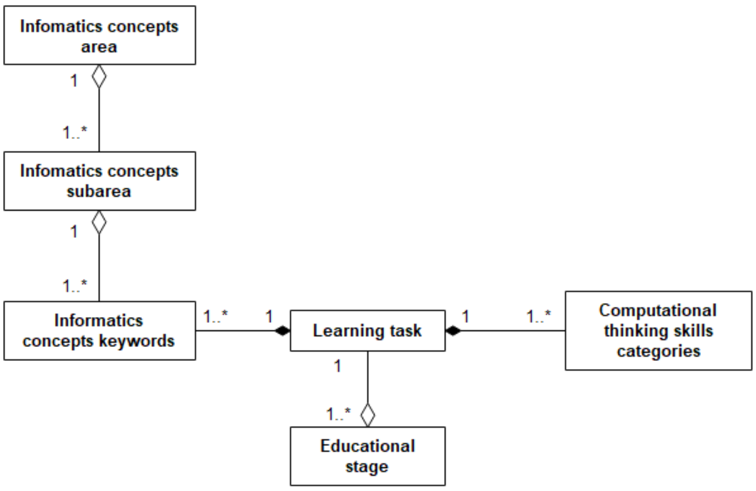

Fig. 3: UML data model of proposed model for informatics concepts education

Process model will be notated by using BPMN 2.0. standard.

5 Design science research

Design science is an outcome based information technology research methodology, which offers specific guidelines for evaluation and iteration within research projects. Design science research focuses on the development and performance of (designed) artifacts with the explicit intention of improving the functional performance of the artifact. Design science research is typically applied to categories of artifacts including algorithms, human/computer interfaces, design methodologies (including process models) and languages. Its application is most notable in the Engineering and Computer Science disciplines, though is not restricted to these and can be found in many disciplines and fields.

March and Smith's differentiation of constructs, models, methods, and instantiations as artefact types (1995) is commonly accepted in IS design science research (Vahidov, 2006). Constructs constitute the 'language' to specify problems and solutions. Models use this language to represent problems and solutions. Methods describe processes which provide guidance on how to solve problems. Instantiations are problem-specific aggregates of constructs, models, and methods. Information systems can be interpreted as aggregates comprising specific instantiations of constructs (e.g., modelling primitives implemented by meta models of modelling tools), models (e.g., process models implemented as workflows), and methods (e.g., project methods implemented during software package introduction).

Design-oriented research has a long tradition in Europe. While design science research is the dominating IS research paradigm in the German-speaking countries, a large number of design-oriented researchers can also be found in the Nordic countries, the Netherlands, Italy, and France, just to name the biggest communities (Winter, 2008).

This design science research method is being considered for involving in doctoral research.

6 Conclusion

Concept-based educational informatics tasks are very important both for students and task developers (teachers): students should be encouraged to think about informatics as science, educators should think about harmonization of syllabus of informatics.

Two-dimensional categorization system, which integrates computational thinking skills and informatics concepts, is description of new approach to informatics tasks. Computational thinking skills are among 21st century skills for Information age students.

Ideas of doctoral research are discussed with ETH Zurich professors Walter Gander and Juraj Hromkovic.

As one of case for fulfilment of proposed educational model is International Informatics and Computational Thinking challenge Bebras (www.bebras.org).

7 References

1. ACM Computing Classification System, 2012. Available from: http://www.acm.org/about/class/class/2012
2. Australian Curriculum: Digital Technologies, v8.3. (2016). Available from http://www.australiancurriculum.edu.au/technologies/digital-technologies/structure
3. Barendsen, E., & Steenvoorden, T. (2016, October). Analyzing Conceptual Content of International Informatics Curricula for Secondary Education. In

International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 14-27). Springer International Publishing.
4. Barendsen, E., Grgurina, N., & Tolboom, J. (2016, October). A New Informatics Curriculum for Secondary Education in The Netherlands. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 105-117). Springer International Publishing.
5. Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. ACM Transactions on Computing Education (TOCE), 14(2), 10.
6. Caspersen, M. E., & Nowack, P. (2013, January). Computational thinking and practice: A generic approach to computing in Danish high schools. In Proceedings of the Fifteenth Australasian Computing Education Conference-Volume 136 (pp. 137-143). Australian Computer Society, Inc
7. Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., Woollard, J. (2015). Computational thinking: a guide for teachers. Available via internet: http://computingatschool.org.uk/computationalthinking. Accessed 20 September 2016.
8. Dagiene, V., Stupuriene, G. (2016). Informatics Concepts and Computational Thinking in K-12 Education: A Lithuanian Perspective. Journal of Information Processing, 24(4), 732-739 (Invited paper).
9. Dagiene, V., Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. In: Lecture Notes in Computer Science, Vol. 9973, pp. 28-39.
10. Department for Education, (2013). The national curriculum in England: Framework document (London, DfE).
11. Hadjerrouit, S. (2009). Teaching and Learning School Informatics: A Concept-Based Pedagogical Approach. Informatics in Education, 8(2), 227-250.
12. Hammond, M. (2004). The peculiarities of teaching information and communication technology as a subject: A study of trainee and new ICT teachers in secondary schools. Technology, Pedagogy and Education, 13(1), 29-42.
13. IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0, Institute of Electrical and Electronics Engineers, Inc., 1998.
14. K-12 Computer Science Framework Steering Committee. (2016). K-12 Computer Science Framework. Retrieved from http://www.k12cs.org.
15. Papaurelytė-Klovienė S. *Lingvistinės kultūrologijos bruožai*. Šiauliai, 2007.
16. Seehorn, D., et al., (2011). CSTA K–12 Computer Science Standards. Computer Science Teachers Association and ACM.
17. Selby, C., Woollard, J. (2013). Computational thinking: the developing definition. Available via internet: http://eprints.soton.ac.uk/356481.
18. Sysło, M. M., & Kwiatkowska, A. B. (2015, September). Introducing a new computer science curriculum for all school levels in Poland. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives (pp. 141-154). Springer International Publishing.
19. Zendler, A., Seitz, C., & Klaudt, D. (2016). Process-Based Development of Competence Models to Computer Science Education. *Journal of Educational Computing Research*, *54*(4), 563-592.
20. Vahidov, R. (2006). Design researcher's IS artifact: a representational framework. Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006) pp 19–33 Claremont.

21. Winter, R. (2008). Design science research in Europe. European Journal of Information Systems, 17(5), 470-475.