Eötvös Loránd University
Doctoral School of Informatics
Information Systems Doctoral Program

EDUCATIONAL METHODOLOGICAL ISSUES OF INFORMATICS (PROGRAMMING)

Theses of Ph.D. Dissertation

Zsuzsanna Szalayné Tahy

Head of the School: Dr. Erzsébet Csuhaj Varjú
Head of the Program and Supervisor: Dr. László Zsakó
Department of Media and Education Informatics

DOI: 10.15476/ELTE.2021.188

Budapest, 2021

Introduction

Twenty-five years after graduation and having spent twenty years as IT teacher, I started doctoral school in 2013. It was the constant change of environment that encouraged my continuous training in IT and in pedagogy, as well as participating in quality management and in skill development trainings, leading to passing the teacher's professional qualification exam. I was shocked when scientific and current political potencies stated that programming is unlearnable for some or, the other extreme, that children born these days are absorbing IT knowledge with breast milk. One of the arguments of the national core curriculum published in 2013 is that all teachers, including the IT illiterate, can educate IT.

Against common belief the knowledge gained in trainings and teaching practice is ineffective and unreliable. Therefore, I decided **to research the nature of informatics, including programming knowledge, and as the aim of my doctoral research, define it and then, relying on the postulates, I define the purpose of informatics education, I recommend suitable methods for teaching and learning, recommend ways of developing skills, declare the aspects of knowledge qualification.**

Objectives

My aim is to provide an effective professional, pedagogical and learning management framework for the teaching informatics, in which I assign the appropriate tools to the goals to be achieved. To do these:

1. Based on literature and the opinion of the CS and the IT profession, I define the subject of education: what is informatics as a science and programming as an activity.
2. Based on my experience in high school education, I determine the characteristics of CS and IT that are important for carrier choice. Based on the significant paradigms of CS and IT in public education, I would like to give the expectations of public education in the field of informatics and programming.
3. I analyse the effectiveness of each pedagogical method in terms of informatics education, especially the teaching of programming, and evaluate it in terms of dropout.
4. Based on the content and methodological analysis of informatics in public education, I recommend tools and methods to achieve expectations indicated above (point 2 & 3), to reduce the gaps and deficiencies.

5. Combining the above, I provide a framework for the informatics teacher for professional solution of the problems in the field of informatics and programming education.

Research Methods

A dissertation of Computer Scientist often contains mathematical models. An IT Engineer's dissertation is characterized by the presence of a large amount of measured data verifying or validating its invention. In my work I tried to apply both research methodologies by modelling informatics education and proving the impact of the use of the tool or method by analyzing large amount of data. However, the main field of my research is the teaching of informatics and —as part of it— programming. The focus of education science is the person, the teaching of that person. My dissertation focuses to the methodological issues of education, how one can teach informatics. Therefore, in a significant part of my research I used the typical methods of educational science.

In all areas of my research, I used analytical methods to analyze and build a system of scientific theoretical knowledge and the regulation applied in practice. I participated in national and international professional methodological conferences, analyzed the related scientific literature, and monitored state and civil initiatives. I studied university courses on CS and IT faculties: I have participated in several lectures, practical courses, and labs.

During the research I tried to apply empirical qualitative methods that do not modify the subject of observation. I observed my environment – my students, colleagues, acquaintances – often without knowing the subject of the observation. A dropped word, a life situation, a reaction was included was listed in the "collection of cases". I "conducted the interviews" in the form of informal conversations and mailing or as part of teaching process. Instead of a questionnaire survey, I observed opinions about specific situation, ways of solving problems. My notes are case studies.

I tried learning as a guest student. In some cases, I observed myself, analyzed my reactions and performance. In other roles, as a demonstrator, a guest lecturer, or tutor I participated in education, proofreading, improving dissertations, monitoring students' progress, and evaluating results. I completed the observation of students and teachers with the observation of myself in the role of students and educator, the "collection of cases" contains several self-reflections.

I also made long-term observations in qualitative research. I tried to monitor the progress and career of alumni. I met students I had been teaching in high school for years again when they attended university. I have been monitoring the evolution of some students for years.

As part of my research, I have built a system of the experiences, consciously combined the methodological and pedagogical aspects of informatics in the analysis of problems related to the teaching of informatics and in the search for solutions.

Results, Theses

Definitions

An important aim of my research is to review the diversified professional material and to qualify its relevance. The result of my analytical research is the definition of basic elements of my research topic, as my theses relate to these definitions.

Learning Activity Unit – LAU

To talk about the learning process in detail, its obstacles and problems, I needed a model describing the process. I integrated six models known in educational methodology. The result is the Learning Activity Unit (LAU), a learning template and analysis tool. In the description and representation of the LAU I prefer the expressions used in informatics, the elements of algorithm concepts and the description of the state machine. Instead of the overlapping levels of the "pyramid" used in the teaching methodology, I emphasize states, transitions as well as temporality. [Fig. 1]:

1. **Initial learn**: The first step in the learning process is to connect with the knowledge. We define three characteristics for the quality of cognition:
   - A) **Active** cognition: Connection between new and previous knowledge are formed immediately. The new knowledge has a prepared place, it can be used immediately.
   - M) **Moderated** cognition: The learner knows where the knowledge is usable; one can recall what has been learned or link to other knowledge with the help of a note.
   - P) **Passive** cognition: The learner is not able to understand and memorize the knowledge in context. New knowledge is not placed in context with another knowledge.

2. **Try**: The first testing and applying of the knowledge in the given context; validity check.
3. **Experiment**: The learner explains the details of the knowledge, understand further relationships. One exercises the use and gain basic practice.
4. **Pause**: Dealing with another knowledge (filtering, forgetting).
5. **Use**:
    a. **Repeat** The use of knowledge is in the same form as it has been learned. It is the **repetition** of learned stuff.
    b. **Modify**: The use of knowledge is in typical situations, in **modified**, but close to the learned forms.
    c. **Create** The use of knowledge is purposeful and goal oriented. The unexpected situation claims creative steps in the solution.

← Back to phase 4 or —after a too long break, big change in the environment with a new approach— phase 1.

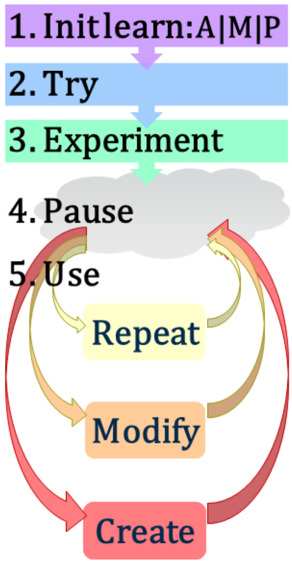

Fig. 1: The LAU's phases

LAU-model

The description of a learning-teaching process contains several learning units and their connection. The LAU-model describes the connection between the LAU-s and helps interpret a longer or more complex learning process. Units in the LAU-model can follow each other if a knowledge bases on the other knowledge or the latter one modifies the earlier one's notion. They may contain each other in the case of parts of complex knowledge and they may be parallel to each other if they are parts in a complex LAU or if they are connected to each other's phase of Pause in time.

The LAU and the LAU-model are rough, simplified, inaccurate and sketchy description of learning. It intuitively combines the result of scientific research on the subject. Its aim is to synthesize theories of teaching and learning methodology at the level of practice, to create a common "language" of findings about learning.

The Informatics

To examine the methods of teaching informatics, it is essential to define the subject of teaching. Based on the analysis of the literature, I gave the following specification:

- The basic concept of informatics is the **data**.

- **Information** is the interpreted data. Information is **knowledge**, the interpretation is a **thinking process**. Information is the source of new data creation (data producer), it gives its dynamics.
- Informatics is the science of creating, storing, transmitting, modifying, and organizing data and information into a system.
- The most important concepts of informatics: **algorithm (sequence, alternative, repetition), object, component, system, model**. These concepts can be represented and interpreted as data by abstraction. Their relationship is also data.
- Interpretation is algorithmic, namely process of sequences, alternatives, and repetitions. Interpretation can therefore lead to several results, including **alternative solutions** and knowledge.
- We qualify the solutions of informatics with **efficiency**, **accuracy,** and **relevance**.
- The implementation of interpretation —as a thinking process— is the **computational thinking** defined by Wing [14].
- The practical activities of informatics are **problem solving**, production, reproduction.
- Informatics deals with **finite** data, structures, models, and systems.
- In informatics, practice does not necessarily justify the theories, but reinterprets the current problem by arguing and checking its applicability.
- Informatics contributes to the knowledge of the world with its typical way of thinking, problem-solving strategy, and solution methods; therefore, informatics is an independent science. Informatics has a great extent in interdisciplinary fields; therefore, we can say it is a 'base science'.
- The Science of Informatics will always be born, reborn. Its renewal is maintained by the data, the dynamic management of information (interpretation) as an internal source. The concepts of informatics science are getting new interpretations, and its subfields are transforming.

The Programming

Within the education of informatics, programming has a prominent role, therefore I define what is meant by programming based on the literature and educational experience.

- The concept of programming is related to coding; the expression of thought, which requires language, as a medium of communication.
- Programming is communication between the machine and creators of the program.

- Programming is an activity when we describe data and algorithms in such a way that it can be interpreted indirectly (or directly) for the machine and determine its operation.
- Programming is the implementation of Informatics science on a machine.
- Programming is the art of using language and medium —art of communication—. The programming language is artificial, its grammar has mathematical roots; the formation of a program can be derived from the science of mathematics.
- Programming is a machine implementation, so it is derived from the engineering sciences.

Programming is an integral part of Informatics, but Informatics is more because it is not just about problems that can be solved with a machine. On the other hand, it is typical that IT problems can be modelled on computer by programming.

Programming is a tool to develop and learn the digital world. Since it is possible to build and program analogue machines, programming and IT cannot be limited to the "digital world".

Digital Literacy

Knowledge of the use of digital tools programmed or programmed for self-learning. Digital literacy refers primarily to the ability to use software and the services provided through it.

THESES

I. In its thinking methods and tools Informatics science forms a unified system, thus its successful teaching requires a science specific teaching methodology.

Analysing the sources, I found that computer science and engineering informatics represent different paradigms, which form a dialectical unit in the definition of the science of informatics given above. The two ways of thinking are equally important, they are complementary. It follows that both paradigms (in the case of further ones) must be taught in education. In the teaching of informatics, the contradictions between paradigms can be resolved by interpreting specific tasks, which are characterized by perspective of efficiency, relevance, and accuracy.

II. Computational thinking, application skills and programming skills can be developed together by teaching all the listed topics in the informatics curriculum with Informatic science.

In informatics education, the separation of digital literacy and programming, as well as the roles of user and creator or programmer, implementer, and organizer, is flawed. Digital literacy and programming —and other smaller topics— should be taught in the context of developing computational thinking. I have presented the methods of integrated teaching in several articles describing “indirect teaching of programming”. Users of applications are also creators; the creator of a program is a user of an application created on a lower level of abstraction. Consequently, the development of user and implementer competencies is related to the development of programmer, and organizer competencies. I have published tasks with the guideline “guess the code” as educational tools of the implementation, in which the problem related to digital literacy requires knowledge related to programming.

The success of using teaching methods focused on computational thinking can be seen in the results of my students' user competitions, in the graduation results, and in the feedback of my students and university lecturers.

III. The LAU-based description, and as a combination of these, the LAU model is suitable for describing and characterizing the elements of knowledge and skills related to informatics, as well as the learning and teaching processes.

I used the LAU to qualify curricula and syllabi, to formulate teaching and learning aims, to detect problems that arise during the learning process, and to accurately determine the outcome (expectations). With the LAU-based description, I presented why some curricula are not suitable for achieving the desired goal. It provided the methodological basis for reversing the order of laboratory and practice in the subject Basics of Programming on BME-VIK. The LAU was the basis for formulating the progress options for the Progress Log. I use it effectively in my daily teaching practice, in the communication about the student's knowledge: in the field of student self-assessment and evaluation, in consultations with the parents, and in discussions with colleagues.

IV. The development and the practice of computational thinking skills, as well as programming skills are facilitated or blocked by motivation, emotions and mental state as a catalyst.

Early childhood (1–3 years) is the age of self-regulation and the development of autonomy. When a child learns to walk, he sees a pattern for walking, but he must solve the problem of standing up creatively in terms of both situational awareness and “control” of muscles.

There are / were students in each group who “cannot learn to program” who are unable to solve the spreadsheet task… For those I have had the time to teach, I tried to find out what “cannot be learned” is mean. During the research, the student had to solve a task by own, only getting help from me. Direct control and questions gave me a way to get an accurate view of the status of the LAUs needed for the solution, my answers, usually in the form of questions, helped to develop the proper level. In these cases, we searched for what was unlearned until the task was solved by the student. Experiments revealed what causes the barrier: usually fear of self-determination or making mistakes; fear of falling short of peers; in individual cases, lack of concentration or impaired learning ability. Since I did not help but encouraged the detection of obstacles, the students gained experience in overcoming the obstacle on his own.

My students, who I teach programming, first write, and test a program that requires a simple algorithm and the use of a simple logical expression on their own. Success depends on being able to persuade the student to study and persevere until the end of the planned period.

In pedagogical practice, —as a consequence of the theses— the method of teaching informatics and programming must be largely adapted to the individual, to the student.

The experience gained during the teaching of controlled independent problem-solving groups is that the way of problem solving, the method of data interpretation and model creation differ from individual to individual. Not only the gaps, but also in the LAU model of required knowledge, everyone has a different condition. The creativity, the independent creation is built on this condition and will naturally be unique. In teacher-led education, the teacher prescribes what "comes to the student's mind," which inhibits, or at least controls the independent thought. Developing computational thinking requires the student to express their own thoughts, and then test them in a specific situation; then discuss the result with the teacher or peers, and check. Each one's own idea is unique, so the result is unique, and the learning paths are also individual.

EXPLANATION

The process that characterizes my research begins with the detection of the problem. This is followed by a detailed observation of the problem, the study of the related literature, then I develop and test a method to solve the problem, and the analyse the result. I examined the perceived problems from the point of view of the informatics professions, the educational science, the teachers, and the students; I interpret the solutions in this four-dimensional space. The definitions and theses formulated here are the most general results of the research process, and the relationships between them are complemented by several additional concepts and statements. I describe in detail the problems experienced and how the source analyses, case studies, questionnaire surveys, and teacher practice carried out to achieve the goals; how it contributed to the formulation of definitions and theses, and to the solution of the problems. The dissertation is supplemented by a detailed presentation of specific source analyses, general practices, case studies of individual but typical cases, and a several list of individual examples.

References (selection)

1. FALUS Iván (szerk): Didaktika – Elméleti alapok a tanítás tanulásához, Nemzeti Tankönyv-kiadó (1998); ISBN: 9631890759
2. Csíkszentmihályi Mihály: Flow. Az áramlat - A tökéletes élmény pszichológiája. (1991) Ford. Legéndyné Szabó Edit. Akadémiai Kiadó Bp. (2001) ISBN: 9630577704
3. D. KAHNEMAN: Thinking, Fast and Slow, New York: Farrar, Straus; Giroux. (2011). ISBN: 9780374533557, DOI: 10.19232/uv4pb.2016.1.92
4. POLYA, G. (1957) How to Solve It. A New Aspect of Mathematical Method. 2nd Edition, Princeton University Press, Princeton
5. LÉNÁRD Ferenc: A problémamegoldó gondolkodás. Akadémiai Kiadó, Budapest (1984)
6. M. NÁDAS Mária: A projektoktatás elmélete és gyakorlata. Magyar Tehetségsegítő Szerve-zetek Szövetsége, Magyarország, (2010)
7. SZLÁVI Péter, ZSAKÓ László: Mi az informatika szakmódszertan? In: Informatika oktatása. ELTE IK, (2012);
8. Jaap M.J. MURRE, Joeri DROS: Replication and Analysis of Ebbinghaus' Forgetting Curve. In: PLoS One Vol: 10, No 07; (2015)
9. Dr. Thomas GORDON: T.E.T – A tanári hatékonyság fejlesztése. Assertiv Kiadó, (1998)
10. Donald KNUTH: Computer Science and Its Realtion to Mathematics. In: The American Mathematical Monthly Vol. 81, No. 4 (Apr) (1974)
11. PAPP László (ford), HE Shaoyi: Az informatika fogalma. In: Tudományos és Műszaki Tájé-koztatás, Vol 50. No 9-10 (2003)
12. Edsger W. DIJKSTRA: A Case against the GO TO statement (letter EDW 215) In Commun, ACM (11) (1968); 3,
13. Robert C. MARTIN: Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall PTR, USA (2008) ISBN: 9780132350884
14. Jeannette M. Wing: Computational Thinking. Communication of the ACM Vol 49 No 3, p: 33–35 (2006) DOI: 10.1145/1118178.1118215
15. Jelentés a Magyar Közoktatásról sorozat. ISSN: 1219-8692

16. CSÉPE Valéria, OKTATÁS 2030 TANULÁSTUDOMÁNYI KUTATÓCSOPORT: A Nemzeti alaptanterv tervezete 2018. augusztus 31. 256-268. oldal (2018)
17. DOS – Magyarország Digitális Oktatási Stratégiája (A Kormány-előterjesztés melléklete) 2016. június 30. Elfogadva: 1536/2016. (X. 13.) Korm. határozat
18. P. SZLÁVI és L. ZSAKÓ: Methods of teaching programming. In: Teaching Mathematics and Computer Science; Institute of Mathematics, University of Debrecen; vol. 1, no. 2
19. Robert C. MARTIN: The Clean Coder: A Code of Conduct for Professional Programmers ISBN: 9780137081073 / Rézműves László ford: Túlélőkönyv programozóknak: Hogyan váljunk igazi szakemberré? Kiskapu Kiadó Budapest (2011)

Author's Related Publications

1. SZALAYNÉ TAHY Zsuzsanna, CZIRKOS Zoltán: Az Informatika Más… In: KERESZTES Gábor (szerk.) Tavaszi Szél 2015 / Spring Wind 2015 Konferenciakötet: III. kötet Eger, Magyarország, Líceum Kiadó, (2015) pp. 207-220. ISBN: 9786155509964
2. SZALAYNÉ TAHY Zsuzsanna, CZIRKOS Zoltán: "ProgAlap" és ami mögötte van. In: INFODIDACT 2015. Webdidaktika Alapítvány, (2015); pp. 1-16 (#14) ISBN: 9789631238921
3. SZALAYNÉ TAHY Zsuzsanna, CZIRKOS Zoltán: Linear search – the breaks in teaching-learning practice. In: XXIX. DIDMATTECH 2016 Budapest: ELTE IK, (2016); pp. 66-70.
4. SZALAYNÉ TAHY Zsuzsanna, CZIRKOS Zoltán: A Lineáris Keresés Buktatói. In: ZSAKÓ, László; SZLÁVI, Péter (szerk.) INFODIDACT, Zamárdi, Webdidaktika Alapítvány, (2016); ISBN 9786158060806
5. SZALAYNÉ TAHY Zsuzsanna, CZIRKOS Zoltán: Így írtok Ti… lineáris keresést. In: ZSAKÓ, László (szerk.) INFO ÉRA 2016 Zamárdi: Webdidaktika Alapítvány, (2016);
6. SZALAYNÉ TAHY Z, CZIRKOS Z: "Why Can't I Learn Programming?" The Learning and Teaching Environment of Programming. In: ISSEP 2016; LNCS 9973; Springer International Publishing; pp. 199-204., 6 p. (2016); ISBN: 9783319467474
7. SZALAYNÉ TAHY Zsuzsanna: Mi van, ha nem tudok teát főzni? In: Dr. Szlávi Péter, Dr. Zsakó László (szerk): INFODIDACT 2013, Informatika Szakmódszertani Konferencia; Zamárdi, Magyarország: Webdidaktika Alapítvány, (2013);
8. SZALAYNÉ TAHY Zsuzsanna (szerk): Beszámoló az Élenjáró Gimnáziumok Igazgatóinak Grémiuma (ÉGIG) szervezésében megtartott Informatika és technika munkaközösségi találkozóról; Bp., Békásmegyeri Veres Péter Gimnázium 2018.01.10. (2018)

9. Zsuzsanna Szalayné Tahy: Teaching Programming Indirectly with “Paint”. In: Matevž Jekovec (eds): The Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives — ISSEP 2015. University of Ljubljana, Faculty of Computer and Information Science, Ljubljana (2015) Page: 67., ISBN 9789616209878
10. Zsuzsanna Szalayné Tahy: How To Teach Indirectly – Using Spreadsheet Application. In: Acta Didacta Napocensia, Cluj-Napoca, Romania, ISSN: 2065-1430 Vol: 9. Nº: 1. p:15–22 (2016)
11. Zsuzsanna Szalayné Tahy: Guess the code of conditional summation. In: Proceedings of the 10th International Conference on Applied Informatics Eger, Hungary, 2017.01.30–02.01, pp. 279–284; (2017);
12. Zsuzsanna Szalayné Tahy, Zoltán Czirkos: Progress Log for Mentoring Programming Education. In: Proceedings of International Conference on Informatics in Schools: Situation, Evolution and Perspectives — ISSEP 2017, Helsinki; November 13-15, 2017;
13. Szalayné Tahy Zsuzsanna, Czirkos Zoltán: Programozástanulás Mentorálása Haladási Naplóval. In: INFODIDACT 2017. Webdidaktika Alapítvány, pp. 1–12 (#19) (2017) ISBN: 97896158060813
14. Zsuzsanna Szalayné Tahy, Zoltán Czirkos: The Two Worlds of Programming. In: Veronika Stoffová (eds): New Methods and Technologies in Education and Practice – XXXth DidMatTech; Trnava University in Trnava Faculty of Education, Trnava pp. 59–67; (2017); ISBN: 9788056800294
15. Zsuzsanna Szalayné Tahy: Guess the Code. In: The 9th International Conference on Informatics in Schools, 2016.10.13-15. Münster, Germany (2016).
16. Szalayné Tahy Zsuzsanna: Guess the Code! In: Magyar Science On Stage Fesztivál 2018.10.05-07. Szeged, Agóra. (2018)