IV INTERNATIONAL DOCTORAL CONSORTIUM ON INFORMATICS AND INFORMATICS ENGINEERING EDUCATION RESEARCH: SCOPE, METHODS, AND VALIDATION

Vilnius University Institute of Mathematics and Informatics in cooperation with
Research Council of Republic of Lithuania
Lithuanian Computer Society

Organisers:
Prof. dr. Valentina Dagienė, chair
Dr. Tatjana Jevsikova
Vladimiras Dolgopolovas
Inga Žilinskienė

3–7 December, 2013
Druskininkai, Lithuania

ALL PARTICIPANTS

Prof. **Erik Barendsen**, Radboud University Nijmegen
Prof. **Juris Borzovs**, University of Latvia
Prof. **Andrej Brodnik**, University of Ljubjana, Slovenia
Prof. **Robin Clark,** Aston University, UK
Prof. **Valentina Dagiene,** Vilnius University
Prof. **Gerald Futschek**, Vienna University of Technology, Austria
Dr. **Tatjana Jevsikova**, Vilnius University
Dr. **Eugenijus Kurilovas**, Vilnius University
Prof. **Lauri Malmi**, Aalto University, Finland
Prof. **Carsten Schulte**, Berlin Freie University, Germany
Dr. **Bronius Skūpas**, Vilnius University, Vilnius Liceum
Dr. **Mary Webb**, King's College London, UK

Vladimiras Dolgopolovas, PhD student, Vilnius University, Lithuania
Natasa Grgurina, PhD student, Groningen University, the Netherlands
Natalija Ignatova, PhD student, Lithuanian University of Educational Sciences
Violeta Jadzgevičienė, PhD student, Lithuanian University of Educational Sciences
Eglė Jasutė, PhD student, Vilnius University, Lithuania
Radovan Krajnc, PhD student, Maribor, Slovenia
Nataša Kristan, PhD student, University of Ljubljana, Ljubljana, Slovenia
Michala Křížová, PhD student, University of South Bohemia, České Budějovice, Czech Republic
Mareen Przybylla, PhD student, Institute of Computer Science, University of Potsdam, Germany
Lennart Rolandsson, PhD student, KTH, Royal Institute of Technology, Stockholm, Sweden
Jūratė Urbonienė, PhD student, Vilnius University, Lithuania
Inga Žilinskienė, Vilnius University, Lithuania

The Fourth International Doctoral Consortium (DC) is organized by Vilnius University Institute of Mathematics and Informatics on December 3–7, 2013 in Druskininkai, Lithuania. The DC provides an opportunity for doctoral students to explore and develop their research interests in a workshop under the guidance of distinguished senior researchers. We invite students who feel they would benefit from this kind of feedback on their dissertation work to apply for this unique opportunity to share their work with students in a similar situation as well as senior researchers in the field. We welcome submissions from students at any stage of their doctoral studies.

The DC has the following objectives:

- Offer a friendly forum for doctoral students to discuss their research topics, research questions and design in the field of computing education / educational technology – informatics engineering and education
- Provide a supportive setting for feedback on students' current research and guidance on future research directions
- Offer each student comments and fresh perspectives on their work from researchers and students outside their own institution, as well as help with choosing suitable methodology and strategies for research
- Support networking with other researchers in the informatics engineering education research field, and promote the development of a supportive community of scholars and a spirit of collaborative research
- Support a new generation of researchers with information and advice on research and academic career paths

Preparing and Submitting your DC Proposal

Each participant should submit an Extended Abstract of two A4 pages research description covering central aspects of your PhD work, which follows the structure, details and format specified in the submission template (VU2014-dc-template.doc). Key points include:

1. Your situation, i.e., the university doctoral program context of your work
2. Clear formulation of the research question(s)
3. Very short background/literature review of key works that frames your research
4. Identification of the significant problems in the field of research
5. Research objectives/goals
6. Sketch of the applied research methodology (data collection and analyzing methods)
7. Results to date and their validity
8. Current and expected contributions

The summary will be made available for other participants of the DC to allow providing feedback and preparing questions on the research. Since the goals of the DC include building community, participants will be expected to read all of the Extended Abstracts of your colleagues prior to the beginning of the consortium with a goal of preparing careful and thoughtful critique.

Upon Acceptance of your DC Proposal

Each student will present his or her work to the group with substantial time allowed for discussion and questions by participating researchers and other students. Each student should bring a poster for sharing with mentors and other students during the DC sessions. Students will have possibility to rethink and improve their submissions according to suggestions of senior researchers or other participants. The corrected posters will be presented at final session.

AGENDA

Tuesday, December 3

13:00 – 17.45	Arrival in Vilnius airport and transfer to Druskininkai
20:00 – 23.00	Welcome and discussions

Wednesday, December 4

07.30 – 09.00	Breakfast
09.00 – 11.00	Presentation and discussion by Prof. Dr. Lauri Malmi (Aalto university, Finland). An Overview of Computing Education Research – Research Topics and Research Processes
11.00 – 11.30	Coffee break
11.30 – 12.30	5 min challenge – Presentation posters and key research questions by doctoral students (individual level). Chaired by Prof. Dr. Valentina Dagienė
13.00 – 14.00	Lunch
14.00 – 17.30	Discussion in groups organized by Dr. Mary Webb (King's College London, UK). How can qualitative research techniques support research in computer science /ICT education research? With coffee break at 16.00
15.30 – 16.00	Coffee break
18.00 – 19.00	Dinner
19.00 – 22.00	Late discussions

Thursday, December 5

07.30 – 09.00	Breakfast
09.00 – 10.00	Presentation by Prof. Dr. Carsten Schulte (Berlin Freie University, Germany): On Computer Science Education Research: Reality and Future
10.00 – 10.30	Coffee break
10.30 – 12.00	Group work: Students will be divided in 3-4 groups according to their topics, the research seniors will be appointed to each group.
12.00 – 13.00	Lunch
13.00 – 15.00	Excursion or/and cryotherapy
15.30 – 16.00	Coffee break
16.00 – 18.00	Group work: Improve your poster which summarizes your research: BIG research question, goal, subtasks, data collection and analysis methods, theoretical framework, scope, and use of results.
18.00 – 19.00	Dinner
19.00 – 22.00	Late discussions in groups

Friday, December 6

07.30 – 09.00	Breakfast
09.00 – 11.00	Prof. Dr. Andrej Brodnik (Ljublana University, Slovenia). Suggestions and criteria for writing informatics education doctoral thesis
11.00 – 11.30	Coffee break
11.00 – 13.00	Reading DC posters: All participants are asked to read the posters and write down their questions and comments
13.00 – 14.00	Lunch
14.00 – 15.30	Group work. Renewing DC posters
15.30 – 16.00	Coffee break
16.00 – 18.00	Groups overview. Summaries and suggestions by group leaders. General discussion, including a brainstorming session about current and future research topics in the area. Reflection from all participants.
18.00 – 20.00	Closing dinner

Saturday, December 7

07.30 – 09.00	Breakfast
09.30	Departure to Vilnius airport

Vladimiras Dolgopolovas

1st year PhD student
Vilnius University Institute of Mathematics and Informatics

1. Research topic

Design and development of constructivists learning objects for theoretical computer science and scientific computing education.

2. Motivation and background

2.1. Motivation 1

"Theoretical computer science is a fascinating scientific discipline. Through its spectacular results and high interdisciplinarity, it has made great contributions to our view of the world. However, theoretical computer science is not the favorite subject of students, as statistics would confirm. Many students even view theoretical computer science as a hurdle that one has to overcome in order to graduate.

There are several reasons for this widespread opinion. One reason is that amongst all areas of computer science, theoretical computer science is the mathematically most demanding part and hence the lectures on theoretical fundamentals belong to the hardest courses in computer science. Not to forget, many computer science students start their study with a wrong impression of computer science, and many lecturers of theoretical computer science do not present their courses in a sufficiently attractive way. Excessive pressure for precise representation of the minute technical details of mathematical proofs plus a lack of motivation, a lack of relevance, a lack of informal development of ideas within the proper framework and a lack of direct implementation and usage, can ruin the image of any fascinating field of science." **[1]**

2.2. Motivation 2

The importance of high level research in various fields of science and engineering is increasing. The high level of computer literacy for engineers and scientists is also very important nowadays. On the other hand, the importance of computational methods in science and engineering is growing up and this influences the impact of informatics on classical fields like mathematics and science.

The traditional approach of teaching programming for non programmers is based first on studying of programming languages and next studying of programming and mathematical models. Under this approach the mathematical literacy of the learner is important as this corresponds to the complexity of the learning objects. So there is always a gap between the complexity of the programming constructions and the literacy level of the learner.

2.3. Seamless approach

G. E. Karniadakis and R.M. Kirby II offer a "*seamless* approach to numerical algorithms, modern programming techniques, and parallel computing. … "Often times such concepts and tools are taught *serially* across different courses and different textbooks, and hence the interconnection between them is not immediately apparent. The necessity of integrating concepts and tools usually comes after such courses are concluded, e.g. during a first job or a thesis project, thus forcing the student to synthesize what is perceived to be three *independent subfields* into one in order to produce a solution. Although this process is undoubtedly valuable, it is time consuming and in many cases it may not lead to an effective combination of concepts and tools. Moreover, from the pedagogical point of view, the integrated seamless approach can stimulate the student simultaneously through the eyes of multiple disciplines, thus leading to enhanced understanding of subjects in scientific computing" [**2**]. Figure 1 presents the definition of scientific computing as an intersection of Numerical Mathematics, Computer Science and Modeling [**2**].

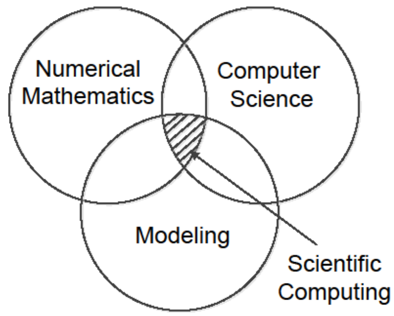


Figure 1 Scientific computing.

2.4. Constructivist learning

R. N. Caine and G. Caine in their fundamental research **[3]** propose the main principles of constructivist learning. One of the most important for us is as follows: "The brain processes parts and wholes simultaneously". So, a well-organized learning process provides details as well as underlying ideas. Using Model - Centered learning, we introduce the goal of the research after constructing a model for simulation. That allows us to observe the results and to draw relevant conclusions.

2.5. Model - centered education

A. S. Gibbons introduced Model - Centered Instruction in 2001 **[4]**. The following main principles are important:

- Learner's *experience* is obtained by interacting with models;
- Learner solves scientific and engineering *problems* using simulation on models;
- Problems are presented in a constructed *sequence*;
- Specific instructional *goals* are specified;
- All necessary *information* within a solution environment is provided.

M. Millard, J.M. Spector and P.I. Davidsen **[5]** propose Model Facilitated Learning using "interactive simulations". The authors present a modern computer technology powered by "promising methodology" based on "system dynamics". "Supportable experiences include the construction of interactive … models as well as their use *for hypothesis testing and experimentation*".

R. Lehrer and L. Schauble **[6]** refer to the experiments with different representations of the model: "Student learning is enhanced when students have multiple opportunities to invent and revise models and then to compare the explanatory adequacy of different models".

2.6. Scientific Computing Education: experiments with models

L. Xue [**7**] introduces "teaching reform ideas in the "scientific computing" education by means of modeling and simulation". He suggests "…the use of the modeling and simulation to deal with the actual problem of programming, simulating, data analyzing…". Model-Centered Learning is used in mathematics education. Plenty of models are constructed using "Geogebra" software [**8**]. Models play a central role in Science Education [**9**], [**10**].

3. Research problem questions

- What is the set of topics, methods, theoretical constructions and algorithms to be included in to the educational framework?
- In which way the relevant learning objects should be constructed?
- What are the basic requirements for the learner's background?
- What are the key instruments for modeling and experiments with models?
- What is the methodology of evaluation and testing?

4. The aim of the research

Investigate computer mathematics algorithms and theoretical constructions. Develop methodology and constructivists learning objects based on computer mathematics models.

5. Research tasks

- Structuring of computer algebra models and algorithms;
- Structuring of Queuing theory models and algorithms;
- Developing of learning objects;
- Testing of developed objects in educational process.

6. Research goals

As a result of the research the proper learning framework and learning objects should be developed to overcome the described difficulties. The models of the main advanced programming constructions like parallel calculations, declarative programming and others which are based on constructivist learning approach must be designed. Under such approach the basic mathematical constructions are introduced in parallel with programming models. The next step is conducting of experiments with computer mathematics models, verifying theoretical mathematical results. This improves computer science and mathematical literacy. This is also provides an introduction for basic mathematical topics of various difficulty levels.

7. Literature

[1] Jurai Hromkovič. Theoreticalcomputer science. Springer-Verlag, Berlin, 2004.
[2] G. E. Karniadakis and R. M. Kyrby, Parallel scientific computing in C++ and MPI. A seamless approach to parallel algorithms and their implementation , Cambridge University Press, 2003.
[3] R. N. Caine and G. Caine, Making connections: Teaching and the human brain, Association for Supervision and Curriculum Development, Alexandria, 1991.
[4] A. S. Gibbons, Model-centered instruction, Journal of structural Learning and Intelligent Systems (2001), no. 14, 511-540.

[5] M. Milrad, J. M. Spector and P. I. Davidsen, "Model facilitated learning," Instructional design, development and evaluation, vol. 3, Syracuse University Press, 2003.
[6] R. Lehrer and L. Schauble, "Cultivating model-based reasoning in science education," The cambridge handbook of the learning sciences, Cambridge University Press, 2005, pp. 371-388.
[7] L. Xue, "Modeling and simulation in scientific computing education," International Conference on Scalable Computing and Communications, 2009, pp. 577 - 580.
[8] "Model-centered learning. Pathways to mathematical understanding using geogebra," Modeling and simulations for learning and instruction, vol. 6, Sense Publishers, P.O. Box 21858, 3001 AW Rotterdam, The Netherlands, 2011.
[9] M. S. Khine and I. M. Saleh, Models and modeling,cognitive tools for scientific enquiry, Models and modeling in science education, vol. 6, Springer, 2011, p. 290.
[10] J. Clement and M. A. Rea, Model based learning and instruction in science, vol. 2, Springer Netherlands, 2008.

Computational Thinking Skills in Dutch Secondary Education

Nataša Grgurina
University of Groningen
PO Box 800, 9700 AV Groningen
The Netherlands
n.grgurina@rug.nl

Erik Barendsen
Radboud University Nijmegen
PO Box 9010, 6500 GL Nijmegen
The Netherlands
e.barendsen@cs.ru.nl

Bert Zwaneveld
Open University
PO Box 2960, 6401 DL Heerlen
The Netherlands
g.zwaneveld@uu.nl

Wim van de Grift
University of Groningen
PO box 800, 9700 AV Groningen
The Netherlands
w.j.c.m.van.de.grift@rug.nl

Idzard Stoker
Radboud University Nijmegen
PO Box 9010, 6500 GL Nijmegen
The Netherlands
i.stoker@student.ru.nl

ABSTRACT
Computational Thinking is regarded as a necessary analytical skill for young people in the present day information society. We report on an ongoing design research project on Computational Thinking (CT) skills in Dutch secondary computer science (CS) education. The first phase of the project investigates the occurrence and nature of typical CT aspects in existing CS teaching materials, teacher's pedagogical content knowledge and policy documents. In the poster we focus on the overall research design and on the method and preliminary results of the first phase.

Categories and Subject Descriptors
K.3.2 [Computers and Education]: Computer Science Education.

General Terms
Human factors.

Keywords
Computational thinking, pedagogical content knowledge, problem-solving, computer science education.

INTRODUCTION

The term Computational Thinking was coined in 2006 by J.M. Wing, who asserted that "to reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" [10]. Since then, educators have recognized this need and have inquired into the precise description of this concept and the ways to teach it. The Computational Thinking Task Force of CSTA suggested an operational definition of CT tailored to the needs of K-12 education [3]. This definition covers a non-exclusive set of characteristic skills essential for a problem-solving process. These involve typical problem-solving aspects such as (re)formulating problems in a way that enables us to use a computer and other tools to help solve them, as well as CS aspects such as representing data through models and simulations, and automating solutions through algorithmic thinking

In 2012, The Royal Netherlands Academy of Arts and Sciences (KNAW) published the report *Digital Literacy in Secondary Education* [6] containing a number of recommendations concerning education in digital literacy and CS. One of the suggestions in the KNAW report is to have CT play a central role in a prospective new digital literacy course and a revised CS course. The report mentions some CT-aspects, but does not completely characterize the intended nature of CT itself.

Current CT research focuses on characterizing CT skills and identifying ways of teaching and assessing them. In present CS education one can find some ‘good practices’ of CS assignments that involve typical CT problem-solving aspects. For example, a typical assignment would require students to design a model of traffic lights for a busy crossing or elevators for an apartment building. Both these problems: (1) are open and can have various correct solutions, (2) come with a minimal specification, (3) originate from the real world; (4) solving (modeling) them necessitates the use of information-processing agents – like those used within a ‘regular’ CS course and (5) are ill defined: some aspects of the problem, such as what one is supposed to do, are vaguely stated [9]. In order to solve the problem, one needs to generate “a problem representation or problem space (the problem solver’s view of the problem)” [4]. One can observe that the CT occurrences in CS education lack coherence. Moreover, CT skills are not explicitly specified as learning objectives in the CS curriculum.

RESEARCH CONTEXT

The research reported here is a part of an ongoing bigger project on CT in Dutch secondary CS education, described in an earlier paper [5]. The general questions of the project are:

- What is a suitable operational definition of Computational Thinking problem-solving skills, tailored to the specific situation and needs of secondary education in the Netherlands?
- How can students’ CT problem-solving skills be assessed?
- What is a suitable pedagogical approach to teaching students and stimulating their learning of CT problem-solving skills?

The overall project is conducted in four phases. The first phase investigates essential aspects of CT and examines their occurrence in existing teaching practice. In the second phase an instrument for the assessment of students’ CT problem-solving skills will be developed. The results of these two phases will yield the data for the pedagogical approach and curriculum intervention that will be developed in the third phase. In the fourth phase, the effects of the curriculum intervention will be assessed in an experiment on a larger scale and the final version of curriculum intervention (i.e. teaching materials) will be developed.

RELEVANCE

There has been little, if any, research into effects of pedagogical content knowledge (PCK) [8] and teachers’ instructions on CT problem-solving skills of their students. This research will provide an assessment tool to make students’ learning of CT visible and a validated assessment instrument to measure students’ CT.

The first phase of research will give an insight into the current state of CT education in the Netherlands concerning teaching materials, teaching practice and teachers’ PCK. The results of this research will aid the curriculum development of the prospective new digital literacy course and revised CS course in the Netherlands as well.

Table 1: Refinement of CT aspects

Category	Subcategory
Data Collection	Collecting data Selecting relevant data
Data Analysis	Drawing conclusions Finding patterns Making sense of data
Data Representation	Arrange data for analysis Organize/represent data
Problem decomposition	Breaking down tasks Merging subtasks
Abstraction	Finding characteristics Creating models
Algorithms & procedures	Making sequential steps in a specific order Understanding and changing algorithms Making decisions in algorithms Implementing algorithms
Automation	Recognizing different forms of automation Recognizing the advantages of automation
Simulation	Creating pseudo-code Creating models of processes Experimenting
Parallelization	Combine/merge activities

AIM OF THE STUDY

The present study is part of the first phase of the project. It focuses on CT aspects in the existing teaching practice. The research questions are:

- Which aspects of CT can be recognized in Dutch CS teaching materials, curriculum specifications and policy documents?
- How can the pedagogical content knowledge of CS teachers in Dutch secondary education be characterized?

METHOD AND PRELIMINARY RESULTS

Starting point was the CSTA characterization of essential CT aspects (data collection, data analysis, data representation, problem decomposition, abstraction, algorithms & procedures, automation, simulation and parallelization). Using the CSTA examples of learning experiences [2] and samples of existing teaching materials, we iteratively constructed a refinement of the CT characterization, see Table 1. This refinement will be used as a coding scheme to analyze the occurrences and nature of CT in CS textbooks (see [1] for preliminary results), curriculum and policy documents. It will also be used to analyze CS teachers' PCK on CT. The latter will be established through semi-structured interviews based on the Content Representation (CoRe) instrument [7].

REFERENCES

[1] Barendsen, E. & Stoker, I. (2013). Computational thinking in CS teaching materials: a pilot study. In Koli Calling 2013 Proceedings. To appear.

[2] CSTA. Computational thinking teacher resources, second edition, 2011. Retrieved from http://csta.acm.org/ Curriculum/sub/CompThinking.html, July 2013.

[3] CSTA, http://csta.acm.org/Curriculum/ sub/CompThinking.html, retrieved February 2012

[4] Gick, Mary L. (1986). Problem-Solving Strategies. *Educational Psychologist*, 12(1&2), 99-120.

[5] Grgurina, N. (2013). Computational Thinking in Dutch Secondary Education. In I. Diethelm,, J. Arndt, M. Dünnebier, & J. Syrbe, (Eds.) *Informatics in schools: local proceedings of the 6th International Conference ISSEP 2013; selected papers.* Oldenburg, Germany.

[6] KNAW. (2012). Digitale geletterdheid in het voortgezet onderwijs: vaardigheden en attitudes voor de 21ste eeuw. Amsterdam: Koninklijke Nederlandse Akademie van Wetenschappen.

[7] Loughran, J., Mulhall, P., & Berry, A. (2004). In search of pedagogical content knowledge in science: Developing ways of articulating and documenting professional practice. *Journal of Research in Science Teaching*, *41*(4), 370–391.

[8] Magnusson, S., Krajcik, J., & Borko, H. (1999). Nature, sources, and development of pedagogical content knowledge for science teaching. In J. Gess-Newsome & N. G. Lederman (Eds.), *Examining pedagogical content knowledge* (pp. 95–132). Dordrecht: Kluwer.

[9] Robertson, S. I. (2003). *Problem solving.* Psychology Press.

[10] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35

Natalija Ignatova

4th Year of doctoral studies
Lithuanian University of Educational Sciences
Suvalkų 1
LT03106 Vilnius, Lithuania
natalija.ignatova@itc.smm.lt

1. **Research area: Personalized learning**
   Title of the research: Evaluation of ICT based learning activities enabling personalized learning in basic school

The main problem is to identify what ICT based learning activities enable learning personalization in basic education that will be examined and described in terms of structured evaluation criteria of best educational practice in Lithuanian schools that enable learning personalization.

The aim of research is to identify key criteria of innovative ICT based LA that could help to improve learning process from the teachers', who implemented iTEC project LA, point of view. While aiming to describe and separate learning activities that enable students personalising learning, authors compared discerned key criteria of innovation and main features of P-route in "Personalisation by Pieces" approach elaborated by D. Buckley (2011).

An outline of the current knowledge of the problem domain.

There is different understanding of personalization strategies. "Within 21st-century parade of change, the notion of personalization in education is moving to the forefront. It's an ambiguous and often broadly defined notion that has been hotly contested in the United Kingdom over the past several years." (McRae, 2010) Some authors are "paying attention to LA suitability to particular learners' groups" (Kurilovas, et al., 2011), or styles (Popescu, 2009).

We are paying attention to general LA features that enable students to personalize learning and which are in line with "five components of personalised learning to guide policy development. i) It needs assessment for learning and the use of data and dialogue to diagnose every student's learning needs. ii) It calls for the development of the competence and confidence of each learner through teaching and learning strategies which build on individual needs. iii) It presupposes curriculum choice which engages and respects students. iv) It demands a radical approach to school organisation and class organisation based around student progress. v) Personalised learning means the community, local institutions and social services supporting schools to drive forward progress in the classroom." (Miliband, 2006). While looking for personalisation enablers on individual teacher – student's level, we refer to the Sanna Järvelä "Personalised learning? New Insights into Fostering Learning Capacity". She examines seven critical dimensions: i) development of key skills which are often domain-specific; ii) levelling the educational playing field through guidance for improvement of students' learning skills and motivation; iii) encouragement of learning through "motivational scaffolding"; iv) collaboration in knowledge-building; v) development of new models of assessment; vi) use of technology as a personal cognitive and social tool; vii) the new role of teachers in better integration of education within the learning society (Järvelä, 2006). We agree with the "Personalisation by Pieces" approach elaborated by D. Buckley, and focus on Personalisation by the Learner rote (P-route) described by the REORDER applied model aspects (Buckley, 2011) while elaborating students own learning personalisation criteria. These recent notions are responding to Lithuanian educator M. Luksiene point of view, where she focused on importance of interaction between culture and personality (Lukšienė, 1985), and correspond to the Lithuanian national curricula that aimed to “awake nurture of each student, their creativity, according to his/her inclinations, abilities and holistic personal development” (Bendrosios programos, 1994).

Advances beyond the state-of-the-art in terms of your specific contribution and research plan. (A description of the Ph.D. project's contribution to the problem solution)
Object of the research is innovative ICT based learning practice as well as its key characteristics that could be applied for the evaluation of particular learning activities. As authors are interested to describe innovative practices according to the teachers’ points of view, they choose analysis of teachers’ interviews as part of the above mentioned best educational practice case studies.

2. A presentation of any preliminary ideas, the proposed approach and achieved results

Current status of the research plan
A sketch of the applied research methodology (data collection and analyzing methods)
Evaluation criteria for success in iTEC project are reviewing throughout all project cycles using both qualitative and quantitative data for the scenario evaluation and development of scenarios during the later cycles. Qualitative data analysis of the teachers’ semi structured interviews about scenario implementation practice in Lithuania was used. Structured template was proposed for the structured teachers’ interviews which were supplemented by the extended interviews on the main features of case study (i.e. semi-structured interviews). After the semi structured interviews with 3 Lithuanian teachers who piloted 1st cycle scenario of iTEC project, key characteristics and innovations related with them were identified. Narrative analysis of teachers’ interviews showed main aspects of new practice that all teachers had noticed. The main aspects were considered as key characteristics or categories that should describe innovative practice used (Bitinas, et al., 2008). Comparative matrix between the categories of the interviews’ analysis and the aspects of the REORDER model was used for the description of LS and technology based learning activities that enable own learning personalisation. Semi structured interviews performed with 4 teachers, who piloted 2nd and 3rd cycles iTEC project scenarios, let to validate LS describing categories, which compared with P-route features described by the REORDER applied model aspects (Buckley, 2011)

Expected achievements and possible evaluation metrics to establish the level of success of your results
Most common or key characteristics of the learning scenario have been indicated: 1) variety of students’ and teachers’ roles, 2) mixed environment, 3) opportunity to choose research topic and level of curriculum, 4) diversity of learning resources and technologies used, 5) recognition of students’ competences and skills, 6) the level of students’ and teachers’ motivation and responsibility for implementation of the goals envisaged and learning activities planned, 7) school principals support, 8) growing learning community. These characteristics are corresponding to the REORDER model aspects of the applied “Personalisation by Pieces” approach elaborated by D. Buckley (2010), and it will be used as learning activities evaluation criteria.

3. Bibliographic References

1. Kurilovas, E.; Žilinskiene, I.; Ignatova, N. Evaluation of Quality of Learning Scenarios and Their Suitability to Particular Learners’ Profiles. In: *Proceedings of the 10th European Conference on e-Learning (ECEL’09)*. Brighton, UK, November 10–11, 2011, pp. 380–389.

2. N. Ignatova, E. Kurilovas. ICT based teaching and learning personalization trends in the context of Lithuanian education. *PEDAGOGIKA*, ISSN 1392-0340. 2012. 106.
3. N. Ignatova, A. Buinevičiūtė. Evaluation of iTEC project innovative practice of ICT based learning scenario from teachers' perspective. In: *Proceedings of the ATEE Spring University 2012 conference: 20 Years for Sustainable Development: Learning from Each Other*, May 3-5, 2012, pp. 130-135.
4. Natalija Ignatova, Valentina Dagienė, Virginija Birenienė, Asta Buinevičiūtė, Alvida Lozdienė. ITEC Project Scenarios Evaluation from the Perspective of Learning Personalisation. In: *Proceedings of the Conference "Computer Days – 2013"*: *11th Conference on Informatics in School Education*. Vilnius, September 19-21, 2013, pp. 119-130 (available at http://www.zara.lt/e-knygos/e-KoDi-2013-metod.pdf).

4. Expectations and motivation to attend doctoral consortium

I expect to share my preliminary research results and to contact more experts in order to find better ways of personalized learning evaluation.

Violeta Jadzgevičienė

Year of your doctoral studies:4
Lithuanian University of Educational Sciences
Studentų 39
Vilnius, Lithuania
violeta.jadzgeviciene@leu.lt

1. Research area

The role of learning styles in programming training

The aim of research: to improve programming teaching by testing and developing teaching programming methods and techniques in relation to students learning styles.

Research Questions:

1. What programming teaching methods and techniques are more suitable for the learning styles of higher education students?
2. How virtual learning environment can help improve the introductory programming training, taking into account the different learning styles?

Programming is a fundamental part of computer science curriculum, but it is often problematic. Due to its complexity programming seems to be not very attractive, that is why, in order to engage the learners, it is necessary to present it as easy as possible, in a clear and attractive way. Many researchers propose methodologies and tools to help students. Although some of these tools have been reported to have a positive effect in students learning, the problem still remains mostly unsolved. There are five components leading to difficulties in programming learning: methods of training, learning techniques, learning skills and attitudes, the nature of programming, and psychological reasons (Gomes, Mendes, 2007).

Different students have different learning styles and can have several preferences in the way they learn. Some may regard learning as a solitary process while others may prefer a more dynamic learning environment, for instance through discussions with their peers. Additionally, some subjects may demand a particular learning approach but, without guidance, students will tend to adopt the style they prefer or which has served them best in the past. It is an important responsibility for the teacher to ensure that the students adopt the most appropriate learning approach for the subject at hand (Jenkins, 2002, Gomes, Mendes, 2007).

In this research, the researcher has chosen the learning style by Honey & Mumford's theory. They divided the learning style into 4 types to provide the method that according with learner's learning style as follow: Activist – prefers doing and experiencing, Reflector – observes and reflects, Theorist – wants to understand underlying reasons, concepts, relationships, Pragmatist – likes to have go try things to see if they work.

The suitable activities of the four learning styles are summarized in the following table 1 (Mobbs, 2005).

If there is a purpose to organize the learning based on learning styles, at the beginning of learning activities the student's learning styles are determined by using a special test. Both, the teacher and the learner, can plan their own learning activities taking into account the inherent style characteristics. In terms of a certain learner the above described classification of learning methods according to learning styles is not absolute. Taking into account personal qualities of a certain learner, he might like to use those methods which, in accordance with his identified learning style, are not appropriate for him. Also, the learner is usually characterized by several learning styles, with one of them to be more expressed.

Table 1. Learning style and activities

Learning style	Activities
Activist	Brainstorming, problem solving, group discussion, puzzles, competitions, role-play
Theorist	Models, statistics, stories, quotes, background information, applying theories
Pragmatist	time to think about how to apply learning in reality, case studies, problem solving, discussion,
Reflector	Paired discussion, self-analysis questionnaires, personality questionnaires, time out, observing activities, feedback from others, coaching, interviews

Virtual environment tools (one of the most popular environment is Moodle) are implemented in parallel with traditional lectures and workshops. Such platforms are commonly used to present the learning material, to test student's knowledge (self-control and knowledge control for evaluation tests), and for communication with students when they are outside classes. According to various studies blended learning is most acceptable for both students and teachers. Increasingly, it comes about the individualization of learning, tutorial adaptation of a particular learner's needs, abilities and learning goals. It becomes relevant to organize teaching process applying it to learner's individual style. Mostly virtual learning environment provide with learning material, organization of learning activities, communication and cooperation means with the students but, the standard kit doesn't meet specific programming learning needs. Active researchers, developers and Moodle users community has developed and constantly creates new and additional instrumentality.

The tasks of the research:

- To investigate and describe the problems of the introductory programming teaching in higher education, to make a comparative analysis of the literature on this subject, to formulate the key questions of the research;
- To describe the selected Honey & Mumford's student learning style classification interfaces with a programming training (learning) methods;
- To develop the pedagogical model for introductory programming learning (take into account the learning styles);
- To draw conclusions and recommendations on how it is possible to improve teaching programming take into account the learning styles in Lithuanian higher education schools.

2. A presentation of any preliminary ideas, the proposed approach and achieved results

A sketch of the applied research methodology (data collection and analyzing methods)

1. Review of scientific literature, analysis of educational documents;
2. Action research:
    - design of the introductory programming learning model (take into account the learning styles);
    - apply the introductory programming learning model;

- reveal the effective methods of this teaching and their application, and both theoretically and empirically to justify them;
- referring to results of practical application, to improve the introductory programming learning model and again apply it for teaching;

3. Data collection: individual interview, survey.
4. Data analysis and interpretation.

Subject descriptions of an introductory programming in Lithuanian higher education schools were reviewed.

Action research consists of three phases. The action research project started in 2012 fall semester. The learning (teaching) methods by using virtual learning environment tools, mentioned above, are experimentally tested with programming and internet technologies students of the Vilnius Business College according to their learning style. At the moment this is completed the last phase of the action research. The collected data will be processed during 2 months starting December 2013.

Detailed distributions of student's preferences according to Honey&Mumford LSQ are presented in the table 2.

Table 2. Distributions of student's preferences according to Honey&Mumford LSQ (n=37)

Preference	Activist	Reflector	Theorist	Pragmatist
Very strong	8 21,62%	5 13,51%	5 13,51%	1 2,70%
Strong	6 16,22%	15 40,54%	4 10,81%	5 13,51%
Moderate	12 32,43%	9 24,32%	18 48,65%	11 29,73%
Low	9 24,32%	5 13,51%	7 18,92%	12 32,43%
Very Low	2 5,41%	3 8,11%	3 8,11%	8 21,62%
Mean Score	8,95	14,16	11,97	11,24
Standard Deviation	3,76	3,33	2,86	3,29

3. Bibliographic References

1. Gomes, A. and Mendes, A. J. 2007. Learning to program - difficulties and solutions. In Proceedings of the International Conference on Engineering Education (Coimbra, Portugal, September 03-07, 2007). ICEE'07, CD-ROM.
2. T. Jenkins, "On the difficulty of learning to program", in Proc. of the 3rd Annu. LTSN_ICS Conf., Loughborough University, United Kingdom, August 2002, pp. 53-58.
3. Bennedsen, J. (2008). Teaching and Learning Introductory Programming – A Model-Based Approach. Dissertation for Dr. Philos degree in the Faculty of mathematics and Natural Sciences, University of Oslo, Norway, 2008.
4. Honey, Peter, Mumford, Alan (1992). The manual of learning styles (Vol. 3). Maidenhead.
5. Honey, Peter (2006). Learning Styles Questionnaire (40-Item). Maidenhead Berks: Peter Honey Publication Limited.
6. Honey, Peter, Mumford, Alan (1999). Learning styles questionnaire. [online]. http://www.psychpress.com.au/psychometric/products/others/LSQ%20Report%20%28Sample%29.pdf.
7. Richard Mobbs. (2005).Honey and Mumford Learning Styles. <http://www2.le.ac.uk/departments/gradschool/training/resources/teaching/theories/honey-mumford>

Eglė Jasutė

4th years PhD student

Vilnius University Institute of Mathematics and Informatics

Research topic: An interactive visualization method of constructionist teaching and learning of geometry (Computer-based model of constructive geometry proof)

The aim of research: To create an interactive visualization method of teaching and learning geometry, based on dynamic geometry.

1. Tasks

1) To explore informatics methods to improve mathematics teaching and learning, in particular - constructionist
2) To set up an interactive visualization principles
3) To create a method of interactive visualization
4) To adapt abstract data type concept for modelling scenarios of interactive visualization
5) To realize model and perform the validation

2. The significant problems in the field of research

To adapt informatics methods for teaching and learning mathematics.

3. An outline of the current knowledge of the problem domain

The problem domain of our work is intersection of fields of information visualization, informatics theory and informatics application. The external domains such as psychology, pedagogic, and mathematics education have influence on our problem too (Figure 1).

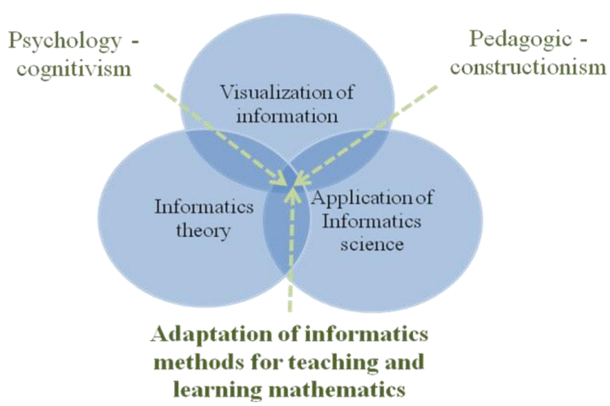


Figure 1. Problem domain

When we solve informatics problem we have to take in an account the pedagogical context of the result. The solution of problem give us too results:

- The theoretical result - method developed by adapting ADT – in Informatics;
- The practical result - an interactive digital learning tool – in teaching and learning mathematics.

Consequently the psychological and pedagogical approaches were studied. Problems related with our research of this approach were recognized and described.

Constructionist ideas can be effectively realized in mathematics lessons. However, there is still a strong focus on mathematical knowledge acquisition [DJ07]. Therefore constructionist approach is integrated into the teaching of mathematics very slowly because teachers have to adapt to the new ideas and methods, to spend more time for preparing. Teaching mathematics is mostly based on an academic approach – it is intended for the national school – leaving mathematics exam obligatory for almost every higher school. In view of that, the majority of our mathematics teachers can be considered as traditional teachers.

Some more reasons, why mathematics teachers do not use constructionist learning tools, i.e. dynamic geometry in their lessons, have been found by analyzing literature: the lack of the skills in information technology has an impact on the use of dynamic geometry for teaching mathematics [SK11]. The dynamic geometry is relatively complex for a math teacher for several reasons: first, a dynamic geometry construction is based on a hierarchy and to construct a sketch, teachers must have (or acquire) new skills of developing algorithms and programming by geometry; second, most tools of dynamic geometry software are rather complex for the teacher [HHL09]. Some scientists see quite other problem of information technology: the usage of digital tools depends on the teacher's disposition. If the teacher uses active learning and constructive methods of teaching, he/she is willing to use the dynamic geometry for teaching, if the teacher uses traditional teaching methods, he/she is not willing to use the dynamic geometry for teaching [SK11]. While there are some problems of using the dynamic geometry, the software can help teachers to use a variety of constructionist teaching and learning methods. Four methods are defined for teaching mathematics with dynamic geometry which are related more or less with the ideas of constructionism:

1) a student is constructing dynamic sketches himself by his experience;

2) a student is analyzing individually geometric concepts and properties of geometric objects in the pre-created dynamic sketches with some instructions and directed questions;

3) a student is analyzing pre-created dynamic sketches with the teacher in the class, if the teacher uses the dynamic sketch to illustrate the explanation of geometry;

4) a student is learning by a pre-created book of dynamic sketches, when he has all the sketches that consistently illustrate all the topics of geometry and can analyze them individually [Jas07].

These studies have inspired the ideas how to develop an approach making the mathematics studies easier for both students and teachers. The developed approach links together a traditional way of teaching mathematics with the facilities of uptodate media. Thus, we are not going to force teachers for quick changes, vice versa we offer them support by developing flexible interactive tools for dynamic geometry.

4. A presentation of the proposed approach and achieved results

There are three steps to create qualitative digital learning tool: first, to analyze curricula and distinguish topics for visualization, second, to write scenarios for each topic, third, to create collection of learning objects [DJ06]. Here three different domains have merged: management, technology and mathematics education. The competencies of three different domains are required to create qualitative learning resource. The presented three steps have to be extended and detailed in the activity diagram (Figure 2). The competence of mathematics educations lets to review mathematics curricula and select topics for visualization. The technological sophistication lets to introduce facilities in technology (in this case dynamic geometer) and to prepare templates for experts of mathematics education. The model of templates requires pedagogical and mathematical knowledge too. Then the templates have to be realized in some application (dynamic geometry, Java ant et.). It can be done by modelling scenarios from templates and realizing them in dynamic geometry. The result of some scenario has to be tested by the mathematics education experts and improved if it requires some corrections. All these activities can be organized by publisher (Figure 2).

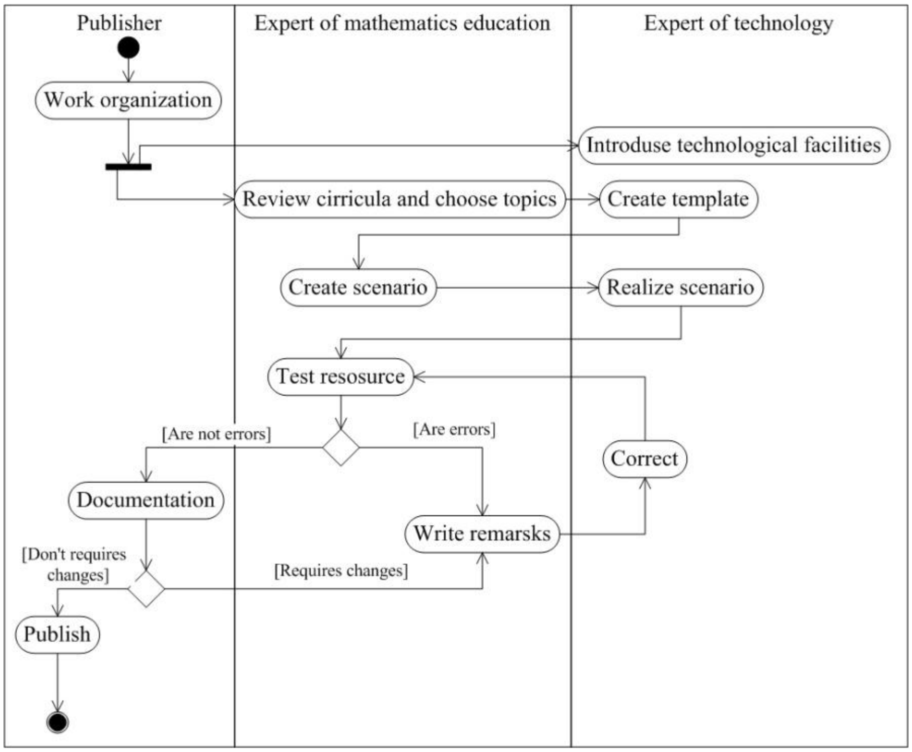


Figure 2. UML activity diagram of creating interactive images

The first thing to began visualize geometry is to know what result has to be. For this the principles of the interactive geometry visualization have to be described. Having in mind criteria of software quality [ISL02] we have grouped principles in three groups what represents three levels of investigation digital learning tool: 1) the general principles of all the digital learning tool; 2) the principles of separate parts of tool – an interactive image, and 3) the principles of the one part of the image – dynamic drawing [DJ11]. Most of general principles are related with internal LO quality criteria described in LO quality model [Ser13]:

- The independence of the dynamic geometry software. The interactive book can be created with any dynamic geometry software.
- The dependence of curricula. The interactive book has to correspond to learning goals and students abilities of national mathematics educational program.
- The division of topic. Whole topic has to be broken into smaller parts depending on student's abilities which have to be trained.
- Interactivity. Have to be taken in account the interactivity type and level of digital learning tool.

The other group of principles for the interactive image are related with external LO quality criteria:

- Friendly interface for uses. The simplicity of image. The image has not be complex and there have to be as many objects (dynamic and not) as it needed to explain main geometric idea or to train student's ability.
- Systematic. There have to be system of templates. They can make visualization faster. Some of interactive images have the same structure and similar objects. It can be predict some templates for separate developed skills
- The flexibility for an image. The image can be simplified or can be made more difficult for separate student;
- The tune of text and draw. Text and image have to meet the goal of learning, regard to the principles of coherence;
- Simplicity. The interactive image cannot be overfull with additional objects (text, buttons, actions, images, and so on.) and activities

And the third group of the principles is applied for the dynamic drawing. We reviewed, concretized and generalized visualization principles which were presented by Casselman for static drawing [Cas00]. These principles let us get better results for the interactive image:

- Invisible make visible. The invisible make visible. The main goal of interactive visualization is to show things what are hidden under the words (concepts, theorems and et.). So this principle is very complicated. It joins three important ideas: 1) image or a tool have to illustrate or visualize phenomena, 2) image have to integrate algebraic and geometric components, if possible, joined with life practice 3) the image should include all information about usage of image;
- The final drawing has to convey concrete problem;
- The basic objects have to be highlighted by colours;
- The drawing have to be minimized of objects at is possible.

The final interactive book has to implement all the principles. The model of an interactive geometry visualization using the dynamic geometry is presented in Figure 2. Model has three main stages:

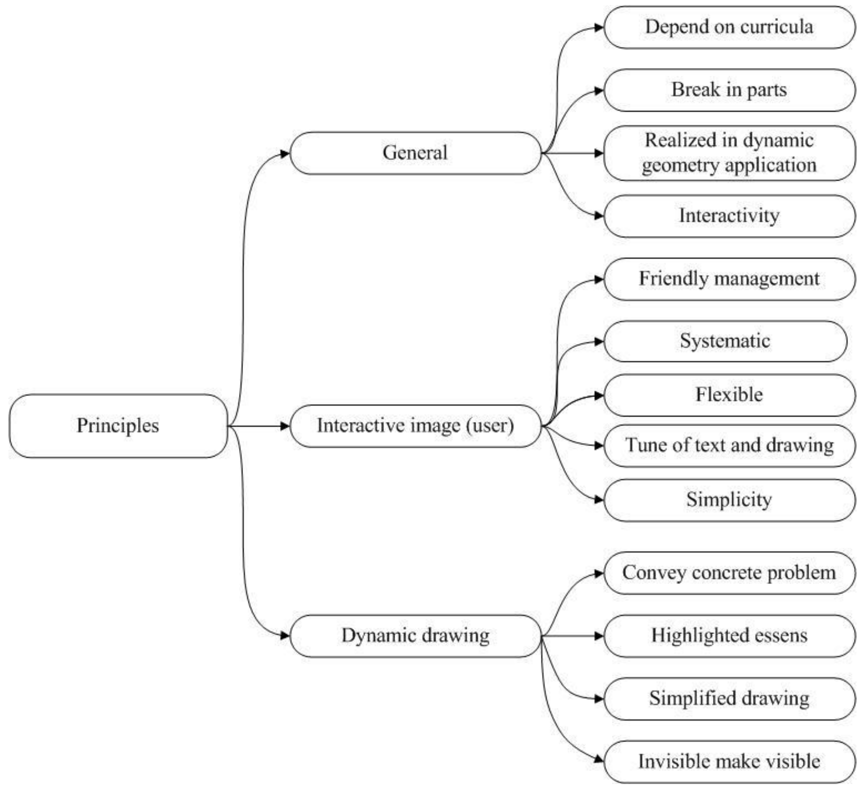


Figure 3. The principles for the interactive geometry visualization using dynamic geometry model

- Pedagogic. The templates types are set depending on learning object types and teaching methods which are discerned in the IEEE document [ISL02] and the Report on the Standardized Description of Instructional Models [DNO09]. We have set three types of scenarios: demonstration step by step, exploration and drill and practice.
- Curricula. The experts of mathematics educations choose themes what have to be visualized with dynamic geometry and models (fills) templates for technological experts. The school geometry has to be divided into small parts depending on the learnable student's abilities. It means to divide it into concepts, properties of geometric objects, axioms, theorems, proofs, methods and problems as it is required in the principles.
- The creation of the interactive image. An interactive image using dynamic geometry is designed by templates using principles of the interactive image. The scenario for every template is model in abstract data type (ADT) concept.

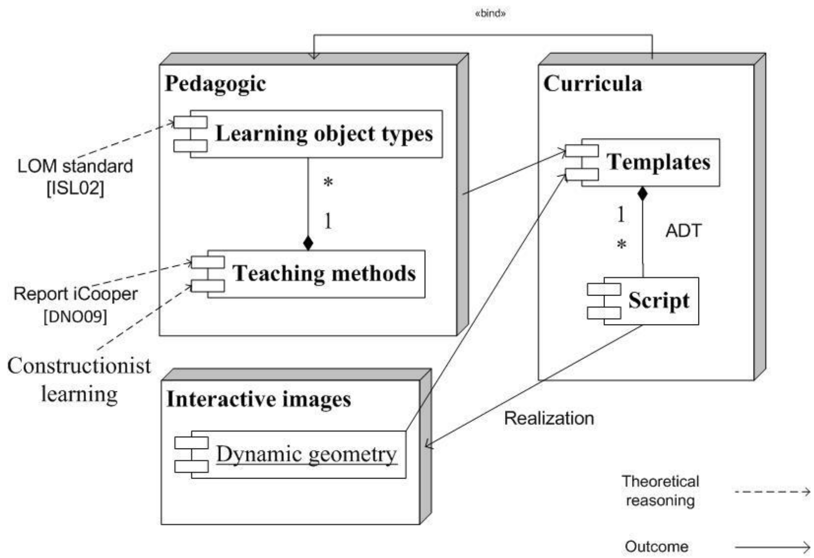


Figure 4. The model of the interactive geometry visualization using dynamic geometry

The main point in this diagram is to move template into dynamic geometry (model scenarios) and make it interactive. For this step we use concept of abstractive data type (ADT).

We have described at least four abstractive data types: geom_obj, measure, text and buttons (Figure 5).

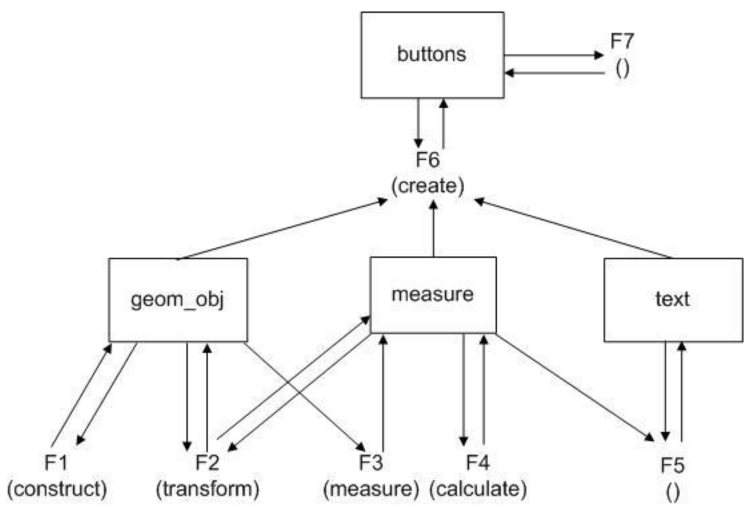


Figure 5. The relations between ADT in dynamic geometry.

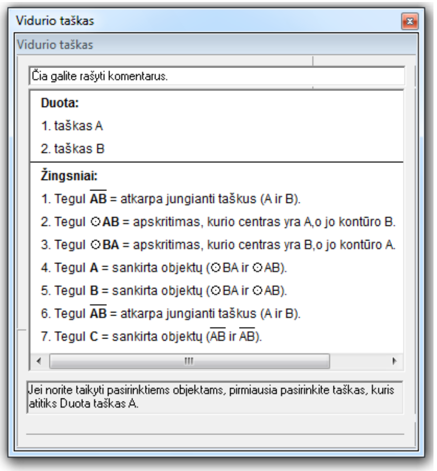
The methods of homogeneous and heterogeneous algebra are used in the description of ADT. The methods of homogeneous algebra are used for description of abstractive data types "geom._obj", "measures" and "text". The data type "buttons" requires methods of heterogeneous algebra. The sets of elements, operations and axioms are written for each data type. And in the final all data types are related using heterogeneous algebra methods and axioms of this algebra are presented too. The scenarios for dynamic geometry are written using these axioms in the realization stage.

Example of csegment midpoint construction:

1. op_taškas (nėra) = A {point}
2. op_taškas (nėra) = B {point}
3. op_atkarpa (A1, B1)= atkarpa_AB {segment}

4. op_apskritimas (A1, A1B1) = apskritimas_ AB {circle}
5. op_apskritimas (B1, A1B1) = apskritimas_ BA {circle}
6. op_sank.taškas (apskritimas_ AB1, apskritimas_ BA1) = (E1,E2)) {intersection point}
7. op_atkarpa ( ) = atkarpa_ {segment}
8. op_sank.taškas (atkarpa_ , atkarpa_AB) = E {intersection point – midpoint}

Realization in the dynamic geometry is presented in Figure 6.

a) 

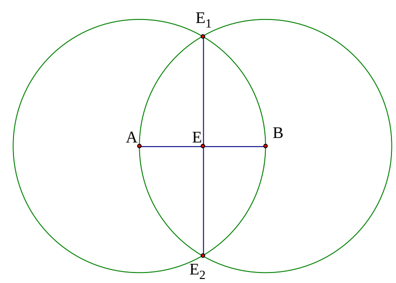
b) 

Figure 6. a) scenario in the dynamic geometry; b) graphic result of scenario

And in the final step of research the method of evaluation is chosen. We choose the method of evaluating of LO quality which was presented in 2013 [Ser13]. We can use LO quality model because our digital learning tool is complicated LO. The expert evaluation is used in this method. We have to choose criteria for digital learning tool quality. We will use questionnaires for experts. We will use Fuzzy numbers and scalarization method for digital learning tool quality evaluation.

5. A sketch of the applied research methodology in the project (data collection and analyzing methods)

1) Systematization and a comparative analysis for analytic part.
2) The construction method for model investigation.
3) The Adaptation of ADT in educational approach.
4) The methods of homogeneous and heterogeneous algebra for ADT description.
5) Approaches and methods proposed by Multiple Criteria Decision Analysis, in particular, modeling, the Goal/ Question/ Metric framework and the expert evaluation, Value measurement theory are expected to be applied in the creation of the evaluation scheme for model.

6. Application of the research work

1) Model of the interactive geometry visualization (Figure 4) can help to create qualitative and consecutive interactive imagines or environments.
2) The realization of model has to make easier teachers and students to use dynamic geometry programs.
3) The realization of model has to help students consecutive analyze math themes.
4) The realization of model has to help teachers to use constructive methods for learning and teaching.

7. How the suggested model is different, new, or better as compared to existing approaches to the problem

1) We provide the first extensive analysis of the problem field of interactive (dynamic) visualisation in geometric contests of secondary education (to our knowledge).

2) A novel idea has been suggested and investigated – to adapt ADT concept to visualize geometric contest of secondary education using dynamic geometry.

3) There is theoretical solution – method developed by adapting ADT – in Informatics.

4) Expected (it is not done yet): the adapted MO quality evaluation method for evaluation of digital learning tool.

8. Publications

[DJ04] Dagiene, V. & Jasutiene, E., (2004). Teaching and learning mathematics with Geometer‘s Sketchpad, Lithuania mathematics XIV conference. [in Lithuanian]

[DJ06] Dagiene, V. & Jasutiene, E., (2006). Developing Dynamic Sketches for Teaching Mathematics in Basic Schools, The 17th ICMI (International Commission on Mathematical Instruction) Study: Technology Revised, Hanoi University of Technology, Vietnam, (p. 120–127).

[DJ07] Dagiene, V. & Jasutiene, E. (2007). Visualization and exploring mathematics using information technologies. Information Sciences (2007) Vol. 41 ISSN 1392-0561 [in Lithuanian]

[DJJ07] Dagiene, V., Jasutiene, E. & Jevsikova, T. (2007) An Approach to Combine Learning Entities to Support Mathematics Curriculum in Schools. In: D. Benzie, M. Iding (Eds.) Informatics, Mathematics and ICT: a "golden triangle". CD proc.: Working Joint IFIP Conference. Northeastern University, Boston, Massachusetts, USA, June (p. 27–29), 2007.

[Jas07] Jasutiene, E. (2007). Constructive methods of mathematics teaching, Lithuania mathematics collection, Vilnius university, no. 47, 2007, 233 p. ISSN 0132-2818 [in Lithuanian]

[JD11] Jasute, E., Dagiene, V. (2011). A Model of Interactive Geometric Visualization with Dynamic Geometry. Information Sciences 2011 Vol. 56. ISSN 1392-0561 [in Lithuanian]

[JD12a] E. Jasutė, V. Dagienė (2012a). Towards Digital competencies in mathematics education: a model of interactive geometry. International Journal of Digital Literacy and Digital Competence, 3(2), p. 1-19.

[JD12b] E. Jasutė, V. Dagienė (2012b). Construcionist learning of geometry. Constructionism 2012: Theory, Practice and Impact, Conference proceedings. August 21-25, 2012. The education technology Lab, Athens.

9. Resources

[Cas00] Casselman, Bill. (2000). Pictures and Proofs. Notices of the AMS, 47, p. 1257–1266.

[DNO09] Derntl, Michael; Susanne Neumann, Petra Oberhuemer (2009). Report on the Standardized Description of Instructional Models. ECP 2007 EDU 417007 ICOPER, eContentplus.

[HHL09] Hohenwarter, J., Hohenwarter, M., & Lavicza, Z. (2009). Introducing Dynamic Mathematics Software to Secondary School Teachers: The Case of GeoGebra. Journal of Computers in Mathematics and Science Teaching, v28 n2 p135-146 Apr 2009

[ISL02] IEEE Standard for Learning Object Metadata. 1484.12.1-2002. ISBN: 0-7381-3297-7. http://ieeexplore.ieee.org/servlet/opac?pu

[ISO13] ISO/IEC FCD 25010: Systems and software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality models for software product quality and system quality in use. 13-Jul-09 JTC1/SC7/WG6. Version 1.62

[Ser13] Sėrikovienė S. (2013). Mokomųjų objektų daugkartinio panaudojamumo kokybės vertinimo metodų taikymo tyrimas. Technologijų moksų krupties Informatikos inžinerijos daktaro disertacija. Vilniaus universitetas Matematikos ir informatikos fakultetas.
[SK11] Stols, G. & Kriek, J. (2011). University of South AfricaWhy don't all maths teachers use dynamic geometry software in their classrooms? Australasian Journal of Educational Technology, 2011, 27(1), 137-151 number=8032

10. Questions

I am concentrate on description of my work (writing thesis) in this stage of my work and my questions are associated with formulating of some main parts of thesis.
My work arose from practical idea – to help teacher to use dynamic geometry program (so powerful tool) for teaching and learning. I expect to present phD thesis for informatics engineering commission. So I have to formulate our scientific problem in this domain. I have written some suggestions about formulating scientific problem: To equip teachers with tools for using dynamic geometry in lessons. Maybe it can be formulated in more scientific way? Maybe I can get some discussions in this way or some navigation how to get answer to this question?
However I have some scientific novelty presented in chapter 7 [page 6]. Here I am not convinced too. I would like to ask if my novelty formulation is correct and possible for informatics engineering domain.
The last stage of my work is to evaluate model. I suppose to find out evaluation criteria and method to evaluate interactive images with dynamic drawing for geometric domain. I would like to know more about Multiple Criteria Decision Analysis for evaluating my model. Maybe I can get some navigation in this way (some articles or books what I must to read)?
I will be very grateful for any note of my work.
Area of the triangle is equal to half product of two sides lengths and the sinus of the included angle.

Radovan Krajnc

2nd year MSc student
Faculty of Electrical Engineering and Computer Science
Maribor, Slovenia
radovan.krajnc@zrss.si

Research area

Title of the research

Correlation between student's success on Beaver computer competition and mentor's educational background and computer science knowledge

The main problem you are trying to tackle and why it is relevant

In Slovenian school system there are no mandatory computer science subjects. There is one subject that can be chosen by the students, and is called computer science. But the topics and goals of the subject are more connected with digital competences and information technologies. Consecutive the topics are not connected with the computer science and the teachers, who teach this subject, have diverse educational background that is not always in computer science area. There is new subject in preparation that will have computer science goals and content. Because this subject will teach the same teacher we want to determine if there are common characteristic of the teachers that results in good student's knowledge.

Because we don't have computer science subject we cannot make such a research. But we can use the data and the results of the computer science Beaver competition because the questions are based on computer science concepts and computational thinking. Teachers prepare students to the competition. We want to find out what are the factors that influence good student's results on the competition.

The aim of research

We want to determine correlation between teacher's computer science knowledge and knowledge of the students. We expect that will be possible to plan future teacher's education on the results of the research.

An outline of the current knowledge of the problem domain (What is the state-of-the-art in relation to existing solutions to the problem)

Computer science discipline at the moment does not have plenty standard validated learning assessment tools. Valid measures would enable researchers and educators to measure student learning, directly compare pedagogical approaches and investigate the success of curricular innovations.

In computer science, establishing valid ways of measuring student learning about fundamental topics, such as computational thinking and programming concepts, is a key goal of assessment research (Allison Elliott Tew, Brian Dorn, "The Case for Validated Tools in Computer Science Education Research," Computer, vol. 46, no. 9, pp. 60-66, Sept. 2013, doi:10.1109/MC.2013.259).

Advances beyond the state-of-the-art in terms of your specific contribution and research plan (A description of the Ph.D. project's contribution to the problem solution)

It is difficult to measure student learning and compare it to the pedagogical approach, when there are teachers with very different educational background. It is a question if all of them understand basic computer science concepts. We want to explore if there is a correlation with educational background (and CS knowledge) of the teachers and achievements (knowledge) of students in Beaver competition.

A presentation of any preliminary ideas, the proposed approach and achieved results

Current status of the research plan

At the moment I am collecting data that are already available. I have to get the data of all teachers of computer science in Slovenia and the names of those teachers that preparing students to the computer science Beaver competition. I am waiting on the results of the Beaver competitions. I will get data about students, their mentors, achievements on the competition. This year there are 11.000 students attending the competition.

A sketch of the applied research methodology (data collection and analyzing methods)

I will extract all the teachers that prepare students on the Beaver competition. Then I have to extract:

- the student with the best result that can be connected with the mentor
- or calculate the average of achievements of all students from one teacher - mentor.

After that I can examine the correlation between teachers and students achievement on the competition.

Expected achievements and possible evaluation metrics to establish the level of success of your results

The research will try to find an answer to the question what is the impact of the teacher's characteristics (education, experience in computer science and other factors) to the students' knowledge or achievement on the Beaver competition.

Bibliographic References

- R. Krajnc. (2011). USING WebCT FOR TRANSFORMATION OF AN EDUCATIONAL PROGRAMME IN TO AN ONLINE STUDY MODE, Detech conference
- R. Krajnc, M. Debevc. (2003). Digital TV and MHP standard, Possibilities and limitations of development interactive TV aplications with MHP standard, International Electrotechnical and computer science conference ERK '03
- R. Krajnc, (2005), Transforming of the educational experience, 10th international conference MIRK '05
- R. Krajnc, (2006), Reusable materials - Learning Objects, International conference Information Society IS 2006
- R. Krajnc, (2007), What role play roles in Moodle 1.7, Moodle Moot conference
- R. Krajnc, (2008), Learning english (and slovene) with Moodle, Moodle Moot conference, Koper
- R. Krajnc, (2011), R. Krajnc, (2011), How to keep enthusiasm for using ICT after experience with the first e-class in Slovenia, International conference Sirikt
- R. Krajnc, M. Đukić, (2012), School collaboration in planning lessons, International conference Sirikt

- R. Krajnc, M. Vičič Krabonja, (2011), Milestone projects on the way towards modern school, International Conference InfoKomTeh 2011
- L. Novak, R. Krajnc, E. Velikanje, (2013), Do we know what the pupils can do in the first nurturing-educational period of primary school in the field of computer science?, International conference Sirikt

Expectations and motivation to attend doctoral consortium

Expect to prepare questionnaires and all research instruments and methodology. I want to work on my hypothesis. I want to be sure that at the end of the research I will get usable results that can be used in my master's thesis.
Expect to share experience and knowledge with attendants of the consortium and to get practical advices from the invited speakers.

Nataša Kristan

1st year PhD student
University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25
1000 Ljubljana, Slovenia
natasa.kristan@fri.uni-lj.si

1. **Research area: Concepts of Computer Science in Education**
   Title of the research: Motivation for learning of Programming and Theoretical Computer Science using Personalization

   What and why: In Computer Science (CS) education there is strong emphasis on ICT competences, which are meant in the broad sense (including programming). If we want to achieve ICT competences, it is necessary to set appropriate learning objectives and implement it in the curriculum. Motivation is crucial for achieving the basic concepts of CS.
   One way to motivate students to learn those concepts, is the personalization of the learning tool, for instance in the fields of programming or theoretical CS.

I am interested in motivation to learn concepts of CS (programming and theoretical CS) in relation to learning style, personality type and temperament. There are few questions, which need to be answered:

- Is student success dependent on learning style or just on motivation?
- Can the content of learning be generalized for all students independently of learning- and personality- types?
- Are students really not mature enough to think in abstract way or does this depend on teacher/material?
- Can there be defined a generalized interface between the tutor based learning environment and the tutoring system that would provide guidance for the pupil in the learning environment by the tutoring system according to his/her learning style/personality type?

2. **A presentation of any preliminary ideas, the proposed approach and achieved results**

I am in the beginning of my research, but as a first step I have already done a review of Computer Science course in our schools (gymnasium).

We made a research about what should be taught in secondary schools at each level. Therefore we made a comparison of learning objectives and curricula of Computer Science among CSTA K-12 Computer Science Standards, English Royal Society Report and Bavarian Standards (*Grundsätze und Standards für die Informatik in der Schule*) with the current Slovenian curriculum. We found out that learning objectives are not significantly different from the Slovenian perspective, since the Slovenian curriculum is quite open, although some contents are not covered in Slovenian secondary schools. The major difference found is in the topics of algorithms and programming. They are mandatory (or optional) in United States, England and Bavaria, while in Slovenia these topics are optional and much more limited.

As one of the results of our research is the Table 1, that presents at which grade the Computer Science Course is mandatory or optional in different country.

Table 2: Comparison of school systems about the subject Computer Science

<table>
<tr><th>Years old</th><th>Grade</th><th>USA</th><th>England</th><th>Bavaria (Germany)</th><th>Slovenia</th></tr>
<tr><td>5-6</td><td>Kindergarten</td><td rowspan="6">Level 1 MANDATORY</td><td rowspan="2">Key Stage 1 MANDATORY</td><td rowspan="6">OPTIONAL</td><td rowspan="6">ICT at other courses</td></tr>
<tr><td>6-7</td><td>1. grade</td></tr>
<tr><td>7-8</td><td>2. grade</td><td rowspan="4">Key Stage 2 MANDATORY</td></tr>
<tr><td>8-9</td><td>3. grade</td></tr>
<tr><td>9-10</td><td>4. grade</td></tr>
<tr><td>10-11</td><td>5. grade</td></tr>
<tr><td>11-12</td><td>6. grade</td><td rowspan="3">Level 2 OPTIONAL</td><td rowspan="3">Key Stage 3 MANDATORY</td><td rowspan="3">MANDATORY</td><td rowspan="3">OPTIONAL</td></tr>
<tr><td>12-13</td><td>7. grade</td></tr>
<tr><td>13-14</td><td>8. grade</td></tr>
<tr><td>14-15</td><td>9. grade</td><td>Level 3 MANDATORY</td><td rowspan="3">Key Stage 4 MANDATORY</td><td rowspan="4">MANDATORY</td><td>MANDATORY</td></tr>
<tr><td>15-16</td><td>10. grade</td><td rowspan="3">OPTIONAL</td><td rowspan="3">OPTIONAL</td></tr>
<tr><td>16-17</td><td>11. grade</td></tr>
<tr><td>17-18</td><td>12. grade</td><td>Key Stage 5 OPTIONAL</td></tr>
</table>

Further, the Table 2 compares different topics of Computer Science and whether they are mandatory, optional or missing in different curricula. In the USA level 3 (grades 9-12) is divided into three discrete courses:

- Level 3A: recommended for grades 9 or 10 and is mandatory,
- Level 3B: recommended for grades 10 or 11 and is optional,
- Level 3C: recommended for grades 11 or 12 and is elective.

Table 3: The main few differences in learning objectives (M = Mandatory)

Learning objective	**USA**	**England**	**Bavaria**	**Slovenia**
Describe how various types of data are stored in a computer system.	3A	M	M	/
Describe the concept of parallel processing as a strategy to solve large problems.	3A	M	M	/
Demonstrate concurrency by separating processes into threads and dividing data into parallel streams.	3B	M	M	/
Critically examine classical algorithms and implement an original algorithm.	3B	M	M	/
Classify problems as tractable, intractable, or computationally unsolvable.	3B	M	M	/
Use mobile devices/emulators to design, develop, and implement mobile computing applications.	3A	M	M	/

3. Bibliographic References

1. Kristan N (2011). 3D visualization in education, Diploma thesis. University of Ljubljana, Faculty of Education.
2. Pikl B., Kristan N., Ham Ž. And Brodnik A. (2012). Developing integrated ACM competitions website. International Electrotechnical and Computer Science Conference, sect. Didactics.
3. Kristan N. and Brodnik A. (2013). Comparison of Learning Objectives and Curricula of Computer Science. *Proceedings of the 16th International Multiconference Information Society*, volume A, 520-521.
4. Kristan N. and Brodnik A. (2013). Integrated ACM competitions website. World Conference on Computers in Education.
5. Kristan N., Mede G. and Brodnik A. (2013). Motivation and Constructive Learning in Secondary Technical School. World Conference on Computers in Education.
6. Vesel J., Kristan N. and Brodnik A. (2013). There and back again - A tale between photographs and bits. World Conference on Computers in Education.

4. Expectations and motivation to attend doctoral consortium

Main goal is to discuss about my topic and other topics, get some new idea about the research, get familiar with other similar topics and distinguish between them. Because I am just in the beginning it would be really helpful to get feedback about my research area and topic, so I can focus more precise on a specific part.

Michala Křížová

Second year of doctoral studies
Faculty of Education, University of South Bohemia
Jeronýmova 8
České Budějovice, Czech Republic, 371 15
krizovam@pf.jcu.cz, krizova.michala@email.cz

1. *Research area*

The quality of students´ presentations and the aspects which students lay emphasis on during creation of presentation

The main problem I am trying to tackle is presentation. In last few years I have been watching a lot of students who were presenting different topics. I worked as a teacher at grammar school and now I work at high school of information technology and law studies. In my practice, I find out that students have often problems with creating correct and interesting presentations. There are many areas where students have problems. I want to mention some of them. Students write too much information into their presentations. Their presentations don´t have any logical structure. They don´t speak about the topic, but only read notes written in the presentation and also have problems with speaking in front of people.

On the other hand I have been to a competition "Prezentiada" three times. It is about creating presentations and presenting them in groups of three students. Sometimes I was very surprised by their performances.

Because of this observation I decided to study this topic deeply. I have two ways to examine. First I want to study the presentations from the perspective of students and the aspects which affect students during creating the presentations. Now I am more thinking over how teachers instruct students during creating the presentations. One of the most interesting questions for me is how teachers evaluate students' presentations. Because we don´t have concrete rules how to do a really good presentation. There is a lot of literature about „How to make a good presentation" (Duarte N., 2012; Bradbury A., 2001), but not many about "How to teach it". I these books, you can choose one of many designs to create a presentation, for example 5x5 (5 points on 5 words), Biker "B" (6 parts of a presentation: Bang!, Introduction, Key points, Examples, Recap, Bang!), invent a story, etc. There are many recommendations, but they can´t guarantee you, that your presentation will be successful. So how to teach something that we don´t know how it really works?

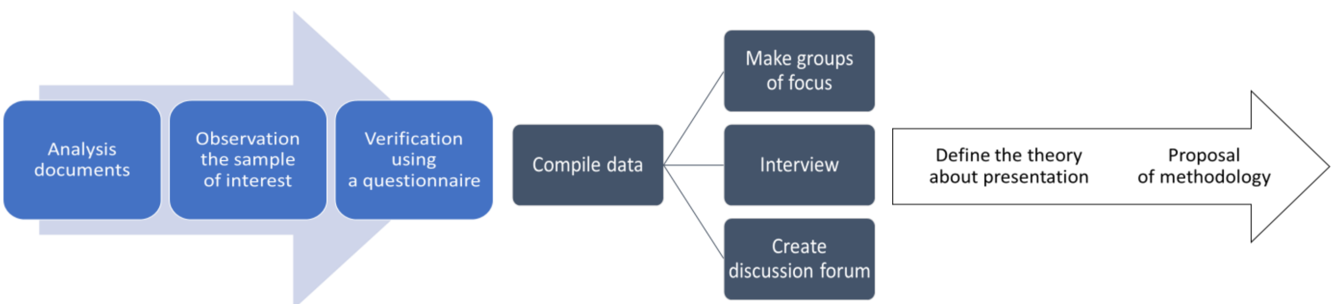
We can also find some researches about presentations at school. The papers speak about how presentations influence students. A lot of teachers use presentations for teaching and students have an enthusiastic interest in technology in education. The presentation is more interesting than a blackboard or a whiteboard. (Craig R. J., Amernic J.H., 2006; Rain T., 2003; Panjwan S., 2010) Teachers can also make very interesting presentations. But can they impart these skills to students?

The main tasks are: How the factors affect students? How to teach students to make better presentations and better speeches with regard to these factors? What is the influence of teachers on the process of making students' presentations?

The contribution of this research is in new information about presentations and the process of creating them. It can contribute to creating a new methodology for teachers, to the development of the theory of information technology or to the development of teaching future teachers of IT.

2. A presentation of any preliminary ideas, the proposed approach and achieved results

I chose the qualitative approach.
This time the research is in the first phase. The research begins with analyzing documents about presentations and an education presentation, which means student´s books and teacher's books of informatics.
The next step is to determinate the main factors, problematic parts and the relationship between teaching presentation and final students' presentations from the perspective of students. This is not the main part, but it is necessary to determine the main terms for this research. I still haven´t found any good literature, which speaks about these factors.
We prepare some groups of factors – external and inner factors.
External: software, theme, teacher's sample, teaching approach, etc.
Inner: motivation, PC skills, student's personality, previous experience, etc.
These days we prepare questionnaire for students at secondary and high school. Questions focus on the important points of student's presentation presentations.
For example:
What do you focus on during creation presentation? (Graphic aspects, resume, logic structure…)
Do you choose different software for every presentation? (Yes/No)
Which kind of presentation software do you choose for school report? (MS PowerPoint, Prezi, Google Docs…)



1. Figure: Design of research

In the second phase I want to compile data from questionnaire and my initial research and create questions for interview and online discussion forum. I want to check identified factors in groups of focus. I am going to ask the first group of teachers to make interview. I want to closer collaborate with these teachers and study their teaching and also study presentations of their students.
The last phase is about defining the theory with the findings. This part is maybe too ambitious, so it is only possible last part of my research

3. *Bibliographic References*

7. DUARTE, N.: Slajdologie: tajemství tvorby skvělých prezentací. 1. vyd. Brno: BizBooks, 2012, xvii, 270 s. ISBN 978-80-265-0009-4.
8. BRADBURY, A.: Jak úspěšně prezentovat a přesvědčit. Vyd. 1. Praha: Computer Press, 2001, xii, 129 s. ISBN 80-722-6424-9.
9. CRAIG R. J., AMERIC J. H.: PowerPoint Presentation Technology and the Dynamics of Teaching. Innovative Higher Education. 2006, Oct 2006
10. RAIN, T.: Využití prezentačního softwaru v pedagogické praxi. INFORUM [online]. 2003 [cit. 2013-05-28]. Dostupné z: http://www.inforum.cz/archiv/inforum2003/prispevky/Rain_Tomas.pdf
11. PANJWAN, S. C.: A Presentation Tool for School Teachers. In: Microsoft Research [online]. 2010 [cit. 2013-06-18]. Dostupné z: http://research.microsoft.com/pubs/135480/Collage-final-v2.pdf
12. JOHNSON K.; SHARP V.: Is PowerPoint Crippling Our Students? Learning and Leading with Technology. 2005. no. Nov. 2005, s. 6-7
13. FEDISSON, M.: PowerPoint Presentations Increase Achievement and Student Attitudes Towards Technology. International Journal of Information & Communication Technology Education. 2007, roč. 3, č. 4, s. 64-75.
14. TOWNSEND, J. S.: Prezentace. Vyd. 1. Ilustrace Phil Hailstone. Praha: Portál, 2007, 109 s. Management do kapsy, 11. ISBN 978-807-3672-140.
15. SHEPHARD, K. KNIGHTBRIDGE.: Exploring presentation styles in higher education teaching and research situations: distance and face-to-face. *Open Learning*. 2011, roč. 26, č. 3, s. 223-236. DOI: 10.1080/02680513.2011.611684.
16. BASTURK, R.: Applying the many-facet Rasch model to evaluate PowerPoint presentation performance in higher education. *Assessment & Evaluation in Higher Education*. 2008, roč. 33, č. 4, s. 431-444. DOI: 10.1080/02602930701562775.
17. The problem with PowerPoint: 25 years of PowerPoint. BBC. BBC NEWS: UK Magazine [online]. [cit. 2013-06-18]. Dostupné z: http://news.bbc.co.uk/2/hi/8207849.stm
18. THE INSTRUCTIONAL VALUE OF POWERPOINT PRESENTATIONS IN TEACHER EDUCATION. E-Journal of New World Sciences Academy (NWSA). 2011, roč. 6, č. 1, s. 918-924.

4. *Expectations and motivation to attend doctoral consortium*

For me is it an opportunity to discuss about my research and a topic of research.

Mareen Przybylla

Institute of Computer Science, University of Potsdam, Germany
przybyll@cs.uni-potsdam.de
Advisor: Prof. Dr. Andreas Schwill

Physical Computing in Computer Science Education

1. Introduction

Nowadays computer science products pervade students' everyday lives. There are sensors and actuators everywhere around us, embedded in computing systems that ease and enrich our lives in many ways. On the other hand, for German students we know that computer science is perceived as a subject full of abstract and unrealistic contents. [Kn11] At the same time, computer science as a school subject is invaluable for students to gain useful competences that help them to succeed in our fast-paced, creative society. In order to counteract this and other problems and to motivate more students to take computer science classes, the teaching concept "My Interactive Garden" (MyIG)1 was developed in a master's thesis and put into practice recently. It is a concept for realizing collaborative exhibitions of interactive objects and installations and encourages creative and constructionist learning. In addition to teaching proposals, it includes a physical computing kit based on Arduino that overcomes the barrier of technical complexity and encourages immediate tinkering. Physical computing with MyIG can be approached in different levels, which makes it possible to stay in the same context while advancing skills and competences. It also allows cross-level projects and thus facilitates collaboration between students without prior experience and students who bring great knowledge. [PR12]

2. Frame of research

Physical computing has received a lot of attention throughout the last years, especially by noncomputerscientists. Among makers (hobbyists, artists, designers) who use Arduino, the most popularplatform for physical computing at the time, physical computing is considered as an activity that involves „...prototyping with electronics, turning sensors, actuators and microcontrollers into materials for designers and artists. It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behavior implemented as software running inside a microcontroller." [Ba11] The term was initially used by [OSI04], who wanted to strengthen the role of the physical body in computing: their focus is on the question „What does the person physically do?". Others, often in educational settings, have adapted the term and use it in a wider meaning: they see physical computing as connecting computers to the physical world. (e.g. [UL13], [CL13], [LMS13]) Taking into account the aspects of creativity from which education can benefit, it becomes clear where the differences to embedded computing or robotics are. Physical computing is happening when students use programmable hardware to creatively design and craft interactive objects. "Interactive Objects […] perceive their environment with sensors, which in turn deliver data to be processed by the microcontroller. According to the configuration of the systems these data are processed and passed on to the actuators. In this way, interactive objects communicate with their environment. They are created with crafts, art and design material. They fulfill a particular purpose, which may be purely artistic." [PR13] In English-speaking contexts, more and more scientific publications devote themselves to the topic. Paulo Blikstein for instance has investigated physical computing kits for their underlying design principles. [Bl13]

3. Problems in the field of research

Physical computing is popular in many extracurricular computer science education contexts, such as afternoon clubs or summer camps. However, it only plays a little role in classroom settings. There are some approaches to teaching that are similar to physical computing, but they are often – and especially in Germany – subordinated to the subject of robotics or embedded systems, even though those terms cannot compete with the creative potentials offered by physical computing. (e.g. [St11], [RRB08], [Bau11], [Bau12], [PA13]). Despite its potentials for constructionist learning environments, physical computing in education is mainly used by non-computer scientists, e.g. in arts, physics or biology classes. 1 See www.informatikdidaktik.de/MyIG, accessed: October 25, 2013

4. Research objectives/goals.

The aim of this work is to analyze the phenomenon of physical computing in order to situate possible new contents within the field of computer science education. In the dissertation both will be investigated, existing approaches and topic areas suitable for new approaches towards physical computing education within computer science education. It will be shown that physical computing perfectly matches the core principles of the constructionist learning theory described in [PH91]. It is intended to analyze what benefits physical computing can bring to the classroom by investigating the effects on students' motivation, creativity, learning success, growth in competences and their understanding of computer science and computing systems. The research will therefore be guided by the following research questions.

5. Research questions

1. How is *physical computing* to be understood in terms of computer science education and classroom teaching?
2. Which new *competences* and *areas of computer science* are accessible and which become irrelevant?
3. Are there *changes in value* (benefits or challenges) to teaching approaches with physical computing as opposed to teaching approaches where non-physical systems are designed?
4. What is the *current status of physical computing* in computer science education, and what *contribution* towards computer science education can we expect in the future?
5. Which *effects* does physical computing in computer science education have on students concerning success in learning, growth in competences, motivation, being creative, perception of computer science and computing systems?

6. Applied research

MyIG includes a constructionist learning environment, which allows students to craft, design, program and build their own interactive objects. This approach was piloted in an afternoon club and gave a first impression of students' perception of embedded computing systems, their acceptance of "informatics pottery making"2, balance between informatics and crafting activities and the added value of physical computing. Data was collected with questionnaires and by observing the students.

7. Results

There were first tendencies observed in this pilot project, suggesting that physical computing may help students in expanding their understanding of computing systems. The students liked the pottery making approach and the amount of crafting influenced the amount of programming: the more complex the students’ interactive objects became, the more complex were their programs. [PR13] These data are not statistically valid, since only a very small number of students were involved in the project. It was a first trial to test the approach and figure out difficulties. To validate these findings in a larger group of students and to explore further assumptions and hypotheses of this dissertation project, an empirical study will be conducted. In advance a preliminary study will be performed in which a series of lessons will be held and analyzed under specific criteria and with methods, which are still to be developed.

8. Contributions

Until now, physical computing is regarded as an interesting and exciting phenomenon by many teachers, but for most of them it has not been suitable for classroom use due to the technical complexity of breadboard and soldering activities. With MyIG teachers are provided with ideas and solutions to this problem. A detailed analysis to extract and define topics relevant for physical computing and to define fields of competencies that can be gained with physical computing is yet to come and will be the contribution of this dissertation to informatics didactics.

9. References

[Ba11] Banzi M. (2011): *Getting Started with Arduino*. 2nd Edition. Aufl. Sebastopol, CA: O'Reilly Media / Make. The approach „informatic pottery making“ contains the idea that in the near future students will bring home from computer science classes digital, interactive artifacts similar to hand made vases made in pottery class that can be investigated, shown around and admired by others in a constructionist sense. This way, computer science students develop a personal relationship to the subject. (See [PR13])

[Bau11] Baumann R. (2011): Eingebettete Systeme verstehen. Teil 1: Kreatives Experimentieren mit Arduino. *LOG IN*, 171, 33-45.

[Bau12] Baumann R. (2012/13): Eingebettete Systeme verstehen. Teil 2: Arduino zwischen analoger und digitaler Welt. *LOG IN*, 174, 37-48.

[Bl13] Blikstein P. (2013): Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In: Hourcade J.P.; Miller E.A. und Egeland A. (Hg.):*Proceedings of IDC 2013. The 12th International Conference on Interaction Design and Children*. New York City: ACM. S. 173-182.

[CL13] Computer Laboratory, University of Cambridge (2013): *Physical computing with Raspberry Pi*. http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/, accessed October 24, 2013.

[Kn11] Knobelsdorf M. (2011): *Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung*. (Dissertation) Berlin.

[LMS13] Libow Martinez S. und Stager G. (2013): *Invent to Learn - Making, Tinkering, and Engineering in the Classroom*. Torrance: Constructing Modern Knowledge Press.

[OSI04] O'Sullivan D. und Igoe T. (2004): *Physical Computing: Sensing and Controlling the Physical World with Computers*. Boston: Thomson Course Technology PTR.

[PA13] Pelz L. und Arnhold W. (2013): *Die Waschmaschine - Embedded computing im Alltag*. Berlin.

[PH91] Papert S. und Harel I. (1991): Situating Constructionism. In Papert S. und Harel I. (Hg.), *Constructionism*. Norwood: Ablex Publishing Corporation. available at: http://web.media.mit.edu/~calla/web_comunidad/Reading-En/situating_constructionism.pdf.

[St11] Strecker K. (2011): Wie viel Programmierkompetenz braucht der Mensch? *LOG IN*, 169/170, 40-47.

[PR12] Przybylla M. und Romeike R. (2012): My Interactive Garden - A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In: Kynigos C.; Clayson E. und Yiannoutsou N. (Hg.): *Constructionism: Theory, Practice and Impact. Proceedings of Constructionism 2012*. Athens. S. 395-404.
[PR13] Przybylla M. und Romeike R. (2013): Physical Computing im Informatikunterricht. In: *Akzeptierter Beitrag zur GI Fachtagung Informatik und Schule (INFOS) 2013*. Kiel: Gesellschaft für Informatik.
[RRB08] Rusk N.; Resnick M.; Berg R. et al. (2008): New Pathways into Robotics: Strategies for Broadening Participation. *Journal of Science Education and Technology*, 17, 1, 59-69.
[UL13] Universität Leiden: *Hardware & Physical Computing*. http://mediatechnology.leiden.edu/programme/curriculum/hardware-physical-computing, accesed July 29, 2013.

Lennart Rolandsson

Ph.D studies 2008-2011, 2012-2015
KTH, Royal Institute of Technology
Osquars backe 25
100 44 Stockholm, Sweden
lennartr@kth.se

1. Research area

Computing education in translation (in time and space)

As a doctoral student I will write a thesis consisting of 4 articles and a cover, which describes how these four articles all fit together. The following text unfolds my intentions going from present work (teachers' experience of object of learning), based on former work (development of curriculum), to motivate future work. I passed the level of licentiate degree in December 2012, with the thesis "**Changing Computer Programming Education; The Dinosaur that Survived in School**: An explorative study of educational issues based on teachers' beliefs and curriculum development in secondary school" which is summarized later under former work. The thesis is a platform for my present work, which concern a research project together with four other researchers[1] investigating "Theory and practice in lab work – a complex interplay".

My present work

The research project is empirically based, with the aim to investigate the interplay between learning of theory and learning of practice in computer programming through laboratory work. The aim for the project is to gain insights about what is required in order that students' learning of practice and learning of theory during lab sessions can support each other mutually. Previous research has shown that learning in the laboratory is problematic, in that neither the learning of theory nor the learning of practice during lab work is satisfactory (e.g., Eckerdal, 2009; Hofstein & Lunetta, 2003; McCormick, 1997; Séré, 2002; von Aufschnaiter and von Aufschnaiter, 2007).

The overarching research question for our project is: How can we understand, describe and develop learning and teaching in lab work, with particular focus on the relation between students' learning of theory and their learning of practice? A research question that we investigate by different levels of abstraction:

- Q1: On a **theoretical level** and with the objective to better understand learning during lab work.
- Q2: On an **empirical level** and with the objective to understand students' learning in specific lab situations.
- Q3: On the **level of contribution** to educational practice, with the objective to enhance lab-based teaching and learning.

Phenomenography and variation theory are used to discuss the space of learning that is opened in a learning situation. In search for the space we focus on a specific object of learning in

[1] Together with Anders Berglund, Anna Eckerdal, Michael Thuné (all from Uppsala, Sweden) and my supervisor Inga-Britt Skogh (KTH, Sweden).

programming (function, parameters and return value), in relation to three different aspects: the intended object of learning, the enacted object of learning, and the lived object of learning.[2]

My research concerns the teachers' perspective, wherefore I focus on the intended object of learning. So far in the project we have found that the OoL is dynamic by nature, and teachers are not in control of the learning processes as the intended and lived OoL are not very well aligned. Teachers are vague in their intentions. As for Marton and Booth (1997) my interest concerns the **concept of learning or the nature of learning.** However I focus on teachers' perspective as I did in my former work (see below).

Lately programming knowledge has received high impetus, see for instance UK[3] and Estonia, as computer science in education, is considered important **for all ages in school**, which according to my research (licentiate thesis) was already the case in Sweden, during the 1980s, when every pupil should have some knowledge (to become literate) about computers and programming. This is according to me a problematic initiative as the instructional setting is not aligned with the diversity of different student experiences of technology in Computer Science/Informatics. As it was the during the 1980s and 1990s discovery learning was expected to happen (Sloane & Linn, 1988; Linn & Clancy, 1992), as it still is among today's programming teachers (Rolandsson, 2012), which is not satisfactory as many pupils will have a hard time understanding the content.

Today, though, education can offer other environments and non-machine experiences for learning programming and the principals behind that technology[4] and there are innovations for embodiment of programming knowledge.[5] The setting could therefore be expected to offer new possibilities as educational technology has evolved, BUT still teachers are faced with the situation of transforming computer science concepts into something applicable for learning (Haberman, 2006).

> "We only need to get more experience with teaching these topics understandably in our schools" (Hromkovič & Steffen, 2011: 24)

The statement in the quotation is common as many teachers, scholars and politicians believe it is true, which I intend to problematize as there is more than teaching experience, to take into account if you would like to teach the principals and ideas implicit in Informatics (see for instance "computational thinking") , instead of adapting to what ICT offers, which Hromkovič and Steffen agree upon

> "Teaching of informatics enrich education in ways … [that brings forward a] discussion with politicians about integrating proper informatics and not only ICT skills in the educational systems, but it can also help us as teachers to focus on the fundamentals and on sustainable knowledge." (ibid: 21)

I therefore would like to discuss on the 4th DC in Lithuania what education (e.g. educational institutions, government, board of education and teacher training) offer in this domain and what

---

[2] The intended object of learning is the object of learning as seen from the teacher's perspective and the lived object of learning is the object of learning as seen from the student's point of view, i.e., the actual outcome of learning. Finally, the enacted object of learning, briefly explained, is a researcher's description, from a particular theoretical perspective, of what the possibilities are for seeing a certain object of learning in a certain learning situation.

[3] See "National curriculum in England: computing programmes of study". Retrieved 2013-11-20.

[4] e.g. Kodu, Alice and Scratch.

[5] e.g. CSUnplugged where you experience for instance a sorting algorithm. Other examples is when you do a break dance with Yenka-software, or give instructions to a peer or a robot, to achieve a specific movement or dance.

teachers perceive as important in the curriculum implementation, and what incitements that matters. Specifically I would like to discuss research approaches for collection of data (teachers' perspective) and what sorts of research outcome to expect in different approaches. The approach sketched bellow is based on Designed Based Research (DBR), which brings the research outcome closer to practice, as it becomes useful for teachers participants , when reflecting on different aspects of learning.

2. **A presentation of any preliminary ideas, the proposed approach and achieved results**

My former work (Text from abstract in licentiate thesis)
"With the intention to contribute to research in computer programming education the thesis depicts the mind-set of teachers and their beliefs in relation to the early enactment of the informatics curriculum in Swedish upper secondary school. Two perspectives are covered in the thesis. Based on original documents and interviews with curriculum developers, the enactment of the informatics/programming curriculum during the 1970s and 1980s is explored (Paper 1). This historical perspective is supplemented with a perspective from the present day where current teaching practice is explored through teachers' statements (seminars with associated questionnaires) regarding their beliefs about teaching and learning programming (Paper 2).
The historical data reveals that experimental work within the informatics curriculum was initiated in the mid-1970s. In the early stages of the curriculum development process a contemporary post gymnasium programme in computing was used as a blueprint. The curriculum relied on programming as well as system development, wherefore a question of importance was raised early in the process; should the subject matter of informatics, be taught by 'regular' Natural Sciences and Mathematics teachers or by contemporary vocational education teachers in ADP? The question was initially solved using stereotypical examples of how to apply system development, which was later suggested as a replacement for programming activities. The initial incitement to offer informatics education during the 1970s was discovered in the recruitment of a broader group of students within the Natural Science Programme and the perception that it would contribute to the development of students' ability to think logically and problem solving skills.
The thesis unravels an instructional dependence among today's teachers where students' logical and analytical abilities (even before the courses start) are considered crucial to students' learning, while teachers question the importance of their pedagogy. Teachers in the study commonly express the belief that their instructions hardly matter to the students' learning. Instead these teachers perceive learning programming as an individual act. The inquiry also discover two types of instruction; a large group putting emphasis on the syntax of programming languages, and a smaller group putting emphasis on the students' experiences of learning concepts of computer science (not necessarily to do with syntax), which corresponds with the existence of two groups of teachers during the 1980s; the *partisans* who perceived learning as based on repeating sequences in a behaviouristic manner, and *defenders* who perceived learning as based on discovery and self-teaching.

In summary the inquiry depicts an instructional tradition based on teachers' beliefs where the historical development of the subject sets the framework for the teaching. Directly and indirectly the historical development and related traditions govern what programming teachers in upper secondary school will/are able to present to their students."

My future work (a sketch)

With the Bebras contest, www.bebras.org, (e.g. Dagiene & Futschek, 2008), teachers with interest in didactics of Informatics/Computer science will be offered in-service training; managed by one or two universities in each country. With that experience from different countries, it would be manageable to find at least five teachers in five different countries, in total 25 teachers. A cohort of teachers would be involved in a research project for three to five years.

For research, it is crucial to build trust between teachers and researchers, wherefore a teacher cohort would be offered in-service training. The data collection should be done qualitative (interviews) and quantitative to pinpoint differences in teachers' practice towards technology and Informatics. It would be beneficial if the cohort holds a spread, where teachers instructional design differ as much as possible.

With DBR it would be feasible to measure change in teachers' perception about the subject as well as their practice. Commonly, in interaction with teachers, learning studies are used to manage the interaction between teachers and researchers. However, it does not necessarily have to be that way, as the proposed research is looking for teachers' perception and value implicit in computing education. Such a work could draw from other frameworks suggested by Cartelli et al. (2010) for a better understanding of what to teach and how to teach Informatics together with computer technology.

3. Bibliographic References

1. Cartelli A., Dagiene V., Futschek G. (2010). Bebras Contest and Digital Competence Assessment: Analysis of Frameworks. International Journal of Digital Literacy and Digital Competence, 1(1), 24-39.
2. Dagienė V. and Futschek G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In R.T. Mittermeir and M.M. Sysło (Eds.): ISSEP 2008, LNCS 5090, pp. 19–30.
3. Eckerdal, A. (2009). Novice Programming Students' Learning of Concepts and Practice. Uppsala,Sweden: Acta Universitatis Upsaliensis
4. Haberman, B. (2006). Teaching computing in secondary schools in a dynamic world: Challenges and directions, Vol. LNCS: 4226 of Informatics Education - The Bridge between Using and Understanding Computers International Conference on Informatics in Secondary Schools – Evolution and Perspectives, ISSEP2006, pp. 94–103: Springer.
5. Hofstein, A., & Lunetta, V. N. (2003). The Laboratory in Science Education: Foundations for the Twenty-First Century. Science Education, 88(1), 28 - 54.
6. Hromkovič, J. and Steffen, B. (2011). Why Teaching Informatics in Schools Is as Important as Teaching Mathematics and Natural Sciences. In I. Kalaš and R.T. Mittermeir (Eds.): ISSEP 2011, LNCS 7013, pp. 21–30. Berlin Heidelberg: Springer-Verlag.
7. Linn, M. C. and Clancy, M. J. (1992). The case for case studies of programming problems," Communications of the ACM, Vol. 35, No. 3, pp. 121–132.
8. Marton, F. and Booth, S. (1997). *Learning and awareness*. Mahwah, N.J.: Erlbaum.
9. McCormick, R. (1997). Conceptual and Procedural Knowledge. International Journal of Technology and Design Education, 7(1 - 2), 141 - 159.
10. Séré, M.-G. (2002). Towards Renewed Research Questions from the Outcomes of the European Project Labwork in Science Education. Science Education, 86(5), 624 - 644.

11. Sloane, K. D. and Linn, C. (1988). Instructional Conditions in Pascal Programming Classes, pp. 207–235, Teaching and learning computer programming: multiple research perspectives: Lawrence Erlbaum Associates, Inc.
12. von Aufschnaiter, C. and von Aufschnaiter, S. (2007) University students' activities, thinking and learning during laboratory work. European journal of Physics, 28(3):51-60.

4. Expectations and motivation to attend doctoral consortium

My concern is about the societal as well as the genetic epistemology in Informatics/Computer Science education. Based on other doctoral students' and senior researchers' experiences, coming from other countries, I believe the setting during the 4th DC in Lithuania could imply a beneficial occasion to discuss the implications of my present work in relation to the sketch above. It would be of huge interest to understand how different doctoral students and researchers in the domain of Informatics and Computer Science perceive the research and how well it could be applied in their own home country. What constraints could they imagine?

Jūratė Urbonienė

4th years PhD student
Vilnius University Institute of Mathematics and Informatics
Research topic: Design of adaptive algorithmization tools
The aim of research: By examining problems of programming training, student learning opportunities and needs, to develop the model of adaptive tool for programming learning.

1. Research Questions

1. How to individualize programming training according to the learners characteristics, learning styles and learning experience?
2. How to find a way to assess the learner's learning style?
3. How to design a system that takes into account learner's learning style?
4. How should look like adaptive functioning programming learning system?
5. How to assess a student learning outcomes using the system?

2. The significant problems in the field of research

1. Systems for programming teaching do not estimate learners learning styles, abilities and learning progress, and therefore it does not work adaptively throughout the learning process.
2. There are no methods to elect learning objects according student's learning styles.

3. Context and background.

Programming is a challenging cognitive process. For programming learning first of all it is necessary to learn syntax and semantics of a programming language. The beginners need this in order to start solving tasks. What is more important than to learn syntax structures is to be able to apply them in solving a real problem. It is also important to master the techniques and methods of a programming language. While learning the programming techniques, the language is only a means for expression and application of common programming concepts. Learning a programming language also promotes student's thinking skills (Mayer et al. 1986). Actually, the case of false conception of programming learning is quite frequent, as it is believed that it means learning to put down the task solution in a form of a program text by using structures of a programming language. However, program writing is just one of the programming skills. The ability to read and understand the program text is equally important. For a programmer spends a considerable time examining the patterns, i.e. programs written by others (e.g. Mannila 2007), and adapting them to his task solving. One might think that while learning to write programs you automatically learn to read them, and to keep track of the program implementation. However, studies have shown that the ability to write a program and the ability to read it has a low correlation (Winslow, 1996). Therefore, during programming learning it should be always kept in mind the importance of developing the reading skills and understanding of programs written by others.

Due to its complexity programming seems to be not very attractive, that is why, in order to engage the learners, it is necessary to present it as easy as possible, in a clear and attractive way. However, no matter how attractively it would be presented, it is not enough to have only knowledge or good patterns, for it is necessary to actively engage oneself in this process, to develop the skills, to think logically and algorithmically. Thus, often skills in creating algorithms (which is an integral part of programming) are implicitly developed in the junior school grades already - by analysing real-life problems, splitting them into smaller tasks, reasoning solution options as well as making synthesis of the results to obtain a general solution to the problem. The acquired thinking skills help to understand the essence of programming and make programming learning more productive (Mayer et. al. 1986).

In a range of literature (Gomes, Carmo, etc., 2006; Bennedsen 2008; Jenkins 2002; Mannila 2007) the causes that determine programming learning problems have been set out:

- ✓ it is difficult to understand program's objectives and their relationship with the computer;
- ✓ it is difficult to understand the specific programming language's syntax and semantics;
- ✓ incorrect understanding of programming constructs;
- ✓ inability to resolve the problems;
- ✓ inability to read and understand the code of the program.

Learning success depends on how the maximum learning goals are being achieved, i.e. whether the necessary knowledge and skills are being acquired, and what emotions are experienced by learners during the learning process. Learning success is to a large extent determined by learning efficiency, which depends on the willingness to learn and knowledge how to learn. It is also influenced by an attractive learning environment.

However, regardless of attractive learning environment good learning results are still determined by learner's personal qualities and his learning style. There are many classifications of learning styles (Hawk and Shah 2007). For programming training are most important the classifications of Herrmann and Felder-Silverman.

Herrmann Brain Dominance Instrument (HBDI). This method classifies students in terms of their relative preferences for thinking in four different modes based on the task-specialized functioning of the physical brain. The four modes or quadrants in this classification scheme are

- *Quadrant A* (left brain, cerebral). Logical, analytical, quantitative, factual, critical;
- *Quadrant B* (left brain, limbic). Sequential, organized, planned, detailed, structured;
- *Quadrant C* (right brain, limbic). Emotional, interpersonal, sensory, kinesthetic, symbolic;
- *Quadrant D* (right brain, cerebral). Visual, holistic, innovative.

Most engineering instruction consequently focuses on left-brain Quadrant A analysis and Quadrant B methods and procedures associated with that analysis, neglecting important skills associated with quadrant C (teamwork, communications) and quadrant D (creative problem solving, systems thinking, synthesis, and design).

Felder-Silverman Learning Style Model. This model classifies students as:

- *sensing learners* (concrete, practical, oriented toward facts and procedures) or *intuitive learners* (conceptual, innovative, oriented toward theories and meanings);
- *visual learners* (prefer visual representations of presented material--pictures, diagrams, flow charts) *or verbal learners* (prefer written and spoken explanations);
- *inductive learners* (prefer presentations that proceed from the specific to the general) or *deductive learners* (prefer presentations that go from the general to the specific);
- *active learners* (learn by trying things out, working with others) or *reflective learners* (learn by thinking things through, working alone);
- *sequential learners* (linear, orderly, learn in small incremental steps) or *global learners* (holistic, systems thinkers, learn in large leaps).

For the past few decades, most engineering instruction has been heavily biased toward intuitive, verbal, deductive, reflective, and sequential learners. However, relatively few engineering students fall into all five of these categories. Thus most engineering students receive an education that is mismatched to their learning styles. This could hurt their performance and their attitudes toward their courses and toward engineering as a curriculum and career.

Computer Science Curriculum 2008 defines the main sections and topics that must be trained in a Programming Fundamentals course (*Fundamental Constructs; Algorithmic Problem*

Solving; Data Structures; Event Driven Programming; Recursion; Object Oriented; Foundations Information Security; Secure Programming;). The knowledge area of Programming Fundamentals includes those skills and concepts that are essential to programming practice independent of the underlying paradigm. As a result, this area includes units on fundamental programming concepts, basic data structures, algorithmic processes, and basic security. There are defined to each topic appropriate learning outcomes. The learning outcome corresponds to some one of Bloom's taxonomy level (see Table 1). In learning process, it is important to gradually move all levels of Bloom's taxonomy. Accordingly, the teaching material and knowledge area appropriate learning objects must be elected considering to Bloom's level.

Table 1: Bloom's taxonomy levels description

Level	Category	Cognitive Processes
1.	Remember	recognizing, recalling, describing, stating
2.	Understand	interpret, exemplify, classify, infer, compare, explain, paraphrasing, summarizing
3.	Apply	execute (i.e. carry out), implement (i.e. use), compute, manipulate, solve
4.	Analyze	differentiate, organize, attribute, discriminate, distinguish, sub-divide
5.	Evaluate	check, critique, assess, compare, contrast
6.	Create	generate, plan, produce, innovate, devise, design, organize

4. Methods

1. Literary analysis, research analysis and technology analysis are used for understanding of existing situation in the world and for finding the best solutions.
2. Questionnaire surveys will be used for gathering various opinion related to my research area.
3. Modeling using UML will be used for create the tool's project.

5. Current progress

We propose the method for electing the knowledge are appropriate learning objects from *Learning Objects Repositories* according student's learning styles.

- This method will allow the selection of instructional material and tasks in accordance with the learner's needs and abilities, learning experience and learning styles;
- Designed adaptive programming teaching tool will assess level of learner's achievement and increase the effectiveness of programming learning.

To find methods to elect learning objects according student's learning styles, it will be defined learning objects characteristics that are appropriate to knowledge area. Also, it will be appointed relation between student's learning styles and knowledge area appropriate learning objects. For electing the knowledge are appropriate learning objects from *Learning Objects Repositories* will be defined relation of the features of RDF (*Resource Description Framework*) and MLR (*Metadata for Learning Resources*). It will be defined how to find learning objects according RDF schema.