Educating future coders with a holistic ICT curriculum and new learning solutions

Pia NIEMELÄ
Computer Science, Tampere University of Technology
Tampere, Finland

Cristiano DI FLORA
Rovio
Helsinki, Finland

Martti HELEVIRTA
Tampere, Finland

and

Ville ISOMÖTTÖNEN
Mathematical Information Technology, University of Jyväskylä
Jyväskylä, Finland

ABSTRACT

Technology-orientation and coding are gaining momentum in Finnish curriculum planning for primary and secondary school. However, according to the existing plans, the scope of ICT teaching is limited to practical topics, e.g., how to drill basic control structures (if-then-else, for, while) without focusing on the high level epistemological view of ICT. This paper proposes some key extensions to such plans, targeted to highlight rather the epistemological factors of teaching than talk about concrete means of strengthening the craftsmanship of coding. The proposed approach stems from the qualitative data collected by interviewing ICT professionals (N=7, 4 males, 3 females), who have gained experience of the industry needs while working as ICT professionals (avg=11.3 y, s=3.9 y). This work illustrates a holistic model of ICT teaching as well as suggests a set of new methods and tools.

Keywords: ICT curriculum, teaching ICT in primary and secondary school, concept maps, UML, holistic ICT model

1. INTRODUCTION

The new curriculum with information and communication technology (ICT) as its focus is currently being reviewed and prepared for publication. The need of more ICT experts in industry has been recognized in decision-making by governing bodies. Not only are various domestic directions promoting ICT education but also the EU and multinational corporations have been actively pursuing new instructions and assessment of e-skills. For example, the EU has outlined a strategy for improving e-skills for the 21st century to foster competitiveness, growth, and jobs.

Moreover, in Finland distinguished pedagogues of especially the University of Helsinki [1, 2] are promoting more student-centered, informal learning: tablets for all students, using online material and social media to co-create in order to gain better ICT and multi-literacy skills. Future needs have guided the planning of the becoming 2016 curriculum. The way of working and living is rapidly changing, and the need for curriculum change is acknowledged. Familiarizing students with technology and learning the basics of coding will be started already in primary school and the skills gained are further strengthened at the secondary level.

In the ICT curriculum, digital literacy and ICT skills are meant to be built gradually, starting from visual coding and tactile learning followed by a more formal approach at the secondary level, where ICT is integrated into math teaching. Hence, programmable calculators and other computational features are well represented in the curriculum plans. The introduction of new ICT concepts by experimenting relies on “Learning by doing” methodology. Graphical and other high-level languages with additional libraries meant for education are utilized.

Learning goals are divided as learning packages that consistently build up the basics of computer science a grade by grade at the secondary level. For example, the 7th grade aims at acquainting pupils with such computing fundamentals as statements, data types, the sequential execution of the program, ‘if-then-else’ flow control structures, and finding errors in syntax and correcting them. Basics of logic are introduced, starting from a truth value of a sentence. In the 8th grade, variables and functions are introduced. State machines are used and visualized concretely by playing with the construction kits (e.g. the switch states ON/OFF). Logic continues with deduction and reasoning. In the 9th grade, new variable classes such as collections, conditional iteration (while, do, for), and recursion are introduced. At a more general level, the learning goal is to model a problem and divide it into smaller executables. For gaining the craftsmanship of coding the planned approach sounds viable, but regarding of the whole palette of needed ICT skills the view is regrettably narrow.

2. RESEARCH QUESTIONS

The current proposal for ICT curriculum emphasizes gaining the craftsmanship of coding with small and valid incremental steps. The order of propagation is well-justified, but still arguable. Instead of addressing all the possible aspects of coding and computer science learning primitives in detail, the epistemology of the ICT teaching as a whole should be discussed to consider all essential higher level needs. It is important to ensure that these findings are sanity checked and that the discussion is not lead by pedagogues only, but validated and augmented by ICT professionals, who know the industry needs. The research questions to be replied are:

1. Which kind of model would give a more holistic view of ICT epistemology?
2. How to support the learners in their becoming not only good coders but also good software architects and designers?
3. Which kind of learning solutions would support the ICT teaching model proposed?

3. INDUSTRY NEEDS RULED IN

To get a better grasp of the current ICT landscape we interviewed seven ICT professionals by email, six of whom are software developers and one a program manager. The email questionnaire contained the following questions: What are the ICT skills needed today/in the future? Which are your best ICT courses/informal learning experiences? How should ICT be taught in the primary school? With seven replies, we are far from scientific significance and based on the data only rough recommendations can be given. However, the anticipated holistic model can be verified by referring to the answers. Based on the replies, we classified ICT related capabilities to four categories: the craftsmanship of coding, modeling, user-centered design, and project management. More generic skills such as critical thinking, future working qualifications, and global citizenship were also mentioned, however, not taken into account here as they were regarded more as all-encompassing, general capabilities to be taught in other subjects as well.

Since the majority of the interviewees represented the implementation side, the craftsmanship of coding perspective was well pronounced. In a list of needed skills, web computing was mentioned six times, followed by data structure & algorithms (3), testing (2), and mobile coding(1). Among the most common computer languages Java (2), JavaScript (2) and C++ (1) were listed, but also specialties such as Rust (1), Clojure (1) and Go (1) received votes. High-availability engineering (1) and the ability to develop games (1) were seen as useful system level capabilities. Modeling was mentioned four times: design, UML, architecture, and being able to recognize meaningful entities were listed. User-centered design occurred once in the form of "*understanding the needs of the customer and managing them*" emphasized by the only project manager involved, who considered also project management and selling as important skills.

Our interviewees regarded hands-on experience as the main building block in learning. They would include in the ICT lessons of primary school e.g. team work and pair-programming exercises, increase motivation and inspiration by providing good examples, combine ICT with sports, and have students build their own e-portfolios. Working in teams or in pairs kids would learn informally scaffolding each other in the zone of proximal development [3]. Regarding good learning experiences the importance of teamwork was emphasized (3), especially pair programming with friends having a similar level was considered rewarding.

From the formal side the basic courses in the beginning were found the most meaningful (2), and those teaching techniques that remain the same regardless of the language such as data structures and algorithms (3) were valued high. Nevertheless, we also received critical views regarding ICT teaching, for example: "*At high school I never attended any good ICT courses. But all the math and physics at school helped me to learn problem solving and how to break down a problem in multiple pieces.*" According to the interviews, the future is drifting in the direction of HTML5 (2), robots (1), internet of things (1), and visualizations (1). These findings were classified into four main categories.

The craftsmanship of coding

In the discipline of handicrafts and craftsmanship, learning happens through doing by hand, which is seen as a way of leveraging innovation and the creativity. Theories such as intelligent hands [4] and learning by doing, are the basis for the tactile learning language. In maths, the tactile exercises such as fraction pieces and decimal system learning tools are used while approaching the symbol language more in-depth. In ICT, bridging the connection between electronics and coding may be achieved with the help of different assembly kits (e.g. LEGO MindStorm and Robots, Arduino, Lilypad, littleBits). Electronic components, such as light emitting diodes, buzzers, and couplers can be controlled by coding and give a more concrete and clear response than visible feedback on a computer screen. As one of our interviewees puts it, "*Learning by doing simply cannot be beaten in efficiency.*"

In addition to construction kits, visual programming languages may be used as primers. Scratch, for example, provides graphical support for a user preventing the faulty code or the connecting of incompatible code sequences. Control structures (such as if-then, for, while) are ready-built, a user only has to adjust parameters, such as counters in iteration loops. Visual programming languages are limited in freedom of degrees, which at the initial learning stage will be good to minimize the cognitive load: time is not wasted hunting syntactic errors. In the long run, the conciseness of such languages starts to restrict freedom and creativity indicating the due date to expand to more expressive programming languages.

Conceptual modeling as a software architect

On the authority of our interviewees, the development of ICT talent requires strong modeling and conceptualization skills. The highlighted modeling skills were designing, mastering UML modeling language, being able to separate relevant entities and build an architecture of systems. Regarding thinking skills mentioned 'logical and critical thinking' overlap partly with the conceptualizing skills, too. Therefore, we propose conceptual modeling as one of the key expertises and concept and mind mapping as its preceding preparatory skill. However, having good conceptualizing and modeling skills is not useful only in ICT, but in deep learning in general, the biggest difference between expert and novice thinking being the consistency and density of underlying concept schemes. Deeper learning implies linking atomic details as bigger and more robust constructions. Tying new knowledge to relevant concepts and previous propositions makes learning more meaningful.

User-centered design to take into account real user needs

The domain of user-centered design was duly emphasized by the only project manager interviewed ("*understanding customer's needs and use*", "*selling*"). A real innovation takes the user's needs into account, it focuses on the user and his context and incorporates his perspectives during the whole design process and new applications may be seen as innovations. Future coders need user-centered design tools to be able to innovate apposite applications. When the design is to be more user-centered, an essential prerequisite is to know how a customer acts by perceiving the typical work sequence and processes. This can be done by observing the environment and interviewing users i.e. becoming more aware of user needs.

In software projects, when the snapshot of the situation in its entirety has been obtained and user needs are detected as detailed manner as possible, the needs are dressed as formal use cases and requirements, which are a starting point to the following implementation. Ultimately, the final project achievements are checked against the use cases. When successful, the intended new solution provides added value and more efficiency by going beyond current practices.

Project management

The golden rule of project management consists of planning, organizing and controlling, efficiency being the ruling principle both time and money-wise, as our project manager states: - Line out, what is needed and how to get the most efficient (both workload and budget) solution. Less is needed in direct implementation, more having the whole picture with technical knowledge about possibilities.

Project management is also seen as an interface between the customer and the team and good communication between different stakeholders is his responsibility. In addition to taking care of the good communication within the team it is also crucial to involve the customer in the communication loop to ensure that the user needs are fulfilled. Often the needs get more detailed - or even changed - during the process. The current de facto standard in project management is an agile methodology that embraces self-organizing teams that are capable of managing themselves. More and more, in agile projects people are working in remote teams that are self-directed.

Agile methodologies target flexibility in taking into account the moving target. A customer may change his mind during the implementation that is called requirement volatility. By iteratively ensuring that the direction is right and the product will better respond to the need, the project management is linked with the user-centered design, too. Albeit of the biased sample (only one project manager interviewed) the interview data confirmed what was expected i.e. that the project management is more concerned about the user needs and communication with the user than developers. Interviews also suggest that the project management level is more situation aware and concerned about retrieving the big picture, whereas developers think more in terms of technical solutions.

4. THE HOLISTIC ICT TEACHING MODEL AND NEW LEARNING SOLUTIONS

The proposed teaching model is depicted in Figure 1. The most elementary building block, the craftsmanship of coding, starts with elementary exercises that combine both the visual coding and tactile learning objectives in order to provide a robust hands-on experience to build the base. When targets gradually grow more complicated or a bigger team is involved in coding, students will have to learn how to model and easily communicate the system structure. To this aim we propose the building block of conceptual modeling. The third thread illustrates the need for user-centered innovations. To be able to innovate, the student has to make observations in order to become acquainted with the customer needs, and to depict the underlying processes. After that optimizing and improving them is enabled, and ultimately, students come up with new, more efficient approaches to the problems and challenges.

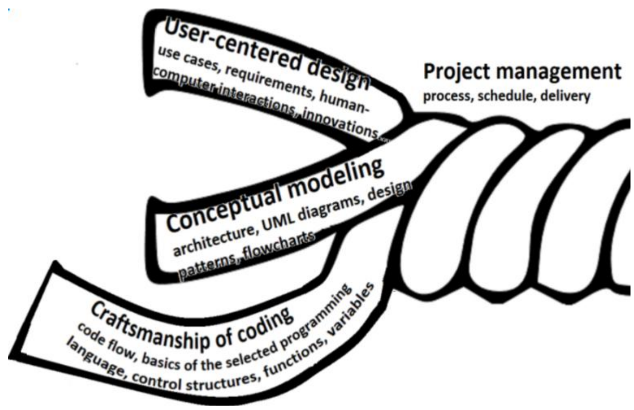

Figure 1: The proposed holistic model for ICT curriculum in primary and secondary school.

Together the threads form the cord of an ICT professional more capable of handling ICT projects successfully. The entwined threads are woven together by project management which includes controlling the process (e.g. using agile methods), being able to divide the project in smaller tasks, scheduling and staying within the deadlines. Next we will examine how these missing threads of current ICT curriculum of modelling, user-centered design and project management may be introduced in the school environment.

Various concept map techniques as knowledge building tools

In computing good software architects are good modelers, whereas in school excellent students are good conceptualizers. In recent years, concept maps have been recognized as an effective visual learning tool that helps learners memorize and organize knowledge. Åhlberg [5] recommends maps in situations requiring data parsing and argues that they illustrate the conceptual and propositional structure of written text. He also considers conversions to both directions, from text to a map and vice versa, as a good way to work on and elaborate meanings. In addition to visually appealing and easy to use concept map applications (bubbl.us, MindMup, CMapTools), the tools of ICT professionals such as the Unified Modelling Language (UML) may be introduced to students as a modelling tool that is applicable in ICT teaching in particular.

Modeling skills are important for communication purposes, too, both between developers and other stakeholders. Nowadays, as the development is often geographically distributed, communications skills are extremely necessary. Even if one is not going to be a developer himself, a better understanding about the overall structure with a map representation gives a quick generic overview and more means to communicate with developers.

In education, we also promote the use of concept mapping as an assessment tool. Currently, assessment is based on the student's test results and activity during lessons. This method is good at measuring whether smaller learning objectives have been reached. However, a more detailed development of conceptual understanding and getting a bigger picture is less frequently examined. Meaningful learning results are achieved when a person consciously and explicitly ties new knowledge to relevant concepts and previous propositions they already possess. Deeper learning implies linking atomic details as bigger and more robust constructions. A concept map can be understood as a visualization of a mental concept construction.

Authentic tasks facilitating user-centered design

Often real world problems are open ended, complex, require extended knowledge and emergent solutions. According to the Engagement Learning Theory by Kearsley and Shneiderman [6], a valid way of getting students committed is to provide meaningful, creative, and authentic tasks, which reflect the interests of students themselves, as much as possible. Finding an achievable yet interesting project target can prove to be the hardest part of the whole project, but once the goal is set, the group may start working backward from it [7], e.g. planning, sharing the goal into smaller executables, scheduling these parts, and finally moving to implementation.

User-centered design implies the participatory methods of inquiry of real - not self-determined - customer needs, modeling the situation, and means of improving current practices. For younger students, gaining the required level of situation awareness is a meta-cognitively demanding task and requires such future working skills as communication and collaboration. In software projects, when the snapshot of the situation has been obtained, the demands are dressed as formal use cases or lighter user stories and requirements, which are a starting point to the following implementation.

The authenticity of the project is increased with a real customer, who is interested in getting the thing done, is waiting for the end product and giving feedback. A real and useful innovation takes user's needs into account; it focuses on the user and his context and incorporates his perspectives during the whole design process, hence providing an opportunity of producing what Scardamalia et al. [7] call emergent outcomes, i.e. through new considerations, creativity and sustained work something brilliant might emerge. However, the emphasis here is not on the outcome but rather to familiarize with project work practices, learn how to co-operate, communicate, and take responsibility.

Nousiainen [8] emphasizes that the continuous involvement of users is a goal as such but also a way to empower users and promote workplace democracy and the means to practice the working life skills of participation, collaboration, and communication. In some schools, students have already started to innovate e.g. new means to recycle [9]. If no customers are available, suitable activities such as school projects could be e.g. doing your own online textbook or encyclopedia or pilot tests of innovative new technologies and rating them. Often the publicity of putting the end result on the web provides a sufficient incentive for the students to do their best. Strategically oriented students might see this also as a sweet spot of exposing their skills for potential employers and polish their CV.

Engaged students as self-directed project managers

Several educators agree along with industry representatives that there is a gap between current education and future work skills. Moylan [10] identifies "Project Learning" as a key methodology for closing this gap between current curriculum and developing their necessary knowledge and skills essential for success in the 21st century. Among other forces, President Obama also instructs schools to transfer from trivial bubble filling exercises to such 21st century skills like problem-solving, critical thinking, entrepreneurship and creativity: the school has to train children for future challenges and work [11].

Project management may be practiced by open assignments i.e. with school projects that have to be scheduled and delivered in a timely fashion. In such projects, students should be responsible for the acquisition of information needed, for example, by interviewing their intended customer or searching data from the net. It would be good to attempt to raise the abstraction level by modeling. For schools to adopt agile project working style, students would come together to work on an ICT project that they have selected by themselves, plan and share the work, take on roles that play to their strengths and interests and then implement and solve problems together till the project is done.

Being ultimately learner-centered this type of learning assignment requires a very different approach of the teacher compared to traditional classroom instructions. The teacher is no longer the lecturer and the decider, but rather a facilitator or coach learning alongside students; sometimes to give up the control may feel like a farmer that is selling his farm. However, the effectiveness of instructor-led lessons and lecturing has long been questioned and as a subject ICT is one of the best suited for applying project oriented learning style. However, we note that often transferring to an open-ended working mode requires some practicing and external pressure to get things done, whether the agent generating the pressure is a teacher or preferably an external customer.

5. CONCLUSIONS

Finland is planning to enrich its primary and secondary school curriculum with coding in order to prepare students for the future working life. Instead of settling on only the basics of coding, e.g. code flow and control structures, we claim that the ability to innovate and design software systems is at the very heart of software engineering. By adding the key areas of modeling and user-centered design to instruction schemes, we create a more holistic ICT curriculum. The conception and modeling ability is needed not only in ICT but in knowledge building and conceptual thinking in general. User-centered design improving the practices of one's own environment may also be seen as a tool of empowerment. An empowered member of the society, who is aware of user needs, will also become more innovative.

Hands-on experimentation was also considered beneficial among our interviewees: games, pair-programming, and learning from others informally were seen as ways to foster learning and engagement. Innovativeness and creativity are buzzwords used in curriculum planning, often in accordance with arts and crafts, which are assumed to enhance them. With new methods and learning solutions, creativity may be fostered with STEM subjects as well: building robots, making animations, and playing and even developing games (e.g. Angry Birds Space to assimilate gravity basics) are new, engaging and motivating ways of learning.

Since many students are passionate about playing, games as new learning solutions have proven to be very powerful. High motivation and engagement appear as a “flow” while playing. Fu et al. [12] examined engaging games and listed properties such as immersion, the clarity of the goal, autonomy, feedback, challenge, and social interaction as the ingredients of flow. Moreover, it has been reported that games can have a positive impact on pupils’ perceptual templates, knowledge acquisition and affective outcomes [13]. By including the suitable features of games in learning environments, serious education may transform to edutainment. The level of shared fun increases interest in ICT in general and ICT learning objects may be gamified, too.

Coding starts in Finnish schools in the autumn of 2016. ICT classes become a laboratory of new learning tools and methodology. By documenting experiments and applying continuous development cycles, we may iteratively improve the learning results. In addition to teaching to code, it is necessary to introduce new tools for modeling and user-centered design.

6. REFERENCES

[1] K. Hakkarainen, L. Hietajärvi, K. Alho, K. Lonka, K. Salmela-Aro, "A Paradigmatic Analysis of Contemporary Schools of IS Development", **International Encyclopedia of the Social & Behavioral Sciences**, pp. 918-923. 2015.

[2] L. Krokfors, "Learning. Creatively. Together", **Educational Change Report 2016**. 2015.

[3] L.Vygotsky, "Interaction between learning and development", **Readings on the development of children**, 23.3: pp. 34-41. 1978.

[4] L. Hyde, **Making it** , New York Times, 6. 2008

[5] M. Åhlberg, **Concept maps as a research method**. 2002.

[6] G. Kearsley and B. Shneiderman, **Engagement Theory: A Framework for Technology-Based Teaching and Learning**. 1999.

[7] M. Scardamalia, J. Bransford, B. Kozma and E. Quellmalz, "New assessments and environments for knowledge building," in **Assessment and Teaching of 21st Century Skills**. Springer, pp. 231-300. 2012.

[8] T. Nousiainen, **Children's involvement in the design of game-based learning environments**. 2009.

[9] M. Aineslahti, **A Journey in the Landscape of Sustainable School Development**. 2009.

[10] W. A. Moylan, "Learning by Project: Developing Essential 21st Century Skills Using Student Team Projects", **International Journal of Learning**, vol. 15. 2008.

[11] B. Trilling and C. Fadel, **21st Century Skills: Learning for Life in our Times**. John Wiley & Sons. 2009.

[12] F. Fu, R. Su and S. Yu, "EGameFlow: A scale to measure learners' enjoyment of e-learning games", **Computers & Education**, vol. 52, pp. 101-112. 2009.

[13] T. Connolly, E. Boyle, E. MacArthur, T. Hainey, & J. Boyle, "A systematic literature review of empirical evidence on computer games and serious games", **Computers & Education**, 59(2), 661-686. 2012.