Commentarii informaticae didacticae | 7

Torsten Brinda | Nicholas Reynolds |
Ralf Romeike | Andreas Schwill (Eds.)

KEYCIT 2014

Key Competencies in Informatics and ICT

University of Potsdam, Germany, July 1st–4th, 2014

Universitätsverlag Potsdam

Commentarii informaticae didacticae (CID)

Commentarii informaticae didacticae (CID) | 7

Torsten Brinda | Nicholas Reynolds | Ralf Romeike |
Andreas Schwill (Eds.)

KEYCIT 2014

Key Competencies in Informatics and ICT

University of Potsdam, Germany, July 1st–4th 2014

Universitätsverlag Potsdam

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.dnb.de.

Universitätsverlag Potsdam 2015
http://verlag.ub.uni-potsdam.de/

Am Neuen Palais 10, 14469 Potsdam
Tel.: +49 (0)331 977 2533 / Fax: 2292
E-Mail: verlag@uni-potsdam.de

ISSN (print) 1868-0844
ISSN (online) 2191-1940

Typesetting: Kristin Schettler, Universitätsverlag Potsdam
Print: docupoint GmbH Magdeburg

ISBN 978-3-86956-292-6

Also published online
at the Institutional Repository of the University of Potsdam:
URN urn:nbn:de:kobv:517-opus4-70325
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-70325

Preface

In our rapidly changing world it is increasingly important not only to be an expert in a chosen field of study but also to be able to respond to developments, master new approaches to solving problems, and fulfil changing requirements in the modern world and in the job market. In response to these needs key competencies in understanding, developing and using new digital technologies are being brought into focus in school and university programmes. According to a definition by H. Orth (1999) key competencies are acquirable general skills, attitudes and knowledge elements, which are beneficial in the solution of problems and the acquisition of new skills in as many domains as possible, so that an ability to act emerges, which makes it possible to fulfil both individual and societal needs. In the context of the OECD project "Definition and Selection of Competencies – DeSeCo" broad categories for key competencies were defined as ability to: Use tools interactively (e.g. language, technology); Interact in heterogeneous groups; and Act autonomously. Informatics and ICT play an important role in the solution of problems and the acquisition of new skills as well as in all the DeSeCo categories. Therefore it is valuable and necessary to address the combination of key competencies, Informatics and ICT in detail, which was the main focus of the conference "KEYCIT – Key Competencies in Informatics and ICT (KEYCIT 2014)".

To provide a forum to present and to discuss research, case studies, positions, and national perspectives in this field, the IFIP TC3 event KEYCIT 2014 was held at the University of Potsdam in Germany from July 1st to 4th, 2014. The conference was organized into strands focusing on secondary education, university education and teacher education (organized by IFIP WGs 3.1 and 3.3) and was accompanied by parallel conference streams on "Key Competencies for Educating ICT Professionals (KCICTP 2014, organized by IFIP WG 3.4)" and on "Key Competencies in Informatics and ICT: Implications and Issues for Educational Management (ITEM 2014, organized by IFIP WG 3.7)".

Around 90 experts from 28 different countries all over the world visited the conference and took the opportunity to discuss the subject with colleagues. The presentations selected out of around 80 submissions covered questions like: What are the key competencies in Informatics and ICT of students and educators? How can such key competencies be derived, even theoretically? How can key competencies be modelled in competence structure models? How can they be measured using competence level models? How can key competencies be taught in motivating ways? Three invited lectures by Johannes Magenheim (GER) and Sigrid Schubert (GER) on their experiences with competence modelling and measurement in the fields of system comprehension and modelling, by Paul Curzon (UK) on his work in the well recognized projects “Computer Science for fun – CS4fun” and the “Magic of Computer Science” and by Ivan Kalas (SL) on his experiences in developing key competencies in Informatics and ICT in primary education completed the program. These keynotes as well as all submissions accepted for the main conference are compiled in this book. All submissions accepted for the streams KCICTP and ITEM are compiled in a separate book edited by Don Passey and Arthur Tatnall (published by Springer).

For the review process, an international program committee of experienced and recognized members of IFIP TC3 was formed and supplemented by additional reviewers also from this community. The reviews were organised according to content criteria and peer reviewed through a double-blind process according to scientific standards.

We would like to thank the members of the program committee and the additional reviewers for their work, the authors for submitting their work to KEYCIT and all participants for attending the conference and therewith contributing to its success.

Torsten Brinda
PC Chair

Peter Micheuz, Mary Webb
PC Co-Chairs

Committees

Program Committee
Torsten Brinda, University of Duisburg-Essen (GER) (chair, editor)
Yvonne Büttner-Ringier, IT Education Consulting (CH)
Andrew Fluck, University of Tasmania (AU)
Marie Iding, University of Hawaii (US)
Ulrich Kortenkamp, University of Halle Wittenberg (GER)
Joberto Martins, Salvador University (BR)
Peter Micheuz, University of Klagenfurt (AT) (co-chair)
Wolfgang Müller, University of Education, Weingarten (GER)
Nick Reynolds, University of Melbourne (AU) (editor)
Ralf Romeike, University of Erlangen-Nuremberg (GER) (editor)
Sindre Røsvik, Volda University College (NO)
Raul Wazlawick, University Federal de Santa Catarina (BR)
Mary Webb, King's College London (UK) (co-chair)

Organizing Committee (University of Potsdam, GER)
Sabine Hübner
Simon Kiertscher
Matthias Knietzsch
Mareen Przybylla
Andreas Schwill (chair)
Wolfgang Severin

Additional reviewers:
Andrej Brodnik, Barbara Demo, Birgit Eickelmann, Anthony Gioko, Monique Grandbastien, Pieter Hogenbirk, Steve Kennewell, Cathy Lewin, Izabela Mrochen, Bob Munro, Sigrid Schubert, Bert Zwanefeld

Table of Contents

Keynotes

Full Papers

Short Papers

Posters

Workshops

Tutorial

Keynotes

Unplugged Computational Thinking for Fun

Paul Curzon
Queen Mary University of London
Mile End
London E1 4NS. UK.
p.curzon@qmul.ac.uk

Abstract: Computational thinking is a fundamental skill set that is learned by studying Informatics and ICT. We argue that its core ideas can be introduced in an inspiring and integrated way to both teachers and students using fun and contextually rich cs4fn 'Computer Science for Fun' stories combined with 'unplugged' activities including games and magic tricks. We also argue that understanding people is an important part of computational thinking. Computational thinking can be fun for everyone when taught in kinaesthetic ways away from technology.

Keywords: Computational thinking, cs4fn, 'unplugged' computing, kinaesthetic teaching, fun

1 Introduction

Computational thinking was popularised by Wing (2006) as a unique set of key competencies that students learn from studying Computer Science/Informatics. Rather than being a single skill, it is an integrated set of approaches to problem solving including algorithmic thinking, logical thinking, abstraction, generalisation, pattern matching, and evaluation. We argue that understanding people is also an important part of computational thinking based problem solving: computer scientists ultimately solve problems for people and those solutions have to, therefore, work for people.

Whilst computational thinking skills may be gained in a traditional way, just by studying for an Informatics degree, Wing raised the question of how it can be taught explicitly and especially at primary and high school level (Wing 2008). This has become a timely question as a variety of countries are increasing the proportion of computing in their school curriculum. In the UK, for example, from September 2014 a new computing syllabus replaces the past

ICT syllabus that focused on the use of technology (Department for Education, 2013). Computational thinking is a core aspect of the skills students will be expected to learn. Similar initiatives are being introduced in other countries too also with computational thinking playing a central role (e.g., Denmark, (Caspersen, Nowack, 2013), New Zealand (Bell, Andreae, Lambert, 2010).

However, many students even those self-selected at university level struggle to learn to program in depth, so teaching everyone to program well based on a deep understanding of the linked skill set, at pre-university level is a challenging endeavour. If programming is a precursor to learning computational thinking because it is by learning to program that one gains computational thinking skills, then following this approach seems unlikely to be successful as a way to learn those skills except for the best students. It is hard to gain deep computational thinking skills or even understand what they involve, from writing small simple programs alone. Therefore, by following this approach the skills will be learnt in any depth only late in the educational system.

Furthermore, while programming is a core part of computing, the subject is much more than just programming, as is computational thinking. Computing is a naturally interdisciplinary subject that if taught that way can be of interest to a much wider range of students (including girls), not just those interested in programming for its own sake. A variety of projects have explored ways to introduce computing without the focus being just on programming (Bell, Curzon, Cutts, Dagiene, Haberman, 2011). Unplugged approaches (Bell, Fellows, Witten, 1998; Bell, 2000; Curzon, McOwan, Cutts, Bell, 2009) – essentially constructivist, kinaesthetic activities – are one such approach. They have proved an extremely popular way to introduce basic computing concepts and enthuse and motivate young students about computing. Whilst the original focus of unplugged techniques was on primary school students, they have been shown to be a successful way of inspiring students of all ages (Curzon, McOwan, Cutts, Bell, 2009).

Given computational thinking is a fundamental skill set, we are exploring ways to introduce it in a way that is fun and relevant to children of all ages with a wide range of interests and educational backgrounds, not just those initially interested in programming. It also potentially provides an alternative more general focus than programming around which to structure the early teaching of computing. We are therefore interested in how one can introduce the overall idea of computational thinking as a coherent skill set, for younger age groups who still have limited computing experience. We are also concerned with how to inspire students about computational thinking in its own right, so that they want to learn the skills independent of learning to program. We believe unplugged activities, linked to contextually rich storytelling, is potentially a powerful way to achieve this.

2 Discussion

We have been using two interlinked approaches. The first is the idea of telling offbeat 'Computer Science for Fun' stories and the second is the idea of using 'unplugged' activities. We discuss the background of each briefly below, before describing two specific examples. We then outline recent evaluation both of our overall project and of workshops with teachers specifically about computational thinking.

2.1 Telling Offbeat Stories with Unplugged Activities

Our work telling fun, computing stories arose out of the cs4fn project (Curzon, 2007; Curzon, McOwan, 2008; Curzon, Black, Meagher, McOwan, 2009; Myketiak, Curzon, Black, McOwan, Meagher, 2012; Meagher, Curzon, McOwan, Black, Brodie, 2013). It is a public engagement project aiming to inspire school students about interdisciplinary computing. It consists of magazines (Curzon, McOwan, Black, 2005–2014), booklets such as on computer science magic tricks (McOwan, Curzon, 2008; McOwan, Curzon, Black, 2009), women in computing (Black, Curzon, Myketiak, McOwan, 2011) and computational thinking (Curzon, 2014). There is also a linked website (cs4fn, 2005) containing further articles.

Across all these outputs we have focussed on telling playful, contextually rich and offbeat stories about interdisciplinary research. By using research stories we focus on leading edge ideas and technology showing the future of the subject. The playful nature using quirky links along with the narrative story structure helps make them engaging. A focus on interdisciplinary problems widens the potential interest beyond just those already interested in computing. By making stories contextually rich we ensure they lead to deeper understanding and can be open ended with more potential avenues to explore beyond the core problem. It also grounds the stories and linked problems much more in the real world, rather than them having the feel of puzzles. Where possible we also make sure the stories are not reliant on existing knowledge meaning they work across a wide agerange and ability. We do not write 'for children' but in a highly accessible style, even about technical topics, again allowing them to work for a wide audience.

The original focus of cs4fn was on written stories, however this led to a strand of linked, fun interactive talks and shows on topics such as Artificial Intelligence, the Magic of Computing (Curzon, McOwan, 2008; Curzon, McOwan, 2013) and computational thinking. These talks and shows, given to student groups in schools and at science festivals, adopted a variation of the 'unplugged' style of teaching (Bell, et al., 1998; Bell, 2000). Kinaesthetic activities are used to explain computing concepts without technology. An important aspect of the way we have presented the activities is to build them into contextually rich stories in a similar way to the written articles. Initially most of these talks focussed directly on computing concepts with computational thinking ideas implicit. More recently, we are using them to more explicitly introduce computational thinking itself. We are also writing a series of linked booklets that tell the full stories of the talks in a way that makes computational thinking explicit (e.g., Curzon, 2014).

We are now taking this approach a step further, providing resources such as props and activity sheets for teachers to use to teach computational thinking. This is being done through our Teaching London Computing project (2014). It is a computing education project aiming to support teachers in London preparing to switch from the past curriculum that focussed on basic digital literacy and the use of computers to a new computer science based curriculum. Despite the project being London focussed, the resources are being made freely available to others through the Teaching London Computing website. To date we have written up a range of activities and given linked workshops focussed on secondary schools (age 11–18) around four themes:

- Introducing computational thinking
- Algorithmic thinking
- Unplugged programming
- Computational thinking: understanding people

We have also given workshops on how these unplugged activities can be used to introduce computational thinking at primary school level (up to age 11). An evaluation of these workshops is discussed below.

2.2 Sample Stories with Linked Unplugged Activities

To illustrate the approach we describe two of our stories and linked activities: one around helping a person who is totally paralysed to communicate, the other around magic tricks and human error.

As a way of introducing computational thinking in general we have used the story of Jean-Dominique Bauby. He had locked-in syndrome, a condition resulting from a stroke that leaves the person totally paralyzed. Despite this, Bauby wrote an autobiographical book about life with the condition (Bauby, 1997). We explore the problem of how he wrote the book and more generally how computational thinking might lead to people with locked-in syndrome being able to communicate more easily. We overview that story below; a full version is given in Curzon (2014). We use it as a way of introducing a range of computational thinking ideas in a real-world problem solving setting but in the absence of technology.

Bauby could see, hear and think but not talk. The only movement he could make was to blink one eye. He had a human helper to aid writing the book, though had no technological help at all. In this story, we explore with the class ways of communicating with locked-in syndrome including issues such as the need for some kind of agreed code. We describe how Bauby had the helper read out the letters A, B, C,... until Bauby blinked. They would write that letter down and then start again. We get the audience to try this in pairs as an unplugged activity asking them to think about problems and improvements. They usually come up with issues like the need to undo a letter and the need for a way to deal with punctuation and digits, for example. Improvements often suggested include starting with the most common letters (which Bauby actually did), and predicting the word before it is finished (taken from predictive texting). We then discuss how long it would take to write the book and look at best, worst and average case (for the basic case 13 questions per letter) in terms of number of questions asked (i.e., letters spoken).

In the final section we point out that we can do much better – at worst only 5 questions are needed to determine any letter of the alphabet. We note that even though they may not realise it everyone knows the right kind of question. We just need to switch problems to the game of 20 questions to show this (a further unplugged activity). We play a game where the speaker thinks of a famous person and the audience work out who it is by asking yes/no questions. From the start the audience ask questions like "Are they female?" not ones like "Is it Ghandi?". They can also normally say why that kind of question is better. Looking at how efficient this is we see it that always asking such halving questions only takes 20 questions to get from a million possibilities to 1. We can now transfer that solution to letters of the alphabet – the 5 questions used each just halve the remaining portion of the alphabet. The audience normally agree we have come up with a better way to communicate with locked-in syndrome. We finish however with a twist, pointing out that Bauby may have found blinking hard and if so we have made it 5 times harder as he must blink 5 times per letter with our solution rather than once with the algorithm he used. It is important to understand the problem from the perspective of the people involved before coming up with solutions.

This story introduces various aspects of computational thinking that we point out as we tell the story. It involves *algorithmic thinking* in coming up with an algorithm to communicate but with a clear focus on solutions that work for people, *evaluation* about the functionality, efficiency and usability of the solution, *pattern matching* and *generalisation* in translating solutions for other problems, *abstraction* for example in thinking about work to communicate one letter, rather than time to communicate the whole book, and so on. All the components of computational thinking are integrated, being used together to solve the problem – a problem that does not involve technology and that is contextually very rich.

A second group of stories we tell are based around magic tricks as the unplugged activity. The core idea is that magic tricks are essentially algorithms, though followed by a magician, rather than a computer. A trick is a series of steps that must be followed precisely and in the right order if the trick is to work. However the link is deeper than this. For a trick to work it needs more than the algorithm. It needs a good presentation based on an understanding of cognitive psychology. Similarly a program combines an algorithm with interaction design based on a similar understanding. Magic tricks can thus be used to teach the importance of taking people into account in computational thinking problem solving. They can also be used in teaching human-computer interaction topics (Myketiak, Curzon, McOwan and Black, 2012). A trick can be used as a demonstration before a discussion or the activity taken further. For example, the class might be set the task of writing down crib sheets for them to do the trick (i.e., write down the algorithm), or of creating new variations such as their own presentation based on the core algorithm of a trick demonstrated.

We have developed one such trick into a story about design to avoid human error in collaboration with CHI+MED (2009). It is a research project on how to design safer medical devices that help clinicians avoid making mistakes in their use. Magicians show how one can engineer a system – a magic trick – so that everyone makes the same mistake at the same time. Software engineers have to engineer systems in a way that ensures no one makes mistakes. One example we use is to present a trick called 'The Four Aces' which uses simple misdirection. In the trick, a set of Aces mysteriously jump from one pile to another without anyone noticing. We point out how this shows that we do not see everything that is in our field of view especially if our attention is drawn elsewhere.

We then consider an interface to a medical device such as an infusion pump which is set up by a nurse to deliver a given dose of a drug (say 15.5mg/hour over 2 hours). If the nurse's attention is drawn away from the screen when setting it up, for example because the start button is not close to the screen, he may not notice that the decimal point did not register, say, especially in a stressful and busy hospital ward. That could lead to the patient being given a massive overdose. Magicians use misdirection to pull our attention to the wrong place. A good interface designer will use similar tricks to make sure the attention is drawn to the right place – the screen of the medical device in this example.

This story also introduces a variety of aspects of computational thinking that we point out as we tell the story. It involves *algorithmic thinking* in creating the algorithms behind magic tricks, again with a focus on solutions that work for people. It also involves *evaluation* in checking that a trick really works in practice. The link between magic and programs is another example of *pattern matching* and *generalization* and this can be made explicit with tricks that are based directly on computer algorithms: search algorithms and error-correcting codes, for example. Students can be set the task of writing their own solutions and so have to decide on a suitable level of *abstraction* for the description. *Abstraction* can also be introduced in exploring arguments about whether a trick always works using an appropriately abstracted model. The separate aspects of computational thinking come together in an integrated way within the story, which is again contextually very rich and so open-ended.

2.3 Evaluation

The cs4fn approach of telling such rich, offbeat research stories has been extremely popular, in general. For example, demand for physical copies of the magazine is strong and has increased steadily. Over 1700 schools across the UK subscribe to copies, in many cases receiving class sets of 30–200 copies to distribute. The total number of magazines sent to subscribers amounts to around 18.000 copies. They are used in various ways according to the local need: with 'gifted and talented' groups, in normal classes and with 'problem' classes, to use directly in class and for students to read in their own time, and as a different kind of reading material placed in literacy boxes.

We have in the past sent physical copies to subscribers in over 80 countries (though unfortunately no longer have funding to continue to do this). In a survey of teachers conducted in 2012, 98 % rated the cs4fn magazine as either "excellent" or "good". 77 % agreed that they could use articles or ideas from cs4fn in their lessons.

The website has been similarly successful. Between 2008 and 2013 it has received over a million visits and PDFs of our magazines and other resources have been downloaded over 890.000 times. Web users have been positive in surveys with over two thirds of respondents saying cs4fn helped them see more ways computer science is used in the real world. After visiting the cs4fn website, they reported thinking of computer science as more interesting and thinking of a variety of careers that would use computer science.

In terms of live performances, we have given school shows on computing topics to nearly 20.000 school students in schools, reaching over 10.000 more at science festivals. Feedback from teachers has been highly positive about these shows: for example, in follow-up surveys 100 % of those surveyed said that they met their needs and that they would recommend them to others. The surveys also suggest teachers believe that students are more likely to take computing courses as a result. Individual surveys with students at the end of shows have also always been highly positive.

An external evaluation concluded that the cs4fn project as a whole has had impact in a variety of ways, including conceptual impacts (e.g., a more positive perception of computer science by students), instrumental impacts (e.g., in the form of long lasting resources), capacity building impacts (e.g., in teachers picking up both the content and style of teaching), attitude/cultural change impacts (e.g., at our home institution in elevating the importance of public engagement) and enduring connectivity impacts (e.g., through ongoing use of resources by a wide range of groups of people). A detailed description of the evaluation is given in Meagher, Curzon, McOwan, Black, Brodie (2013b) with summary in Meagher, Curzon, McOwan, Black, Brodie (2013a). The project has shown that the approach of using accessible writing in offbeat ways about interdisciplinary computing is a highly popular approach that does appear to inspire students and teachers. It remains further work to explicitly evaluate the magazines, booklets and shows in terms of their success specifically in introducing computational thinking ideas, however.

For our 'Teaching London Computing' workshops supporting teachers we have conducted an evaluation directly related to computational thinking. We gave out a short postevent feedback survey to teachers attending. Survey questions were on a 5-point Likert scale with a ranking ranging from 'strongly disagree' (1) to 'strongly agree' (5). In total 117 completed the survey. The feedback was extremely positive. Across all five workshops answers to the question: “The workshop was Useful” gave average response 4.57 (n=116). Similarly a question as to whether “The workshop was Confidence Building” gave average response 4.41 (n=113).

The surveys for the first and last workshop about introducing computational thinking and how understanding people mattered to computational thinking included explicit questions about computational thinking itself. The responses show that the workshops certainly helped participants understand what computational thinking was and give them useful ideas about how to teach it. Average responses (on the same 5-point Likert scale) were:

- “As a result of the workshop I now have *a better understanding of* computational thinking”: **4.28 (n=72)**
- “As a result of the workshop I now have *a better understanding of* computational thinking, about people” **4.63 (n=24)**
- “As a result of the workshop I have *new ideas about how to teach* computational thinking” **4.25 (n=68)**
- “As a result of the workshop I have *new ideas about how to teach* computational thinking, about people” **4.75 (n=24)**

Thus there is evidence that this approach of unplugged activities embedded in contextually rich stories is an effective way to explicitly introduce computational thinking ideas at least to teachers themselves. More detailed analysis of this data is in progress, however. It also remains further work to evaluate whether the approach leads to better understanding of students taught by those teachers.

3 Conclusion

Kinaesthetic “unplugged” activities embedded in contextually rich stories provides a potentially powerful way to introduce the ideas and skills of computational thinking in an integrated way that is accessible to all. The focus on understanding people as part of computational thinking helps add to that richness and power of the stories. We have argued that this approach provides a fun way to engage a wide range of students with computational thinking, not just those directly interested in coding. Evidence to date suggests the approach certainly does help teachers themselves understand the concepts and gives them confidence and ideas of how to teach it.

In future work we hope to investigate the approach’s effectiveness for teaching students as part of the curriculum. Selby, Dorling and Woollard (2013) are developing a framework for teaching and assessing computational thinking, within the UK curriculum. This is based around a set of core computational thinking concepts including abstraction, decomposition, algorithmic design, evaluation, and generalisation (Selby, Woollard, 2013). We intend to also integrate our activities and stories with that framework.

4 Acknowledgements

‘Teaching London Computing’ was created by Paul Curzon and William Marsh and is supported by the Mayor of London and Department of Education. cs4fn was created by Paul Curzon and Peter McOwan and funded by EPSRC (EP/F032641/1) (2008–2013). Both projects have been supported by Google’s CS4HS programme. The CHI+MED project is funded by EPSRC (EP/G059063/1). Nicola Plant, Jonathan Black, Jo Brodie, Chrystie Myketiak, Laura Meagher and many others have contributed to this work. We have also had helpful conversations about computational thinking with Mark Dorling, Michael Caspersen and Henning Agesen.

References

Bauby, J.-D. (1997): *The Diving-Bell and the Buttery*. Fourth Estate.

Bell, T., Fellows, M., Witten, I. (1998): *Computer Science Unplugged: Offline Activities and Games for All Ages*. Available at: www.lulu.com

Bell, T. (2000): A low-cost high-impact computer science show for family audiences. In *Austral-asian Computer Science Conference (ACSC 2000)*, pp 10–16.

Bell, T., Andreae, P., Lambert, L. (2010): Computer Science in New Zealand high schools. In *Proceedings of the Twelfth Australasian Conference on Computing Education* – Volume 103. Brisbane, Australia: Australian Computer Society, Inc.

Bell, T., Curzon, P., Cutts, Q., Dagiene, V., Haberman, B. (2011): Introducing students to computer science with programmes that don't emphasise programming. In *Proceedings of ITiCSE 2011, The 16th Annual Conference on Innovation and Technology in Computer Science Education ACM SIGCSE*, p. 391, June 2011. Darmstadt, Germany.

Black, J., Curzon, P., Myketiak, C., McOwan, P. W. (2011): A study in engaging female students in computer science using role models. In *Proceedings of ITiCSE 2011, The 16th Annual Conference on Innovation and Technology in Computer Science Education ACM SIGCSE*, pp. 63–67, June 2011. Darmstadt, Germany.

Caspersen, M. E., Nowack, P. (2013): Computational Thinking and Practice – A Generic Approach to Computing in Danish High School, In A. Carbone, J. Whalley (Eds.), *15th Australasian Computer Education Conference (ACE 2013)*, Adelaide, South Australia,. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 136. pp. 137–143.

CHI+MED (2009), CHI+MED: *Making Medical Devices Safer*, Retrieved from http://www.chi-med.ac.uk/cs4fn (2005), Computer Science for Fun, Retrieved from http://www.cs4fn.org/

Curzon, P. (2007): Serious fun in computer science. In *Proceedings of the 12th Annual Conference on Innovation and Technology in Computer Science Education ACM SIGCSE* (Invited Keynote), p. 1. ACM. Dundee, Scotland.

Curzon, P. (2014): *Computational Thinking Searching to Speak*. Queen Mary, University of London. Available from http://teachinglondoncomputing.org/resources/inspiring-computing-stories/

Curzon, P., McOwan, P. W. (2008): Engaging with computer science through magic shows. In *Proceedings of ITiCSE 2008, The 13th Annual Conference on Innovation and Technology in Computer Science Education ACM SIGCSE*, pp. 179–183. ACM. Madrid, Spain.

Curzon, P., Black, J., Meagher, L. R., McOwan P. W. (2009): cs4fn.org: Enthusing students about computer science. In T. O. C. Hermann, T. Lauer, M. Welte Eds), *Proceedings of Informatics Education Europe IV*, pp. 73–80. Freiburg, Germany.

Curzon P., McOwan P. W. (2013): Teaching formal methods using magic tricks. In *Fun with Formal Methods: Workshop at the 25th International Conference on Computer Aided Verification*, July.

Curzon, P., McOwan, P. W., Black, J. (2005–2014) cs4fn: *Computer Science for Fun Magazine*. Queen Mary University of London.

Curzon, P., McOwan, P. W., Cutts, Q., Bell, T. (2009): Enthusing and inspiring with reusable kinaesthetic activities. In *Proceedings of ITiCSE 2009, The 14th Annual Conference on Innovation and Technology in Computer Science Education ACM SIGCSE*, pp. 94–98. ACM, July. Paris, France.

Department for Education (2013): *The National Curriculum in England*, Framework Document. Available from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/210969/NC_framework_document_-_FINAL.pdf

McOwan, P. W. &. Curzon, P. (2008): *The Magic of Computer Science*. Queen Mary, University of London. Available from http://www.cs4fn.org/magic/magicdownload.php

McOwan, P. W., Curzon, P., Black, J. (2009): *The Magic of Computer Science II: Now we have your attention*. Queen Mary, University of London. Available from http://www.cs4fn.org/magic/magicdownload.php

Meagher, L. R., Curzon, P., McOwan, P. W., Black, J., Brodie, J., (2013a): *cs4fn Final Evaluation Executive Summary*. Queen Mary University of London. Available from http://www.cs4fn.org/project/evaluation/

Meagher, L. R., Curzon, P., McOwan, P. W., Black, J., Brodie, J., (2013b): *cs4fn Final Evaluation Report*. Queen Mary University of London. Available from http://www.cs4fn.org/project/evaluation/

Myketiak, C., Curzon, P., Black, J., McOwan, P. W., Meagher L. R. (2012): cs4fn: A flexible model for computer science outreach. In *Proceedings of the 17th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 297–302.

Myketiak, C., Curzon, P., McOwan, P. W., Black, J. (2012): Teaching HCI through magic. In *The Contextualised Curriculum: A CHI 2012 workshop*, May.

Selby, C., Woollard, J. (2013): *Computational Thinking: The Developing Definition* Available: http://eprints.soton.ac.uk/356481/.

Selby, C., Dorling, M., Woollard, J., (2014): *Evidence of Assessing Computational Thinking*, Submitted for publication.

Teaching London Computing (2014): *Teaching London Computing*, Retrieved from http://www.teachinglondoncomputing.org.

Wing, J. M. (2006): Computational thinking. *Communications of the ACM*, 49, pp. 33–35.

Wing, J. M. (2008): Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), pp. 3717–3725.

Biography

Paul Curzon is a Professor of Computer Science at Queen Mary University of London, interested in inspirational ways of teaching computing/computational thinking. He runs cs4fn enthusing students about computer science worldwide and 'Teaching London Computing' developing playful resources for computing teachers. He was made a UK National Teaching Fellow in 2010 in recognition of his excellence in teaching.

Programming at Pre-primary and Primary Levels: The Pipeline Can Start That Early

Ivan Kalas
Comenius University
Bratislava, Slovak Republic
kalas@fmph.uniba.sk

Extended Abstract: In my panel address, I will summarize experience accumulated in our group in developing software interfaces for young and very young children to sustain the development of their computational thinking and problem solving skills. Our deep engagement in this field results from two intertwined facts: first, we believe that computational thinking supported through educational programming is a valid contribution to the *general primary and secondary education for all* – not because we want to attract young people into university Computer Science programmes, but because it constitutes important part of so-called *skills for the 21st century learning*. Second, in our department, we are for more than 20 years involved in designing national curricula for Informatics (this is how the subject is called in our country), and during all that time we are trying to develop new pedagogies – and software interfaces, which would be *developmentally appropriate* and yet would mediate the potential of programming to education.

In 2008, we managed to establish Informatics as a mandatory subject in Slovakia for every primary student (after establishing it in 2005 at the lower secondary level and in 1985 at the upper secondary level). Since the beginning of that process, we have considered programming to be the substance of the subject. Therefore, we continuously apply so-called *design-based research* to better understand what are the fundamental cognitive operations of such programming for young and very young children, what are appropriate cognitive requirements of elementary educational programming, what the primary and secondary students consider difficult and easy when solving programming tasks.

In my talk, I will solely focus on our research and development at the kindergarten level (working with children aged 4 to 6, corresponding to ISCED 0) and at the primary level (working with pupils aged 6 to 10 or

11, corresponding to ISCED 1). In both contexts, our activities always include working with the teachers, an interesting challenge in itself.

We explored what are the appropriate methods for developing productive and constructive *learning software interfaces*1 for pre-primary children and recently applied these findings for developing new software environment to support their early computational thinking and problem solving skills. Inspired by our older Thomas the Clown2 environment, we developed new application for collaborative work of a group of children playing in front of an IWB and *planning future behaviours* for Thomas, who is solving various tasks in the world of a ZOO, a town, a garden or a farm yard. While doing so, children apply direct manipulations to control the character, they read and interpret symbolic plans for his future behaviours, they build such plans by themselves and fill in their missing parts. While doing so, they have to cope with *dynamically changing constraints*. The whole activity is organized in a constructivist way, providing children with opportunities to discover the problems and build their own solutions – in an environment with rich social interactions.

In the primary level, we exploit a unique occasion we have gained due to running Slovak version of a famous international contest Bebras. In 2011, we initiated a new category of the contest for the primary students, with rapidly growing participation by itself – 12.448 children solving 12 tasks in November 2013. Each year, two to four of the tasks are programming tasks. While creating these tasks and then analyzing the actual scores achieved by the primary students, we have exceptional opportunity and motivation for further research, as far as we want to understand cognitive requirements and difficulty of the tasks. In my talk, I will present some of our findings and conclusions resulting from that research and development, which we are trying to exploit in developing our new pedagogy of early educational programming.

Keywords: Learning interfaces development, computational thinking, educational programming, primary level, pre-primary level

Endnotes

1 We presented our findings at the IFIP TC3 working conference in 2012, see Moravcik, Kalas (2012).
2 See our IFIP TC3/WG3.1 and WG3.5 conference paper from 1993, see Blaho, Kalas (1993).

References

Blaho, A., Kalas, I. (1993). Thomash the Clown's circus: order in action in picture languages. In: *Informatics and changes in learning*. Amsterdam, North-Holland, pp. 79–86.

Moravcik, M., Kalas, I. (2012). Developing software for early childhood education. In: *Addressing educational challenges: the role of ICT*. Manchester MMU, pp. 1–12.

Biography

Ivan Kalas is a professor of Informatics Education. For more than 25 years, he concentrates on developing Informatics curricula for preschool, primary and secondary stages, developing textbooks and other teaching and learning materials for Informatics and ICT in education. Ivan is also interested in strategies for developing digital literacy of future and in-service teachers and enhancing learning processes through digital technologies.

Modelling and Measurement of Competencies in Computer Science Education

Johannes Magenheim[1],Sigrid Schubert[2], Niclas Schaper[3]
[1]Didactics of Informatics
University of Paderborn
jsm@upb.de
[2]Didactics of Informatics and E-Learning
University of Siegen
sigrid.schubert@uni-siegen.de
[3]Organizational Psychology
University of Paderborn
niclas.schaper@uni-paderborn.de

Abstract: As a result of the Bologna reform of educational systems in Europe the outcome orientation of learning processes, competence-oriented descriptions of the curricula and competence-oriented assessment procedures became standard also in Computer Science Education (CSE). The following keynote addresses important issues of shaping a CSE competence model especially in the area of informatics system comprehension and object-oriented modelling. Objectives and research methodology of the project MoKoM (Modelling and Measurement of Competences in CSE) are explained. Firstly, the CSE competence model was derived based on theoretical concepts and then secondly the model was empirically examined and refined using expert interviews. Furthermore, the paper depicts the development and examination of a competence measurement instrument, which was derived from the competence model. Therefore, the instrument was applied to a large sample of students at the gymnasium's upper class level. Subsequently, efforts to develop a competence level model, based on the retrieved empirical results and on expert ratings are presented. Finally, further demands on research on competence modelling in CSE will be outlined.

Keywords: Competence Modelling, Competence Measurement, Informatics System Application, Informatics System Comprehension, Informatics Modelling, Secondary Education

1 Motivation

The paradigm-shift to a learnercentred and an outcome-oriented view on learning processes has been influenced by discussions and ongoing research in different areas of education. Besides results of research according to constructivist and cognitive learning theories, the discussion on learning taxonomies and competencies were crucial for the design and evaluation of learning processes. The shaping of domain-specific competence models with regard to their internal structure and different competence levels basically served two main goals: They are used to define educational standards and thereby contribute to the development of curricula and they enable the measurement of competences and learning outcomes in diverse educational settings.

Especially as a result of the Bologna Process and the OECD Program for International Student Assessment (PISA) the development and assessment of educational standards became a high level objective in the educational system (Adams, 2002). In Europe standards for the major school subjects, like mathematics, natural sciences, and the first language were developed for different levels of education. In Computer Science Education (CSE) the development of educational standards is not as advanced as in those main school subjects. On an international level there are some standard-oriented curricula of CSE like the Model Curriculum for K-12 Computer Science published by the IEEE-CS and the ACM (Tucker et al., 2006) which has been revised later on by the Computer Science Teachers Association (CSTA, 2011) in 2011. In Germany the national CS-Society 'Gesellschaft für Informatik' (GI, 2008) published a proposal of informatics standards for lower secondary schools.

Nevertheless, these standards weren't based on an empirically proofed competence model for CSE. Therefore, in 2004 the research community of Didactics of Informatics in Germany started during the Dagstuhl-Seminar "Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics" (Magenheim, Schubert, 2004) a discussion about educational standards of CSE on a higher secondary school level. A result of this seminar was the comparison of the different approaches to educational standards in Mathematics and CSE. The results of the seminar revealed that further theoretical and empirical research was necessary to examine the opportunities of the measurement of educational standards of CSE and that respective research should be founded on a sound CSE competence model.

In an effort to develop such a competence model researchers in the fields of CSE and psychology started their research on this subject area. The project

MoKoM (Modelling and Measurement of Competences in CSE) funded by the German Research Foundation (DFG) from 2008 to 2014 developed a competence model and measured related competences of senior class students. The research project focused on two specific domains: informatics system comprehension and object oriented modelling. In the present paper we describe the objectives and research methodology of the project MoKoM on object-oriented modelling and system comprehension (section 2) along with the actual research results: an empirically refined competence model (ECM) (section 3), a derived measurement instrument (section 4), results of an empirical survey which has been conducted in Germany by applying the MoKoM-instruments (section 5) and finally first steps towards a competence level model (section 6). In conclusion we give an outlook on the necessity of further research in this subject area (section 7).

2 Objectives and Research Methodology

In alignment with the discussion on CSE standards in secondary education and in order to develop a CSE competence model the project MoKoM investigated the following main research questions:

- Which competencies are necessary for informatics system application, informatics system comprehension and informatics system modelling in upper secondary education?
- How can these competencies be related to a theoretical derived competence model (TCM)?
- How could the TCM be used to gain an empirically refined competence model (ECM)?
- Which test items are adequate to measure these competencies of the learners in CSE with a competence-based test-instrument?
- Is the test-instrument able to measure the described informatics competences in a valid and reliable way when applied to a large sample of senior students?
- Can such a test-instrument be used to distinguish between different competences of a group of students?
- Does the test instrument validate the assumed competence model resp. competence structure?
- How can this model be used for the grading of competencies and how can it be used to evaluate the learning outcomes of a specific CSE-learning setting?

In a first phase of the project, competence definitions, expert papers and CSE curricula were analysed. Thus, all competence dimensions were theoretically derived from international syllabi and curricula, e.g., the "Computing Curriculum 2001" of ACM and IEEE (Cross, Denning, 2001), the "Model Curriculum for K-12 Computer Science" of the ACM (CSTA, 2011) and a variety of other ACM, IEEE, IFIP, GI and CSTA (e.g. CS2013) publications. Additionally, expert papers like the Rational Unified Process for software development (IBM, 1998) were used to identify important competence components for system modelling. Based on the analysis of these resources and applying Weinert's definition of competence (Weinert, 2001), a first competence framework, containing cognitive and non-cognitive competences was developed.

But a restriction on exclusively theoretically derived competencies would risk that the reference of competencies to complex requirements in real situations is neglected or disregarded. Therefore, an additional step was necessary in order to determine competencies more reliably, that is, ensuring an empirical access to determinate the relevant competencies. Conducting expert interviews by applying the Critical Incident Technique represents an appropriate empirical approach to detect the relevant competencies in the subject domains 'system comprehension' and 'object-oriented modelling'.

The interviews of the 30 experts (experts in the domain of didactics of informatics, computer scientists and expert informatics teachers) were based on a structured questionnaire manual and included twelve hypothetic scenarios (see figure 1) concerning application, testing, modifying and developing of informatics systems. The expert interviews were transcribed in full and analysed by means of qualitative content analysis according to Mayring (Mayring, 2003). The requirements of intercoder reliability were also considered during this empirical phase of analysis and were sufficiently achieved.

Scenario: “*You are asked by a colleague to test his software, which was developed to solve configuration problems, e.g. to set up a new car or a new computer.”*
Question 1: *“What is your strategy of testing to solve this problem? Which aspects do you have to bear in mind?”*
Question 2: *“Which cognitive skills are required for such a software exploration?”*
Question 2.1: *“Which informatics views are important for this task?”*
Question 2.2: *“Which complexity would you assign to this task?”*
Question 3: *“Are there any attitudes or social communicative and cooperative skills which are necessary to accomplish this?”*
Question 4: *“Which differences of competence levels would you expect between novices and experts?”*
Question 5: *“Could you imagine a potential pupil‘s procedure to solve this problem?”*
Question 6: *“Which obstacles would pupils have to cope with?”*

Figure 1: Interview scenario

The results of the qualitative content analysis have to be structured according to the dimensions of the competence model. Relations between the competence components and meaning units in the interview have to be found and described. An example shows the answer about social-communicative skills: “There is a serious contradiction between the competence of problem solving and the social-communicative competencies.” This means it is necessary to supervise the development of social-communicative competencies, since they are not fostered as a side effect of informatics problem solving. Another example shows the answer about empathy, change of perspectives and roles: “When we test software of others, we have to learn to criticize in a fair and sensitive way.” The task of systematic testing gives the opportunity to gain non-cognitive competencies on a higher level when the learner presents his results to other learners, e.g. the explanation of use cases, the presentation of test results including the visualization of large data collections.

3 Competence Model on Informatics System Comprehension and Object-Oriented Modeling

The described content analytic procedure led us to an empirically refined competence model (see figure 2). But the described empirical procedure to complement the theoretical model is nevertheless restricted. One methodological restriction implies, that the relevant competence requirements are closely linked to the used scenarios. So it is important that the scenarios contain at least typical and representative tasks and problems to be solved. This was ensured by the representative ratings of the experts. Furthermore, the actions described by the informatics experts might not necessarily mirror their actual behaviour in those scenarios because they could describe idealized actions to solve the problems in the scenarios. On this issue, the different orientations of expertise of the interviewees serve as a corrective to some extent. The deployment of the qualitative content analysis took place adhering to comprehensible, methodical rules and principles. Nevertheless, qualitative analyses include inevitably interpretative processes, which might restrict the objectivity, reliability and validity of the described analyses.

As a result of these research efforts in the MoKoM-project a theoretically derived and empirically refined competence model was developed.

The empirical refined competence model contains four cognitive dimensions K1 'System application', K2 'System comprehension', K3 'System development' and K4 'Dealing with system complexity'. Additionally a non-cognitive dimension K5 covers 'Non-cognitive skills'.

A condensed version is depicted in figure 2. The extended version with all sub-categories was published in 2013 (Linck et. al., 2013).

While these categories of the competence model represent only the structure of the model in terms of components and hierarchy the derived items were contextualized and meet the requirements of competence definitions regarding a person´s ability to perform observable action.

K1 System application

K1.1 Structuring of application field
K1.2 System exploration
K1.3 System selection
K1.4 Use of media to foster system application
K1.5 Transfer to new application fields

K2 System comprehension

K2.1 System requirements
K2.2 Systematic tests
K2.3 System exploration
K2.4 Evaluation of software quality
K2.5 Architecture & organization
K2.6 Algorithms & data structures
K2.7 Informatics' Views

K3 System development

K3.1 Software development process models
K3.2 Business Modeling
K3.3 Requirements
K3.4 Analysis
K3.5 Design
K3.6 Implementation
K3.7 Test
K3.8 Iterative development

K4 Dealing with system complexity

K4.1 Measures of complexity: Time & Space
K4.2 Number of components
K4.3 Level of networkedness
K4.4 Stand-alone vs. distributed systems
K4.5 Level of human-computer interaction
K4.6 Combinatorial complexity

K5 Non-cognitive skills

K5.1 Attitudes
K5.2 Social-communicative skills
K5.3 Motivational and volitional skills

Figure 2: Competence Model

4 Development of a Competence Measurement Instrument

During further methodical steps, as described in the following sections, test items and an empirical test instrument was developed on the basis of the refined competence model. The empirical test instrument was applied on a representative sample of students in secondary schools in Germany, mainly from Bavaria and North Rhine Westphalia. The results of this survey not only provides an insight into the competencies and abilities of students in CSE at secondary schools but enables us to development a competence level model for the needs of grading competences.

4.1 Principles of Competence Measurement

Based on the empirical competence model the test instrument was developed following the principles of Situational Judgment Tests (SJT; Weekly, Ployhart, 2006). This means that we created knowledge application scenarios which specifically addressed the specific competence requirements of each model facet that had to be operationalized. We also took into consideration experiences of how to construct competence measurement items gained in large scale studies like TIMMS, PISA and DESI.

Based on detailed competence descriptions, tasks for every single competence item were created. After this, the answering format was created. In the majority, this included closed answering formats like multiple choice or classification items. But also tasks with open questions that required short sentences or the statement of keywords as answers were used. The answering format was chosen and created in accordance with the cognitive requirements and levels (according to the cognitive dimension of the Anderson, Krathwohl, 2001 taxonomy) that had to be addressed. We also used a complex item format which included multiple items resp. tasks that were integrated in one complex application scenario. So, we were able to address different competence facets in one task context and by this economize the measurement. To allow an objective and reliable appraisal of the answers (especially when evaluating open item format), a comprehensive grading manual was created alongside the test items. This contained different sample solutions as well as approaches to grade answers.

The test instrument was examined and optimized by conducting a preliminary test with students from local secondary schools. In addition, student computer science teachers from didactical courses at the universities of Paderborn and Siegen were asked to review the instrument. The main issues found during this pre-test were ambiguous wording and oversight mistakes on the one hand and difficulties of applicability of the tasks on the other. Rewriting or extending the context of the tasks could easily fix the latter.

4.2 Design of Test Items of CSE

The empirically refined competence model allows the defining of competence profiles, which are the basis for the model for the instruments of competence measurement (see figure 3).

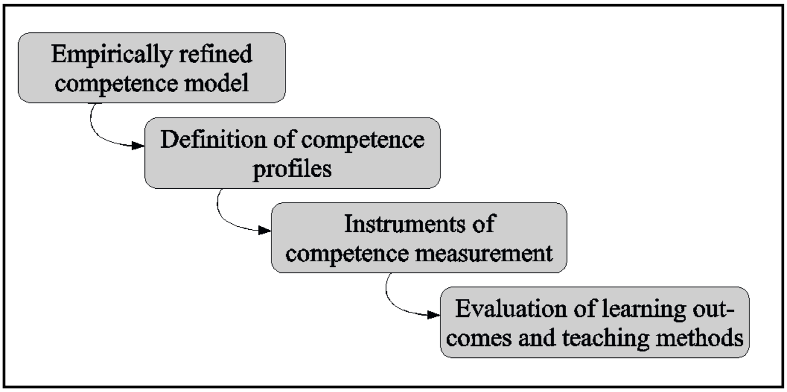


Figure 3: Impact of competence profiles on learning

To illustrate the procedure of defining competence profiles on informatics system comprehension, we will start with an example: (1) A competence component of the empirically refined competence model is chosen, e.g. "Errors as Learning Opportunities". (2) All main expert statements, which are related with this competence component, are collected from the spreadsheets. (3) Step 3 is to select citations of the collection of expert statements, which have the most meaningful expressions. Such citations of expert statements will be called "anchored examples". In this case two anchored examples are related to this component:

I. *"Most important is the ability not to give up after the first syntax error, but to learn from them, and to determine error messages. I want to deliver a completed product, which actually does, what it should do."*

II. *"You have to intervene in this case and reflect once again, and in the very moment, when it happens, say: This error, you will never do it again."*

(4) A first competence profile definition of "Errors as Learning Opportunities" is based on the content of both statements. (I) implies that students should require the competence to identify errors and (II) implies that learners should detect and avoid errors:

> *"The learners are able to determine, to assess and to examine system-based errors. This acquired knowledge will be applied to error avoidance and improvement of tests."*

(5) Is to improve this definition. Therefore, keywords, referring to the cognitive processes in (Anderson, Krathwohl, 2001), are used. These keywords are called operators. An operator is a work instruction, which refers to the content and to the methods to solve a given task. In the competence profiles the operator ensures that misinterpretations of the requirements towards the learners are reduced. In the competence profile definition above is "apply" an operator. In contrast, "determine, assess and examine" have to be discussed. The challenge is to find synonyms or similar expressions and express the meaning of the first competence profile definition. A refined definition of the competence profile follows:

> *"The learners are able to identify, to differentiate, and to judge system-based errors. This acquired knowledge is applied to error avoidance and improvement of tests."*

Four operators "to identify, to differentiate, to judge, and to apply" are used in the definition of competence profiles. These operators support our aim, which is to assure the standardisation. After defining competence profiles for each component of the empirically refined competence model, test items can be developed. These test items measure the individual performance of a learner related to different components of the competence model in classroom practice. All cognitive and non-cognitive process dimensions, which are defined in a competence profile, have to be tested by such items. We developed and improved such test items with CSE teachers. This is an example of a successful test item: "You got the homework to write an algorithm, which sums up all numbers from 0 to n. Your friend already gave you his ideas noted in a pseudo code (see figure 4). Decide which of the two algorithms is better regarding the running time."

Enter: n
Set sum = 0
Set i = 0
Repeat from 0 to n
    Set sum = sum + i
    Set i = i + 1
Return sum
	Enter: n
	Set sum = 0
	If n odd-numbered, then
	    Set sum = sum + n
	    Set n = n - 1
	Set sum = (n / 2) * (n + 1)
	Return sum

Figure 4: Algorithms in pseudo code

5 Applying the Measurement Instrument

5.1 The Population of the Test

Due to the large amount of items, the test instrument was not applicable in a classroom setting with timeslots of usually 90 minutes. Furthermore, students' attention to the test instrument shouldn't be required more than this time span. In order to adapt the instrument to a 90-minute timeslot, the items were divided into six blocks. Then six booklets were compiled from three item blocks each. Together with an additional questionnaire on attitudinal, motivational and volitional competences (representing facets of the dimension "non-cognitive skills" resp. K5), the whole test can be accomplished within 90 minutes. The application of such an arrangement, called "matrix design", is possible due to the application of the 'Item Response Theory' to analyse the test results. Though not all students answer every task due to not having them in their booklet and thus produce a lot of "missing values" in the final data, the IRT allows the calculated estimation of student abilities in combination with the overall item difficulty. This method provides coherent results even if the students worked on different subsets of items (Hartig, 2008), (Rost, 2004). The booklets were distributed to more than 800 computer science students in German upper secondary schools. The analysis of the returned data was done with ACER ConQuest, applying a 1PL partial credit model to estimate the item difficulties (Wu, Adams, Wilson, Haldane, 2007).

The booklets were originally distributed in 26 classes with 522 students in North Rhine Westphalia. Additionally 6 classes from Berlin, Hessen and Lower Saxony with a total of 82 students also participated. In Bavaria 244 students from 11 different classes (6 classes of grade 10 and 3 of grade 11) took part in the test. According to the curriculum, the current learning content of most of the responding students was focused on object-orientation, the use of standards software like databases or spread-sheets and simplest concepts of programming. The test was conducted as a pencil-and-paper-test. The print-versions of the booklets were sent to teachers who volunteered to deliver them to their classes. To prevent the students from cheating, each teacher received two to three different booklets to distribute them among the class. From more than 800 tests we sent out we received back 583 completed and evaluable booklets. The investigated sample consists of 86 % male and 14 % female students with an average age of 17.5 years. 17 % of them had an immigrant background. Their self-assessed proficiency in computer science on a scale from 1 to 6 averaged at 2.65 points. They had participated in computer science classes for a mean of 3.5 years. Only 3.3 % had dropped the subject in the interim.

5.2 Analysing the Test Data – Test of Model Fit

The gathered data were analysed according to the Multidimensional Item Response Theory (MIRT). The main goal was to examine the dimensional validity resp. structure of the competence model and the reliability measurement instrument. To do so, several different IRT models were used to analyze the empirical data and the results were compared to assess the best fitting model.

IRT models assume that personality traits cannot be measured directly and test results can only be interpreted as an indicator for the existence and intensity of such a trait. Therefore, IRT models differentiate between latent variables, that can't be measured directly, but influence the response to a test item, and manifest responses that are assumed to be the observable manifestations of the latent traits. Thus, the ability of the test subject can be inferred from the responses. Furthermore, it is assumed that any subject has a certain probability to answer any item right or wrong. The difficulty of the item and the ability level of the subject determine this probability.

IRT has several advantages for the assessment of competences. For once, the estimation of the item difficulties and student abilities does not require for every participant to work on every task of the test instrument. This allows to use a matrix design with different booklets that only represent a part (about three-fourths) of the item resp. task pool of the competence test. Furthermore, the estimated parameters can be interpreted on the same scale and easily related to each other.

Since competence structures are complex constructs, they often result in multidimensional competence models. In our case this applies to the cognitive dimensions K1 to K4 with the additional non-cognitive dimension K5. The latter was excluded from the IRT analyses because the data for this dimension was raised by a questionnaire. To evaluate the dimensionality of the empirical data, multidimensional IRT models can be utilized, which assume that multiple latent variables (one per dimension) cause the responses to a test. Furthermore, MIRT allows for the comparison of different models, by analysing the conformity of the theorized model to the empirical data.

We also had to choose between a speed test and a power test variant to analyze the data (this has consequences concerning the handling of missing values). There are reasonable arguments for both variants. Therefore, we analyzed both. Since the results of both variants are very similar though, this article will concentrate on the results of the speed option. To calculate the MIRT analysis we used ACER ConQuest Version 2.

To evaluate the structure of the competence model, we analyzed four different IRT models with one to four assumed dimensions respectively. Since the test items were crafted with the intent to test for one specific competence, a between-item multidimensionality model was used in all cases. Because not all items could be coded as dichotomous responses, the partial credit model was applied to analyze dichotomous and polytomous data alike. Starting with the one-dimensional model, for which it was assumed that all items loaded on the same latent trait, every model added one additional dimension in accordance with the assumptions concerning the structure resp. dimensionality of the competence model described above. The analyses results concerning the IRT models with different competence dimensions can be seen in table 1.

Table 1: Final deviance, estimated parameters and reliability for evaluated models

Model	**Final Deviance**	**Estimated Parameters**	**Reliability for dimension 1 to 4 (if available)**
1-Dim	87379.09538	316	0.872 (K1,K2,K3,K4)
2-Dim	86695.99173	319	0.831 (K1) / 0.831 (K3)
3-Dim	86403.83657	323	0.749 (K1) / 0.806 (K2,K4) / 0.812 (K3)
4-Dim	85891.85717	328	0.779 (K1) / 0.763 (K2) / 0.861 (K3) / 0.759 (K4)

To compare the models, the final deviance – an indicator of how well the empirical data fits the IRT model – and the number of estimated parameters reported by Con-Quest can be used. Usually, both parameters should be as low as possible. If it is not possible to choose the better model by comparing the values alone (because one value is lower, while the other one is bigger than the parameters of the second model), a Chi-Square-Test can be calculated, using the difference in deviance and the difference in estimated parameters as the degrees-of-freedom. If the result is significant, the model with the smaller deviance parameter is selected. Otherwise the model with the lower amount of estimated parameters is deemed the better one. The parameters for each evaluated model can be seen in table 1.

Table 2: Chi-Square statistics for model comparisons with difference in deviance and difference in estimated parameters as degrees of freedom

	2-Dim	**3-Dim**	**4-Dim**
1-Dim	$\chi(3)2=683.1$, $p<.001$	$\chi(7)2=975.26$, $p<.001$	$\chi(12)2=1487.24$, $p<.001$
2-Dim		$\chi(4)2=292.15$, $p<.001$	$\chi(9)2=804.13$, $p<.001$
3-Dim			$\chi(5)2=511.98$, $p<.001$

Since with increasing dimensions the deviance decreases and the number of parameters increases, a Chi-square-test was calculated for every combination of models (see table 2). In every case the result was statistically significant and since the models with a higher number of dimensions have a lower deviance, it can be assumed that they better match the empirical data than the models with fewer dimensions. Thus, the four-dimensional model has the best model fit overall.

5.3 Analyzing the Test Data – Item Fit and Reliability

ConQuest calculates the EAP/PV reliability for each dimension, which can be compared to Cronbach's Alpha. Table 1 shows the reliability for all dimensions in each model. All values exceed 0.7 and can be considered acceptable.

To further evaluate the models, the item fit for individual items can be examined. The fit compares the predicted probabilities for each item within the model with the observed responses. To do this, ConQuest calculates the weighted mean squares (wMNSQ), which are expected to be 1 for perfectly fitting items. The wMNSQ for a good fitting item should fall between 0.8 and 1.2, and the corresponding t-values should not be greater than 1.96. Furthermore, the discrimination parameter shows how an item correlates to the overall test results. With the discrimination close to 0, an item may not be useful to differentiate between students with high levels of a trait and those with low levels. Values between 0.4 and 0.7 are considered good while values above 0.3 can be considered as acceptable.

The data for all models (see table 3) showed a good item fit overall, but the percentage of unfit items increased for models with more dimensions, from below 1 % (2 items out of 292) for the one-dimensional to 4.7 % (14 items) for the four-dimensional model. In addition, the number of items that might have a bad fit according to the t-values increased from 27 to 37 items. Unfortunately the discrimination parameters are not very good for a large part of the items. Just 22.6 % (66 items) are above the 0.4 threshold and even if we adjust the point at which an item is considered to have a too small discrimination to 0.3, roughly 43.8 % (128 items) remain under that line. Only one item had a negative discrimination, which was close to 0. The high ratio of low discrimination items necessitates a throughout examination of the items and how they fit to their corresponding dimension in further steps.

Table 3: Range of mean squares, t-values and discrimination values for all models

Model	**wMNSO**	***t***	**Discrimination**
1-Dim	0.86≤wMNSQ≤1.3	-2,9≤t≤4.4	-0.04≤Disc. ≤0.58
2-Dim	0.77≤wMNSQ≤1.42	-4.1≤t≤5.2	-0.04≤Disc. ≤0.58
3-Dim	0.76≤wMNSQ≤1.42	-4,2≤t≤5.2	-0.04≤Disc. ≤0.58
4-Dim	0.65≤wMNSQ≤1.42	-5.7≤t≤5.3	-0.04≤Disc. ≤0.58

5.4 Analyzing the Test Data – Difficulty Parameters and Latent Abilities

Main goal of IRT analysis is the estimation of two parameters: the item difficulty, that denotes the probability of answering an item correct given a certain level of the measured construct, and person parameters, that assess the level of the latent trait for individual students. One advantage of IRT analysis is that both estimates can be arranged on the same scale and easily compared. The item-person-map for each model visualizes the item difficulties on the right, by ordering them from more difficult (top) to less difficult (bottom), and the latent trait levels on the left (grouping persons with the same values together). Ideally, the item difficulties should be well dispersed around the mean, having the most items in the medium difficulty range, but also providing items with high and low difficulties. Additionally, the latent traits are separated by dimension. Figure 5 shows the maps for the one- and four-dimensional models. As can be seen, the item difficulties are well distributed along the axis, though there are some aspects that have to be noticed and commented.

First, there are some outliers in the upper part of each map. This indicates, that some items are way to difficult for the targeted student groups, since no person was estimated to have a high enough proficiency to solve these items with an adequate probability.

Second, the latent traits in the different dimensional solutions are somewhat uneven dispersed. While the one-dimensional model indicates, that the overall difficulty of the test matches the ability of the population, the four-dimensional model reveals, that only the third dimension can be considered well matched. Dimension 1 and 4 lack items in the upper difficulty range, while dimension 2 necessitates less difficult items to adequately assess its competences.

Figure 5: Overview of the estimated item parameters for the one- and four-dimensional model

6 Modelling of Competence Levels

In further evaluation steps of our test instrument we want to grade the measured competences of modelling and comprehending informatics systems which can be interpreted as competence levels of the developed model. To create a competence level model you have to choose between different approaches (Hartig, 2004). We decided to use an inductive approach which is based on systematic post hoc analyses of task contents and requirements. To apply this approach, different steps have to be conducted to identify and generate the desired competence levels measured by a certain competence test:

First, you have to identify and define task features that determine the difficulty when coping with the task requirements and contents. Secondly, you have to determine and describe the different grades or levels of difficulty concerning each difficulty feature. In a third step you have to determine how the different difficulty features and grades are represented in each test item. Therefore you have to conduct an expert rating at which the experts examine and rate each item if specific difficulty features and levels are given or required when coping with the item. In a fourth step the expert ratings of the difficulty features of each test item are related to the empirically determined difficulty parameters (when the test is applied to a large sample of students). This is conducted by regression analyses to test if the assumed difficulty features and grades are really determining the empirically determined difficulty of the items. Only those difficulty features and grades that prove to be significant predictors of the empirical difficulty are kept in the further process of defining the competence levels. In a fifth step the items are ordered concerning their empirically determined difficulty and in an adjunct table for each item it is systematically determined and described if a difficulty feature is realized in the requirements and at which difficulty grade resp. level. This table is used in a sixth step to determine and define thresholds of competence levels. This is usually the case, when new difficulty features or grades appear at a certain type of items. After you have determined such thresholds and the number of different competence levels you have to describe each level in a seventh step. Therefore, you have to take reference to the requirements of the items that belong to a specific competence level. These requirements are especially derived from the difficulty features and grades, which characterize these group of items typically. In a last step you have to classify the persons of your sample according to the competence levels to determine how the sample is distributed over the competence levels.

In the following we describe the analyses we have conducted so far to generate a competence level on the basis of our test instrument and study sample.

6.1 Identification and Description of Difficulty Relevant Features of the Competency Test Items

To identify and describe difficulty relevant features of the competency test we first defined difficulty relevant features of the competence test items. We derived those features from the literature concerning difficulty relevant features of competence tests in general (e.g. Schaper et al., 2008). Furthermore we analysed the items concerning informatics specific difficulty facets and tried to define and grade them analogue to the more general features. On this basis altogether thirteen features were identified and defined: addressed knowledge taxonomy level (KTL), cognitive process dimensions (CP), cognitive combination- and differentiation capacities (CCD), cognitive strain (CS), scope of tasks (necessary materials, reading effort and understanding) (ST), inner- vs. outer computational task formulation, aspects of demands of computer science (IOC), number of components, level of connectedness (NC), stand-alone vs. distributed system (SDS), level of human-computer-interaction (HCI), (mathematical) combinatorial complexity (CC), level of the necessary understanding of systems of computer science (LUS), level of the necessary modelling competence of computer science (LMC). Because of extent restrictions only two of these features are described in more detail.

6.2 Cognitive Process Dimensions

Concerning this difficulty determining feature we analysed the structure of the cognitive process dimensions of the revised taxonomy for learning, teaching, and assessing by Anderson and Krathwohl (Anderson, Krathwohl, 2001). We assumed that these addressed the following process categories: 1. Remember, 2. Understand, 3. Apply, 4. Analyze, 5. Evaluate and 6. Create, which were also used to differentiate between different cognitive requirement levels of our test items. So we defined this difficulty-relevant feature with the following six feature levels:

- CP1: The successful solution of the task requires a memory performance. The students recall relevant knowledge contents from their memory.
- CP2: The successful solution of the task requires a comprehension performance. The students understand terms, concepts, and procedures of computer science and can explain, present and give examples for them.

- CP3: The successful solution of the task requires an application performance. The students are able to implement known contents, concepts and procedures within a familiar as well as an unfamiliar context.
- CP4: The successful solution of the task requires an analysis. The students are able to differentiate between relevant and irrelevant contents, concepts and procedures. They choose the suitable procedures from a pool of available procedures.

6.3 Cognitive Combination and Differentiation Capacities

We assumed that this feature addresses different forms of knowledge utilization like *Reproduction, Application, Networked application*, and that these requirements differentiate between different levels of difficulty concerning our test items. So we derived the third difficulty-relevant feature with the following three feature levels:

- CCD1: Reproduction of computer science knowledge and application of single, elementary terms, concepts and procedures of computer science in close contexts (no cognitive combination capacities required).
- CCD2: Application of single terms, concepts and procedures of computer science in bigger contexts, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made.
- CCD3: Networked Application of terms, concepts and procedures of computer science in different, especially bigger scenarios, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made (multiple challenging cognitive combination capacities required).

6.4 Expert Rating of the Difficulty Determining Task Resp. Item Features

In a second step we used the described features of task difficulty to rate the difficulties of the items of our competence test. Therefore experts in computer science education were asked to rate each item of the competence test with reference to the thirteen difficulty features. To conduct the expert rating a rating scheme and instructions were formulated. Furthermore, the measurement instrument was split into four parts of roughly equal size to keep the amount of ratings at an acceptable extent. Each of the four instrument parts – including solutions for the items – was presented to two selected experts in the field of didactics of informatics, along with an explanation of each feature and its rating levels. The experts were asked to answer each item on their own, compare the solution with the given sample solution and then rate the item for each of the features. In addition, the experts had to give a subjective rating of the item difficulty on a scale from one to ten.

The resulting two ratings for each item were compared and treated in three ways: 1. exact matches between the two ratings of a certain feature per item were accepted and not further treated; 2. items with small rating differences (if the ratings only differ one point or grade from each other) were discussed within the project group to decide upon a final rating; 3. items with big rating differences (if the ratings deviate two points and more from each other); these cases were presented to two further experts that had to rate these features for a certain item again while considering the ratings of the two preliminary experts; again, resulting differences of these experts were discussed in the project group to decide upon a final rating. The expert group was composed of seven researchers with background in computer science, computer science education and psychology.

The rating process resulted in a classification of 74 items concerning each of the described difficulty determining features. The rating levels for each feature were coded as ordinal dimensions, e.g. coding KTL1 as 1 and KTL2 as 2. For every feature the "not relevant" rating was coded as 0. This way, we ended up with 13 nominal variables with n+1 categories for a feature with n levels. For almost all features it was reasonable to assume a ranking of the levels in the order they are described above. The assumption is that a higher level correlates with a higher item difficulty. As this assumption does not necessarily have to be true, the order was examined by the analysis of the rating data. This was done using descriptive and explorative methods to determine the relevant features that influence the item difficulty.

In the following only some of the results of the expert rating are described and summarized: The number of ratings of features related to cognitive demands like KTL, CP and CCD are mostly distributed at the medium rating levels. This makes sense and was intended when creating the tasks: The instrument should provide mainly items with a medium difficulty, since it can be expected for most subjects that they are able to solve items of medium difficulty. Therefore, the test instrument has to differentiate the best at this difficulty resp. competence level. In the upper difficulty range fewer items are required, since this would be enough to show the expertise of the more competent students.

The expert ratings though, show a tendency to lower rating levels. For example the cognitive process dimension "remember" was assigned more times (8) than the dimensions "evaluate" and "create" which were combined at one grading level (4 times). The same can be observed for the two features CS and CCD. For the features CS and ST the predominance of the lower rating levels is a result of the test design. To create an applicable instrument, the tasks need to adhere to certain constrains and thus the most items require only few processing steps and a minimal amount of additional materials. The overall difficulty of the test instrument was subjectively rated by the experts with a mean of 4.2 on a ten-point scale.

6.5 Regression Analysis and Further Analysis Steps

To determine which features have the most influence on the item difficulty, the expert ratings were related to the empirical difficulty estimates that were calculated by means of the Item Response Theory (IRT) (Schaper et al., 2008). The relations between the difficulty determining features rated by experts and the empirically determined item difficulty are examined by regression analyses. These analyses are not computed and evaluated at the moment though and therefore cannot be reported here at the moment. Also, to model the competence levels for our test instrument and model we still have to conduct the further analyses steps described at the beginning of this section. This will therefore be reported at another place later on.

7 Conclusion and Further Work

In this article we outlined essential research questions and the corresponding research methodology of the project MoKoM concerning upper secondary students' competences. As a first main result we developed a theoretically grounded and empirically refined competence structure model in the subject area of informatics system comprehension and object-oriented modelling. Based on this model an empirical test instrument was developed and an empirical survey conducted. By applying IRT evaluation methodology to construct the test-instrument and to assess the data, gained from the survey with 583 upper secondary students in Germany. We finally took first steps to develop a competence level model that also considers the results of an expert rating on the difficulty levels of the test-items. Thus, we answered several of the research questions, which have been raised at the beginning of this article.

We also proved that our test instrument was able to identify competence profiles of learners and to indicate the difference of competences between members of a learning group (Neugebauer et al., 2014). We also conducted a survey in a joint project with the German University of Distance Learning (FernUniversität Hagen) on students in an introductory course of object-oriented software engineering. We were able to show, that the instrument could even be applied at undergraduate university level. The students underwent the test at the beginning and the end of the CS-course and we were able to analyse the students' increase in subject-related CS-competences during the course (Hering et al., 2014). Further research of the project will concentrate on the application of the MoKoM test-instruments to evaluate the learning outcomes of specific learning design settings in CSE. In general the MoKoM competence model and the related test instrument should be used to contribute to the theoretically founded and empirically based development of standards in CSE. Furthermore, the application of the test-instrument on an enhanced sample of students could provide an overview on students competences in CSE and reveal a possible gap between these competences students' really own and the expected learning outcomes according to the curricula of CSE.

References

Adams, R. J.: *Scaling PISA cognitive data. PISA Programme for International Student Assessment (PISA 2002)*. PISA 2000 Technical Report. pp. 99–108. OECD Publishing, Paris (2002).

Anderson, L. W., Krathwohl, D. R. (2001): *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Addison Wesley Longman.

Cross, J., Denning, P. (2001): *Computing Curriculum 2001, The Joint Curriculum Task Force IEEE-CS/ ACM Report*. 2001, www.computer.org/education/cc2001.

CS2013 Steering Committee, The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society (2012): *Computer Science Curricula 2013*. http://ai.stanford.edu/users/sahami/CS2013/

German Informatics Society (GI) (2008): *Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I* (in German). Retrieved from: http://www.informatikstandards.de/

Hartig, J. (2004): Skalierung und Definition von Kompetenzniveaus. In: B. Beck, E. Klieme (Eds.): *Sprachliche Kompetenzen. Konzepte und Messung*. DESI-Studie. Weinheim: Beltz.

Hartig, J. et al. (2008): Methodische Grundlagen der Messung und Erklärung sprachlicher Kompetenzen. In: DESI-Konsortium (ed.) *Unterricht und Kompetenzerwerb in Deutsch und Englisch. Ergebnisse der DESI-Studie*. pp. 34–54. Beltz, Weinheim, Basel.

Hering, W., Huppertz, H., Krämer, B., Magenheim, J., Neugebauer J., Schreier, S. (2014): On Benefits of Interactive Online Learning in Higher Distance Education. Case Study in the Context of Programming Education. In: *Procceedings of the IARIA EL&ML 2014*, Barcelona.

IBM (ed.) (1998): *Rational Unified Process, Best Practices for Software Development Teams*. Retrieved from: http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Lehner, L., Magenheim, J., Nelles, W., Rhode, T., Schaper, N., Schubert S., Stechert, P. (2010): Informatics Systems and Modelling – Case Studies of Expert Interviews. In: Reynolds, N.; Turcsányi-Szabó, M.: *Key Competencies in the Knowledge Society*. Springer, Boston, pp. 222–233.

Linck, B., Ohrndorf, L., Schubert, S., Magenheim, J., Nelles, W., Neugebauer, J., Schaper, N., Stechert, P. (2013): Competence model for informatics modelling and system comprehension. In: *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*. Berlin, pp. 85–93, DOI: 10.1109/EduCon.2013.6530090

Magenheim, J., Schubert, S. (eds.) (2004): *Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics*. GI-Dagstuhl-Seminar, 19th–24th September 2004, Volume 1, Seminar-Edition "Lecture Notes in Informatics", German Informatics society (GI).

Magenheim, J., Nelles, W., Rhode, T., Schaper, N., Schubert, S., Stechert, P. (2010): Competencies for Informatics Systems and Modeling. Results of Qualitative Content Analysis of Expert Interviews. In: *Proceedings of the Engineering Education Conference 2010, IEEE Computer Society*, pp. 513–521, DOI: 10.1109/EDUCON.2010.5492535

Mayring, P. (2003): *Qualitative Inhaltsanalyse*. Beltz, Weinheim, 2003.

Nelles, W., Schaper, N.: Developing a Competence Model for the domains of Informatics Modeling, Informatics System Comprehension and Application. In: *Proceedings of Earli Jure 2012 "A Learning Odyssey: Exploring new Horizons in Learning and Instruction"*, 23–27 July 2012, Regensburg, in press, URL: http://www.earli-jure2012.org/

Neugebauer, J., Hubwieser, P., Magenheim, J., Ohrndorf, L., Schaper, N., Schubert, S. (2014): *Measuring Student Competences in German Upper Secondary Computer Science Education*; accepted Full paper to be published in the Proceedings of the ISSEP 2014, Istanbul.

Rost, J. (2004): *Lehrbuch Testtheorie–Testkonstruktion*. Huber, Bern, Göttingen, Toronto, Seattle (2004).

Schaper, N., Ulbricht, T., Hochholdinger, S. (2008): Zusammenhang von Anforderungsmerkmalen und Schwierigkeitsparametern der MT21 Items. In: Bloemeke, S., Kaiser, G., Lehmann, R. (Eds.): *Professionelle Kompetenz angehender Lehrerinnen und Lehrer*. Muenster: Waxmann.

Tucker, A. (ed), Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2006): *A Model Curriculum for K-12 Computer Science. Final Report of the ACM K-12 Task Force Curriculum Committee*. (2nd Edition). Association for Computing Machinery (ACM).

The CSTA Standards Task Force. (2011): *CSTA K-12 ComputerScience Standards*. (D. Seehorn, S. Carey, B. Fuschetto, I. Lee, D. Moix, D. O'Grady-Cunniff, et al., Eds.) (pp. 1–73). http://csta.acm.org/Curriculum/sub/K12Standards.html

Weekly, J.A., Ployhart, R.E. (2006): *Situational Judgement Tests. Theory, Measurement and Application*. Mahwah, N.J.: Lawrence Erlbaum.

Weinert, F. E.: Concept of Competence: A Conceptual Clarification. In: Rychen, D., Salganik, L. (eds.): *Defining and Selecting Key Competences*. Seattle, (2001).

Wu, M. L., Adams, R. J., Wilson, M. R., Haldane, S. A. (2007): *ACER ConQuest Version 2.0: Generalised item response modelling software*. Melbourne, Australia: ACER Press.

Biographies

Johannes Magenheim is a Professor for Computer Science Education at the Institute of Computer Science at the University of Paderborn. His primary areas of research and teaching are CSE, E-Learning, CSCL. He is also member of different working groups of the German Informatics Society (GI) and member of IFIP WGs 3.1 and 3.3.

Niclas Schaper is a Professor for Work and Organisational Psychology at the University of Paderborn. His primary areas of research are job analysis, occupational training, approaches of competency modeling, methods of competency measurement, E-Learning, and teaching in higher education. He is a member of the German Society of Psychology and vice president of the German Society of Teaching in Higher Education.

Since 1979 **Sigrid Schubert** has taught informatics in secondary, vocational and higher education. She has been professor of "Didactics of Informatics and E-Learning" (Universities Siegen and Dortmund, Germany) since 1998. Her research interests are Informatics Teacher Education and E-Learning. She is Fellow of the German Society for Informatics (GI), member of the Technical Committee 3 "Education" of the International Federation for Information Processing (IFIP) and chair of the IFIP Working Group 3.1 "Informatics and digital technologies in School Education".

Full Papers

Teacher Perceptions of Key Competencies in ICT

Jan Barnes[1], **Steve Kennewell**[2]
[1]University of Wales Trinity St David, UK
jan.barnes@sm.uwtsd.ac.uk
[2]Cardiff Metropolitan University, UK
steve.kennewell@smu.ac.uk

Abstract: Regardless of what is intended by government curriculum specifications and advised by educational experts, the competencies taught and learned in and out of classrooms can vary considerably. In this paper, we discuss in particular how we can investigate the perceptions that individual teachers have of competencies in ICT, and how these and other factors may influence students' learning. We report case study research which identifies contradictions within the teaching of ICT competencies as an activity system, highlighting issues concerning the object of the curriculum, the roles of the participants and the school cultures. In a particular case, contradictions in the learning objectives between higher order skills and the use of application tools have been resolved by a change in the teacher's perceptions which have not led to changes in other aspects of the activity system. We look forward to further investigation of the effects of these contradictions in other case studies and on forthcoming curriculum change.

Keywords: ICT competencies, Teacher perceptions, Activity Theory, Contradictions

1 Introduction

In England and Wales, the introduction of the National Curriculum in 1989 required all students between the ages of 5 and 16 to follow the same subject-based curriculum which was to be designed by government based committees and approved by the secretary of State.

Prior to this, there was no established curriculum for Informatics up to age 14; indeed there was little formal basis for the learning of Information and Communication Technology (ICT). The ICT curriculum of 1990 was strongly influenced by the earlier document from the government inspectorate for Education (DES, 1985). The curriculum has been revised on a number of occasions, and since the devolution of Wales in 1998, there have been differences between the curriculum in England and Wales, with the latest revision in Wales coming into being in 2008 (DCELLS, 2009). However, what is taught and learned in different schools, and different classrooms in the same school, may vary considerably from what is intended by authors of curriculum specifications. It is the purpose of this paper to explore the key competencies or 'skills' which are inherent to ICT as it exists within the Welsh education system and to investigate teachers' individual constructs and perceptions of the subject and the methods they use in the classrooms to develop those skills.

In an attempt to identify the key competences within Informatics and ICT, it is important to define the subject area. According to a number of authors (e.g. Staggers, Gassert, Curran, 2001), Informatics can be defined as being related to the structure of information, that it is a "problem or purpose-oriented" discipline, further that it has particular reference to the structure and use of information within our environment, and as such has a social implication. However, more recently, authors have used the word Informatics as being synonymous with Computer Science (Sysło, Kwiatkowska, 2008). For the purposes of this paper, the focus will be on the earlier definition which is more clearly related to that of ICT as defined within the National Curriculum documents for England and Wales. The definitions include such aspects as "communicating, problem solving" (HMI, 1977); "transmitting information and interpreting information conveyed by table, diagrams and models" (ACCAC, 2008); "Aesthetic and creative; human and social; linguistic and literary; mathematical; moral; scientific; spiritual and technological" (DES, 1985). In further defining ICT, Kennewell, Tanner, Parkinson (2000: 1) suggest that

> *Information and communications technology refers to the set of tools used to process and communicate information; to be 'ICT capable' is to be competent in controlling the situations in which those tools are applied.*

This refers to the use of higher order skills as well as the tools provided by particular applications; ICT involves the use of those skills involved in deciding which tools to use and how to use them to bring about the optimum solution. This is supported by DCSF (2002), which states that students should have the ability and confidence to use ICT equipment and software with purpose to support their work. They should also be able to identify situations where the use of ICT would be relevant. To enable this, students should be able to reflect and comment on the use of ICT, and to recognise that ICT affects the way in which people live and work (Gaskell, 2003). These ideas are similar to those of ICT literacy, which Markauskaite (2007) suggests is the use of digital technology, communication tools, and/or networks to access, manage, integrate, evaluate and create in order to function in a knowledge society.

In order to understand the nature of ICT and the key competencies within this subject area, it is not enough to analyse what is contained in curriculum specifications and schools' schemes of work. We need to ask how these competencies are developed in the classroom and, indeed, if learning these skills are particularly suited to the discrete subject of ICT/Informatics or are they better developed across the curriculum. An investigation of this development of competencies thus also needs to consider the perceptions and practices of the teacher involved, and it is that aspect on which this paper will focus.

1.1 Theoretical Framework for Pedagogical Research

Shulman (1987) designed a model for the processes involved in developing teaching and classroom practice, based on observing and interviewing a large number of teachers. The model is not a mechanistic process, but rather an underlying concept that drives best practice in pedagogy. In analysing the teachers' knowledge base, Shulman generated a number of categories to evaluate the knowledge base: Content Knowledge; General Pedagogical Knowledge; Curriculum Knowledge; Pedagogical Content Knowledge; Knowledge of learners and their characteristics; Knowledge of educational contexts; Knowledge of educational ends. One of the key points was the link between the content knowledge and the pedagogical knowledge, that the teaching of a subject will be improved if there is not only specialist knowledge of the subject but also knowledge of how best to teach the subjects and develop the skills inherent to that subject (Shulman, 1987).

More recent studies of teachers and teaching have recognised the importance of the context, including the school and the wider social and political environment. It is the work of Engestrom (2001) which has been examined in order to establish a possible analytical framework for this study. Cultural Historical Activity Theory or CHAT, allows the researcher to pay particular attention not only to the specific object of activity under research, but also the Vygotskian focus of 'mediation and discourse'. Activity theory is a developing resource which has the flexibility to adapt to any given activity within the workplace (Daniels, Edwards, Engestrom, Gallagher, 2010).

The proposition of activity theory is that human activity consists of much more than mere action, but is a socially-situated phenomenon. It is a theory or framework that examines practice, but situates that practice within an environment, which also examines the process and the purpose of that practice (Daniels, et al., 2010). In doing this, it attempts to account for the complexity of real-time activity, investigating factors that influence the activity such as the beliefs and perceptions of those central to the activity (Engestrom, Meittinen, Punmaki, 1999). Webb (2005) suggests that our increasing understanding of cognition and meta-cognition has led to the need for researchers to develop more complex models of analysis, involving aspects of influence such as the environment in which learning is to take place. Within CHAT, the activity triangle (Figure 1) represents the relationships and networks within related activity systems, in which any change and alteration of an aspect of one system, whether it be part of the tools, rules, roles, individuals, or outcomes are likely to affect another part of the system or systems. The analysis works by examining the individual components of the activity system, and looks for contradictions, which are essentially disturbances within the system (Daniels, et al., 2010).

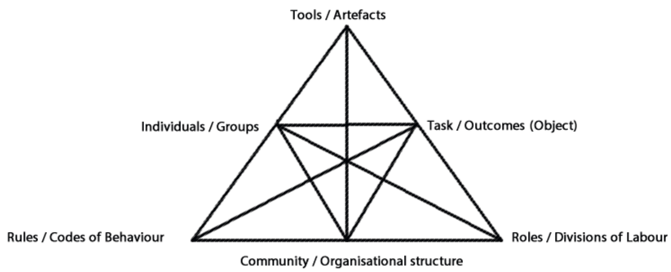


Figure 1: Activity Triangle adapted from Engestrom (1999)

This framework does not address the detail of classroom interactions which are so important in pedagogical practice, however. Kennewell (2010) suggest the use of a model based upon the analysis of the affordances (or potential for action within the setting) and constraints (the structure allowing that action to take place), which are related to the goals observed within the classroom. Consequently, this model was adopted as a framework for observing and recording classroom practice, details of which can later be extracted for analysis within the activity setting triangle.

2 Methods

This research is concerned with why the teachers teach in the way that they do, and how the development of key competencies matches with their own personal constructs. Case study is a preferred method when researchers are asking the how and why questions and it is particularly applicable when the researcher has little control over the events taking place in the macro environment (Yin, 2009). Consequently, it was decided to use three separate schools as the basis for independent case studies. The three schools represent different ways in which they develop ICT, ranging from teaching within a discrete subject based environment to one where the skills are developed through one week of intense study during the year and supported on a cross-curricular basis.

The data has been collected within each school using an initial interview, a lesson observation and subsequent reflective dialogue, then a repeat of the observation and reflection and finally another interview.

The interviews were conducted in order to gain an understanding of the teachers' perception of those skills which are definitive to the subject of ICT, whilst the observations were conducted to establish if the activity within the classrooms supported the development of the skills the teacher had identified. Because each teacher was also looking at their own practice in the development of these skills in their students there was also a post-observation dialogue, in which attempts were made to encourage the teacher to reflect on their own practice prior to a different observation and finally another interview to find out if any of the teachers' earlier perceptions had changed.

2.1 Data Analysis

Thematic analysis was used for identifying and analysing patterns of meaning in the data and ultimately to highlight the most salient meanings present. The coding has been carried out independently, in that each item of data, that is interviews, reflective dialogue and observations have been scanned for themes independently of each other, and then re-scanned in order to establish common themes prior to deeper analysis. The broad themes from Activity Theory were used across all the analysis, but the subthemes that emerged when coding the interview data differed from those found in the observational data (see Figures 2 and 3).

Using activity settings as an analytical tool makes it possible to detect contradictions, either between different activity settings or indeed within the same activity setting (Engestrom, 2001). These contradictions have been identified as existing on a number of levels, primary, secondary, tertiary and quaternary contradictions, whereby the primary contradictions are those which exist within a single mediating artefact within the triangle; secondary are those which exist between two mediating artefacts of the same triangle. Tertiary and quaternary are those where there is a disturbance between the elements of differing triangles (Hu, Webb, 2009).

Analysis of activity systems is particularly helpful in characterising change and professional environments, and identifying contradictions is valuable in explaining change in activity systems (Engestrom, 2001, Roth and Lee, 2007).

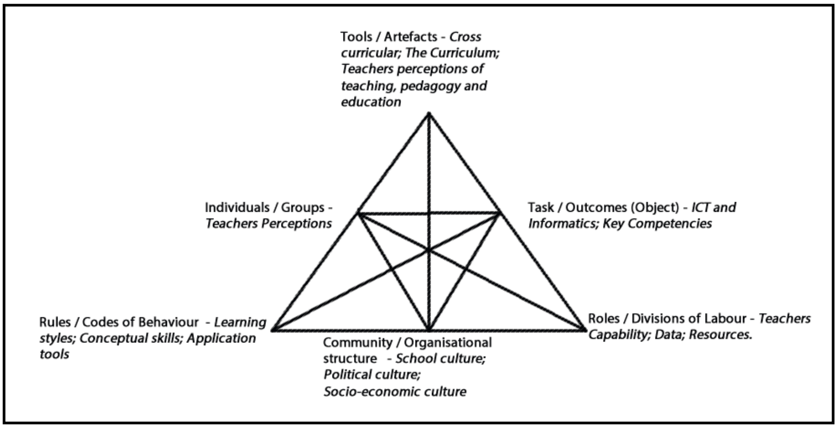


Figure 2: Interview analysis

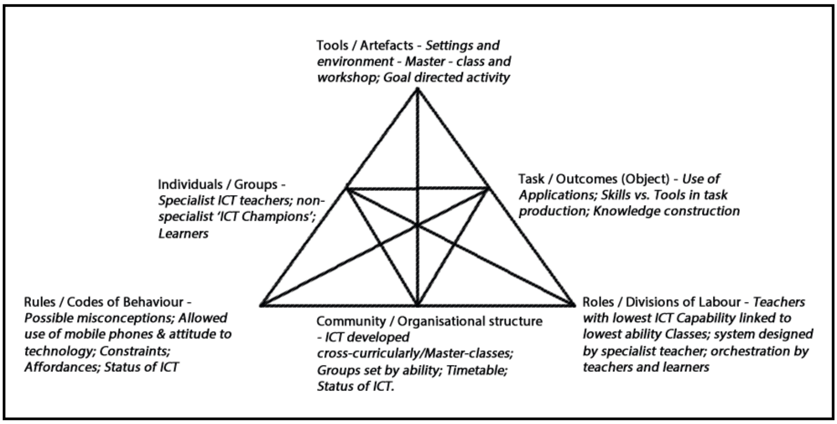


Figure 3: Observation analysis

3 Results

The results are presented in terms of the elements or mediating factors of Engestrom's triangle, cross-referenced to the themes and examples emerging from the analysis of classroom observations together with both the initial and final interviews. This paper concerns one teacher/school, for which the activity setting involves ICT being taught to the whole year group (Year 9, aged 13–14) at once through a series of 'master-classes', lasting over an hour each morning throughout a week, followed up by workshops in ability groups for the rest of the school day. The masterclasses were given to the whole cohort as a lecture by the lead ICT specialist teacher and follow up workshops supervised by the lead teacher and a number of non-specialist teachers with an interest in technology.

Tasks/Outcomes In the case of this study, the object is the development of the key competencies inherent to ICT. These are generally referred to as 'skills' by teachers, a term which can cover a wide range of general competencies and tool-specific techniques. There was a change between the initial and final interviews in that during an initial discussion of skills, there was reference to *"higher order"* skills; *"Problem solving that goes with it"*, whereas in the later interview there was an indication that skills are synonymous with the tools used to operate the various applications.

The observations were of a master class and lesson which concentrated on the strand of the ICT curriculum 'Communicating Information', and the media for implementing this was the production and editing of a video. The observations showed that there was a strong emphasis on the use of the applications associated with the tasks. Both the masterclass and the subsequent workshop concentrated on what certain tools within the application were for and what the subsequent effects were. The product here was task based, with procedures and expected outcomes shared and reinforced with the group. During the masterclass and the class-based recap there was no reference to higher order skills and the sessions were focused on the use of the tools in the software, and thus the learning outcome was likely to be tool based. The product (video) could be quite polished depending on the individual skills and creativity of the student; however evaluation and reflection seemed to be absent. There was no difference between the two observations in the outcome or object of the activity setting in that both were concerned with the production of a piece of work, and the learning that took place was focused on the tools used in the two applications, that of video production in the first observation and that of a presentation in the second.

Tools and Artefacts These include the curriculum; interpretations of the curriculum (e.g. lesson plans); software and hardware; language and conceptual understanding.

The aim of this school's curriculum is to develop the competencies predominantly through a cross-curricular approach, but with one week each year during which there is an intensive course of ICT. *"The tools are built up in the specialist weeks and then the other subjects can use those skills in their subjects."* It was felt that the statutory ICT curriculum is currently not motivating enough for the students in itself and their approach to the curriculum is designed to overcome this: *"move away from the boring routine of regular classes and do cool stuff with ICT."*

There were two physical settings: the main hall where all students were seated and listening to the lead teacher, and a variety of classrooms with networked PCs. The class observed were situated in the ICT-equipped portion of the library for the workshop, using Serif Video Plus for the task.

Examination of the classroom activity settings indicated that this aspect remained constant; the two interviews provided evidence of a change in the teacher's perception. She recognised the need for a change in curriculum, influenced by the external environment, particularly changes in the political position existing within the community and organisational mediating artefact of the triangle. In particular, if there were change in the curriculum to include aspects of computer science imposed externally then there would be a need to teach the subject with a more conceptually orientated approach. *"There is going to be huge change, not only what is going to be implemented, but also how it is going to be implemented by WAG [the body governing education in Wales] ..."; "hopefully it will move away from the tools aspect and towards more of a concept driven aspect"*.

Individuals and Groups This element appears as the 'subject' in more traditional activity theory triangles. When analysing the activity within the classroom there are at least two different perspectives, with corresponding activity systems taking place; that of the teacher, whose object is likely to be the learning that they wish to take place, and that of the students, who are less likely to recognise the development of the competencies as the objective, but are more likely to be concerned with the successful completion of the task: in this case, the production of a movie. For the purpose of this paper, the analysis of the observations will focus on that of the teacher as the Individual and it is their object that will lead the activity triangle. The individuals and groups consisted of the lead specialist ICT teacher (who designed the unique approach to developing ICT competencies used within the school), the non-specialist

ICT teachers (who were supporters of technology within the school, but apart from workshops during the masterclass weeks were only involved with the development of ICT competencies on a cross-curricular basis), and the groups of students. The observed group was perceived to have high ability in ICT. There were no differences in the activity settings of the two observations of classroom practice.

Rules and Codes of Behaviour This may include the attitude of the participants, for example the motivation of the students; it may also be linked to how the key competencies are perceived, for instance in cross-curricular development compared with discrete lessons; any use of specialist knowledge and also any perceptions of the students which may change the status of ICT to a societal role.

The motivation of the students was high when observed, and this may be due to the perception that this was not an ordinary week of lessons. The operational delivery of ICT within the system means that ICT may be perceived as a tool rather than a subject in its own right. This reflects students' use of ICT outside the classroom, as there is a culture within the school of students being allowed access to mobile phones with games or 'apps' within their lunch breaks and free time. This may reinforce the perception of ICT as a tool instead of a subject.

There was no change in the attitude of the teacher interviewed, but there was a difference in the constraints and affordances supplied by the teacher in the second observation of classroom practice, in that the students were informed of "success criteria" in order to judge the actions they needed for successful completion of the task. This also enabled the students to orchestrate their own affordances in the construction of their knowledge. There was a status quo in the pedagogical process in that the specialist teacher still interacted with the top ability set.

Community and Organisational Structures The environment of both the school and the wider community influence this area of the activity. There is a political drive to change the ICT curriculum, which is already taking place in England and is currently under review with recommendations to Welsh government under consultation. This has influenced the perception of the key competencies within this study; in the beginning there was a clear focus on skills and tools with an acknowledgement of higher order skills but subsequently there was more emphasis on the skills needed for socio-economic success with allusions to industry and the world of work. *"Greater programming so we get better industry"; "Need to look at what skills they need in the wider world to know what they will need at schools".*

During the course of this study there has been increasing impetus in the external educational arena for the need to change the curriculum and this has had an effect on the perception of ICT and informatics in the school, as noted in other sections of the results.

Roles and Divisions of Labour The key roles are that of the teacher under interview and her team of non-ICT-specialist teachers who help facilitate the workshops within this system. Two of the key themes that emerged here were resources and differentiation. The specialist ICT teacher holds the responsibility for the design of the system, the mode of delivery and the design of the teaching material, any differentiation and the assessment of the work produced. Whilst she designs any differentiated resources these resources are taught by non-specialist 'ICT Champions': *"We use teachers who have a particular interest in the use of ICT, all of them are masters at using ICT within their own subject area."* Higher ability students were supported by the teachers with more specialist ICT knowledge. The ICT specialist also has responsibility for placing the students in classes according to ability: *"Their capabilities you see very quickly. You have to differentiate very carefully, and we will set the classes by achievement"*. However she is the teacher who has least knowledge of students individually, and is reliant on general cognitive ability data which may not be the best indicator of ICT potential. Whilst the interviewee had expressed a belief that this method suited her students – *"The tools are built up in the specialist weeks and then the other subjects can use those skills in their subjects"* – its place in the curriculum implies that the subject is not held in the same esteem as other subjects afforded an hourly lesson a week: *"In secondary schools generally it would need more money, for example to have specialist teachers, I mean at the moment I am a department of one"*.

Participation of the different individuals and groups varied, in that during the master-class portion of the teaching and learning experience the action was in the hands of the specialist ICT teacher, there was no opportunity for interaction other than in a superficial checking of the recall of the instructions given. There was also no participation from the non-ICT-specialist teachers at this point. However, in the workshop setting, there was opportunity for greater questioning regarding the workings of the software under investigation. There was also opportunity for the students to interact with their own learning, as the learners in the group had a greater control over how they organised the learning of the software, and there was evidence of exploration as a strategy. If the effect of an action was perceived as successful, students discussed this with those sitting close by. In this way the learning was shared. The degree of autonomy of learning was dependent on the overall ability of the group and the confidence that this ability brings. Therefore this autonomy is likely to be reduced in the less able groups as the difficulties of the learner are greater; the group observed had the confidence to explore the tools of the software independently and had developed clear ideas about how they wished their finished product to appear.

4 Discussion and Conclusions

When discussing the nature of ICT and trying to establish the key competencies within ICT with teachers, the key themes that emerge appear to exist on two levels: those which are concerned with the operation of a specific application *"this Microsoft application process that we're going down in our current curriculum"* and those competencies which are concerned with the use of higher order skills and concepts, e.g. *"increase capability by working at higher order skills and teaching not just skills but the content behind it"*. There was also an allusion to the need for metacognition within the desired competencies.

There is a primary contradiction in the object – the intended learning – between the tool-based teaching observed in lessons and the goals stated in the initial interview which refer to 'problem solving', 'higher order skills' and concepts: *"Communicating information often is posters and PowerPoint. The kids have the skills – they get those skills in primary school – we don't need to spend the time developing that skill, we need to be looking at the concepts behind it"*. However, there are apparent differences in the goals between the two interviews, which warrant further investigation. In the final interview, there is no reference to higher order thinking and the discussion of skills focuses on tools: *"for example in spreadsheets ... we are looking at the tools within the application"* and at one point the teacher expressed the view that the term 'skills' was synonymous with the term 'tools', indicating that the key competencies within ICT were perceived to be of lower order.

This change in the object of the activity may resolve the primary contradiction, but leaves a secondary contradiction between the tools/artefacts and the object of the teachers perceptions, in that the teacher's perceptions tend to focus on the lower order tools used to operate a variety of applications and less on the higher metacognitive skills which are discussed within the National Curriculum in Wales (ACCAC, 2008), where ICT capability is described for example as having the ability to use ICT in problem solving. Examination of the scheme of work also supports the need for evaluation: *"state ways in which you can improve your work"*. However, in reality, this was superficial with an absence of any real metacognition.

Thus a change in the perception of the key competencies is emerging, in that the teacher's perception is moving from one incorporating higher levels of metacognition to a merely tool-based definition, which matches the practice observed. The contradiction within the object – between the espoused goals and the practical objectives – has been resolved but the conflict is now between the object (learning to use tools) and the artefacts (the scheme of work and the National Curriculum). Furthermore, examining the rules, community and roles, all point to a lesser status for ICT within the school compared with other subjects, which is at variance with the lead teacher's initial perceptions of having a subject which in itself develops higher order key competencies such as problem solving and metacognition.

Another contradiction lies within between the individuals/groups and the roles/division of labour in the use of non-specialist staff in the development of ICT competencies, and furthermore the assignment of those staff with lesser ICT competencies themselves to the lower ability groups. Currently it is the specialist ICT teacher who is responsible for the formation of the scheme of work, the system of teaching and the differentiation taking place to enable both the less able and the more able student. However, because this teacher does not teach at this age group apart from isolated periods throughout the school year, she does not know the students but is reliant on general data concerning cognitive ability. The non-specialist teachers know students as they teach them in other subjects. Furthermore the rationale driving this is that the higher ability group needs the specialist teacher so that she can 'push' them, however her object is tools based and given that the very students she is 'pushing' have the ability to explore tools, to orchestrate their own learning, to build their own constraints and affordances to bring about the knowledge construction, whereas the lower ability would need the constraints and affordances as implemented by the specialist teacher. Shulman's (1987) work suggests that the development of learning within a subject is dependent on the pedagogical content knowledge of the teacher rather than just subject knowledge and knowledge of learners, and it may be that those classes with the least ability would gain most from a specialist ICT teacher whereas students with good higher order skills would gain most from being challenged to apply ICT in learning other subjects.

This study has demonstrated change, predominantly the change in teacher perceptions of the key competencies of ICT. The realisation that the teacher's object is really the learning of ICT tools rather than higher-order skills may have been brought about by the study itself, or indeed by external influences such as the present political and educational arena within England and Wales. In order to resolve the contradictions and implement the current curriculum, a number of changes to the activity system will need to be adopted. There will need to be changes to the curriculum as it is perceived within the school, in that there will need to be more expectation and support for the students to use the metacognitive skills. At present, the scheme of work and mode of teaching limits the students to a tool-based construction of knowledge. Furthermore, in order to effectively achieve this, there may be a need to re-assess the scheduling of the subject to allow for the students to reflect on their learning and explore the use of the applications in differing situations. As the political environment drives change in the subject towards a more conceptual nature the impact on the mediating factors of community and organisational structure within the school environment may take the form of greater status afforded to the subject. Any subsequent increased teaching time is likely to have cost implications as there may be the need to employ further specialist teachers.

Finally, there may be a need to further evaluate the pedagogical knowledge which is fundamental to this scheme of work and use specialism where there is greater need, with those less able students. By facilitating the more able students to orchestrate their own learning and construction of knowledge, using self-created affordances and con-straints, especially those which may be prevalent in problem solving scenarios, whilst correcting any misconceptions which may occur, then those higher order metacognitive competencies which are being highlighted within the subject area are likely to be better developed.

The analysis of one case study is not sufficient to fully identify contradictions in the systems for developing key competencies in ICT. The use of CHAT in the analysis has proved valuable, however, and promises to help reveal further insights in subsequent case studies and cross-case analysis. Furthermore, the work has provided a baseline from which to explore the effects of changes to the activity systems as the statutory curriculum experiences a more fundamental shift towards computer science.

References

ACCAC (2008). Information Technology in the National Curriculum in Wales. Cardiff: ACCAC.

Daniels, H., Edwards, A., Engestrom, Y., Gallagher, T. (2010). *Activity Theory in Practice*. Oxford: Routledge.

DES (1985). *The curriculum from 5–16. Curriculum Matters 2*. London: HMSO.

DCELLS (2009). *Information and Communication Technology guidance for Key Stages 2 and 3*. Cardiff: Welsh Assembly Governement.

DCSF (2002). *National Strategies Secondary ICT Framework*. [Online]. Available at: http://nationalstrategies.standards.dcsf.gov.uk/secondary/secondaryframeworks/ictframework [Accessed 20 August 2009].

Engeström, Y. (2001). Expansive Learning at Work: toward an activity theoretical reconceptualization. *Journal Of Education & Work*, 14(1), pp. 133–156.

Engestrom, Y., Meittinen, R., Punmaki, R.-L. (1999). *Perspectives on Activity Theory.* Cambridge: Cambridge University Press.

Gaskell, C. (2003). Capability or Office Fodder. ACITT, SUMMER, pp. 22–37.

Hu, L., Webb, M. (2009). Integrating ICT to higher education in China: From the perspective of Activity Theory. *Education and Information Technologies*, Volume 14, pp. 143–161.

HMI (1977). *Curriculum 11–16 Working Papers by HM Inspectorate: a contribution to current debate*, London: HMSO.

Kennewell, S. (2010). Analysing the impact of information technology on activity and learning. In: A. McDougall, J. Murnane, A. Jones, N. Reynolds, eds. *Researching IT in Education: Theory, Practice and Future directions*. Abingdon: Routledge, pp. 112–120.

Kennewell, S., Parkinson, J., Tanner, H. (2000). *Developing the ICT Capable School*. London: Routledge Falmer.

Markauskaite, L. (2007). *Exploring the structure of trainee teachers' ICT literacy: the main components of, and relationships between, general cognitive and technical capabilities*.

Roth, W. M., Lee, Y.-J. (2007). Vygotsky's Neglected Legacy: Cultural- Historical Activity Theory. *Review of Educational Research*, 77(2), pp. 186–232.

Shulman, L. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, 57(1) pp. 1–22.

Staggers, N., Gassert, C., Curran, C. (2001). Informatics competencies for nurses at four levels of practice. *The Journal of nursing education*, 40(7), pp. 303–316.

Sysło, M., Kwiatkowska, A. (2008). Challenging face of informatics education in Poland. In: *Informatics Education-Supporting Computational Thinking*. Berlin: Springer, pp. 1–18.

Webb, M. (2005). Affordances of ICT in science learning: implications for an integrated pedagogy. *International Journal of Science Education*, 27(6), pp. 705–735.

Yin, R. (2009). *Case Study Research Design and Methods*. 4th ed. London: Sage.

Biographies

Steve Kennewell is Visiting Professor at Cardiff Metropolitan University, and is Editor-in-Chief of the ITTE association journal Technology, Pedagogy and Education. Until recently, he worked in research and teacher education at Swansea Metropolitan University.

Jan Barnes is Senior Lecturer and Lead in TEL, Computer Science and Business Studies in Secondary ITET at University Wales Trinity Saint David. She is presently finishing her PhD.

Competences of Undergraduate Computer Science Students

Kathrin Bröker, Uwe Kastens, Johannes Magenheim
Institute of Computer Science
University of Paderborn
Fürstenallee 11
33102 Paderborn Germany
{kathyb, uwe, jsm}@upb.de

Abstract: The paper presents two approaches to the development of a Computer Science Competence Model for the needs of curriculum development and evaluation in Higher Education. A normative-theoretical approach is based on the AKT and ACM/IEEE curriculum and will be used within the recommendations of the German Informatics Society (GI) for the design of CS curricula. An empirically oriented approach refines the categories of the first one with regard to specific subject areas by conducting content analysis on CS curricula of important universities from several countries. The refined model will be used for the needs of students' e-assessment and subsequent affirmative action of the CS departments.

Keywords: Competences, Competence Measurement, Curriculum Development, Computer Science Education, Recommendations for CS-Curricula in Higher Education

1 Introduction

The central demand of the Bologna reform is the outcome-oriented implementation of study courses, which include competence-oriented descriptions of the curricula and respective competence-oriented assessment procedures. These demands are often implemented only in a superficial way. To accomplish the demands of Bologna it is not enough to adapt the curricula. Developing a competence-oriented course of studies is a process, which consists of several activities. According to Schaper (2012) these are e.g.: developing a competence oriented curriculum, approaches for a competence oriented organization of teaching and learning, methods for a competence oriented assessment and also approaches for a course related advancement of competences.

The base for most of these activities is a competence model. It is needed for the definition of learning outcomes on different levels of granularity of a study course like modules, lectures or seminars. Additionally for the needs of assessment and in order to identify affirmative action a concept of competence measurement is necessary.

This paper explains two different methodologies for developing a competence model for computer science undergraduates. The results of these approaches and the additional value of relating them to each other are described.

The first methodology represents a normative approach. The competence model is the agreed and confirmed result of a broad discussion between CS domain experts on the basis of a generic psychological competence model. In this case, a CS expert group of the German National Computer Society (GI) developed a CS competence model in order to issue design recommendations for CS curricula in Higher Education; see *Methodology A* in Fig. 1.

The second methodology represents an empirically oriented approach and derives the competence model by means of the content analysis of international CS curricula; see *Methodology B* in Fig. 1. The applied content analysis reveals common CS-competences addressed in different CS study courses of universities. The students at the end of their CS-bachelor studies should have achieved them. This research is part of a project conducted by the Computer Science Learning Center (LZI)[1] of our university. The main objective of the project is to support CS-students' learning processes by identifying learning barriers and contribute to resolve them. An empirically grounded competence model in the area of software engineering, software development and programming forms the basis of competence measurement instruments, which can be used for purposes of diagnosis, students' self-assessment and for affirmative action of the LZI.

1 Funded by the Ministry of Education and Research (BMBF).

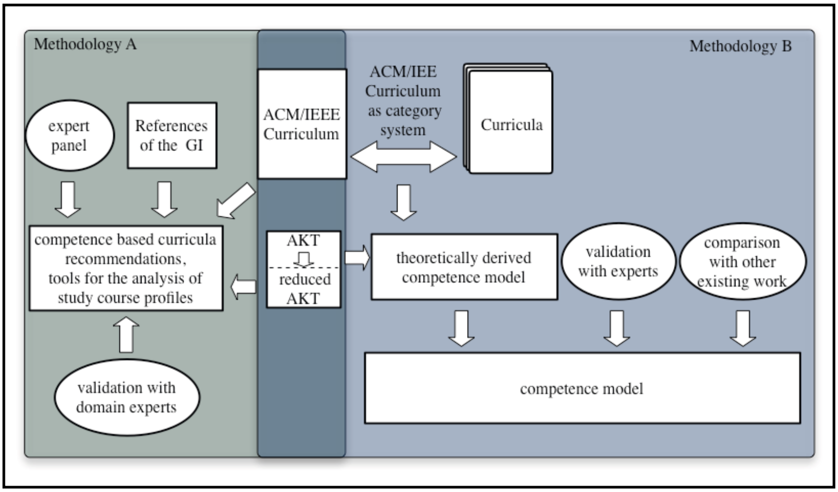


Figure 1: Methodologies for competence model development

2 Concepts of Competence

During recent years the research landscape in the field of competence models and competence measurement diversified. Origin of these research efforts was the Bologna process and STEM-related research, especially the PISA project. Nevertheless, only little research for higher education and academic competences exists. Before we introduce both approaches to a competence model we give a short definition of the term competence followed by a short introduction to AKT and related psychological models of cognition and a short overview on theoretical aspects of competence models and measurement.

2.1 Competences and Models of Cognition

The term competence has many definitions in different subjects. In the area of curriculum development e.g. the competence structure model provided by the EU-Tuning project is often applied (Tuning, 2013). This is not the place to discuss all the different definitions. Consequently we only introduce the definition used in our projects. As applied in most of the recent projects we use the definition of Weinert (2001) (original in German, translation by the author). Competences are “the existence of learnable cognitive abilities and skills which are needed for problem solving as well as the associated motivational, volitional and social capabilities and skills which are essential for successful and responsible problem solving in variable situations." (Weinert, 2001, p. 27) This definition implies, that competences are learnable by interventions. Accordingly we can use a competence assessment to measure if any intervention during the educational process helps to develop competences or to achieve a higher competence level. To take account of this aspect during curricula development we need a competence model that allows grading. Therefore in both methodologies presented here, we refer to the "Taxonomy for learning, teaching, and assessing" by Anderson and Krathwohl (Krathwohl, 2002) also named as AKT, which is a revision of the well known taxonomy of Bloom. It can be applied for grading of the cognitive dimensions of competence. The AKT model regards learning objectives as a combination of a certain type of knowledge and a certain cognitive process, forming the two dimensions of the original Bloom's taxonomy in the following way (see Krathwohl, 2002, p. 214): A: *Knowledge Dimension* (Factual, Conceptual, Procedural and Metacognitive knowledge); B: *Cognitive Process Dimension* (Remember, Understand, Apply, Analyze, Evaluate, Create). While the knowledge dimension offers classification of knowledge types, which are relevant in the context of learning, the second dimension of cognitive processes can be used for a hierarchical classification of competences. The AKT model is also applied in both methodologies for CS curriculum development and competence assessment presented in this paper.

2.2 Competence Modeling and Measurement in CS

In Computer Science many projects exist in the context of key competences, especially ICT competences. In contrast there are only a few projects where subject specific competence models are developed. Two of these projects are the German KUI (Berges et al., 2013) and MoKoM (Linck et al., 2013) project. The MoKoM project has its focus on developing a competence model for students at schools. They developed a competence model for informatics modeling and system comprehension. In contrast to that of the project KUI which focuses on competences for future teachers. The competence model here is divided into 3 parts: competences on subject matter knowledge (CK), competences on pedagogical content knowledge (PCK) and non-cognitive competences (NCC) (Hubwieser, Magenheim, Mühling, and Ruf, 2013), (Schaper et al., 2013). In addition there are some projects with research in students' competences at universities. Nevertheless until now, no national or international project has developed a concrete competence model for subject specific competences in computer science for academics. The only existing models are for example the IEEE Software Engineering Body of Knowledge (SWEBOK) (Society, Bourque, and Dupuis, 2004) or the ACM/IEEE Curriculum (The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society 2012, 2013). Nevertheless these documents rather describe knowledge, which should be part of a curriculum, than real competences. Furthermore, these documents are not empirically verified. Consequently we don't know if universities really teach the topics mentioned in these curricula. Only accreditation rules give us a first hint. Moreover they are not as specific as the ACM/IEEE Curriculum. However these documents form a good basis for the development of a specific CS competence model. Therefore, we refer to these approaches when we contextualize and specify our cognitive competence according to the AKT-model. In addition to missing CS competence models there are hardly assessment results for CS competences, which are based on a sound methodology derived from a CS competence model by now.

2.3 Competence Models and Measurement

Competence Models are the basis for developing measuring instruments and the description of their results. According to Koeppen, Hartig, Klieme, and Leutner (2013) there are several requirements for competence models. First such a model should represent the internal competence structure. Second the levels of competences should be described and at last these models should consider changes arising in educational processes. The competence structure models should reveal existing relations between the accomplishments of different requirements. Competence level models in particular can describe which requirements different people can accomplish. These models give us the opportunity to identify persons with distinguished competences. Competence level models are used for measuring the outcomes of educational processes, for example, if we want to measure the effectiveness of interventions at the Informatics Learning Center. For the measurement of competences psychometric models are used. For both forms of models the *Item-Response-Theory (IRT)* (Hambleton, Waminanthan, and Rogers, 1991) is applicable. For CS Fuller (Fuller et al., 2007) developed a competence model based on AKT. The differences and similarities between the CS-AKT presented here and the concept of Fuller is not subject of this article. In the authors' opinion the CS-AKT meets better the demands for developing CS program guidelines, while Fuller's concept fits very well for the analysis of specific competence-oriented CS learning processes.

3 Theoretical Approach to a CS Competence Model

In the following we describe two different strategies for developing a competence model. The first approach was applied by an expert group of the German Informatics Society (GI). The task of the group was to develop recommendations and guidelines for the educational design of Bachelor and Master CS study programs at German Universities and Universities of Applied Science. The group's strategy to resolve this problem comprises the following tasks (see Fig. 1):

1. Selection and rationale of a generic competence model that enables the grading of cognitive competence components: AKT
2. Adaption of the AKT to the needs of a CS competence model: adapted CS-AKT
3. Selection and rationale of core CS subject areas according to former and current national and international CS curricula and CS curricula guidelines: CS core subject areas (CS-CSA)
4. Definition and classification of CS competences according to CS-AKT and CS-CSA
5. Describing the competence profile of CS study courses according to the expected competences students' should achieve when attending mandatory modules, lectures and seminars during the study program.

Competence-Dimensions				
Cognitive Process Dimension	**P1**	**P2**	**P3**	**P4**
Knowledge Dimension: K1: factual K2: conceptual K3: procedural K4: Metacognition	Understand	Apply	Analyse	Create
Type and Complexity of Context: C1: without Context C2: small Examples C3: complex Examples C4: internal projects C5: industrial projects		**P2a** Transfer	**P3a** Evaluate	

Figure 2: Adapted CS-AKT for cognitive CS competences

(1): In order to describe CS competence structures and levels we use the AKT-matrix (Krathwohl, 2002) as a generic cognitive competence model. The AKT provides us with a differentiated concept of competence components regarding action-oriented process dimensions of knowledge achievement as well as dimensions of knowledge that take internal aspects of knowledge processing in the human brain into account. We are aware of the fact, that the AKT originally addresses learning objectives. Therefore, we relate the activities, which were described in the cells of the matrix, to specific CS topics, combined CS subject-related activities to a specific context of action and also considered the complexity of the tasks. Thus, the descriptions in the cells meet the requirements of the competence definition according to Weinert. The motivational and volitional aspects of competence remain largely excluded from this classification.

(2): In order to reduce the complexity of the matrix we deleted those cells, which are not or less relevant for CS competences according to the following principles:

a. Competences that are characterized only by remembering without understanding are not sufficient in higher education CS study programs. Therefore, the column 'remembering' of the original AKT-matrix was deleted.
b. At the other end of the process dimension scale the generation of new knowledge is mostly out of scope in CS Bachelor study programs, but it is not deleted in the adapted CS-AKT in anticipation of future descriptions of master's degrees.
c. In order to enable a compact and coherent description of competences of the process dimension with regard to different types of knowledge, we merged the original four rows into one. The addressed types of knowledge in the competence description are indicated by respective annotations W1 to W4.
d. For the description of CS study programs, it is important that competences are characterized with regard to their requested application context. Since the original AKT-matrix doesn't offer a specific structure for this demand, we added a second row that addresses the types and the complexity of the application context. By means of annotations we can classify competences with K1-K5 when describing them (see Fig. 2). For the level 'understand' we didn't introduce this concept, because small examples are always used in this area of knowledge achievement and the application of more complex examples already can be assigned to the process dimension 'apply'.

e. The competence component *P2a 'transfer'* considers the ability to perform competence related action in different application contexts, while *P3a 'evaluate'* indicate a student's ability to evaluate an informatics system on the basis of a previous analysis. In CS study programs the competence components 'transfer' and 'evaluate' do make sense in our opinion only with regard to the context. Therefore, those components are assigned to the dimension (row) 'Types and complexity of context'. The accuracy of the competence descriptions is not affected by this assignment.

The adapted CS-AKT has proven to be effective for describing competences of CS Bachelor programs. The reduction and adaption should have preserved the underlying AKT methodology.

(3): After an intensive discourse considering current CS curricula recommendations, (e.g. ACM/IEEE, 2013) and former GI CS recommendations the GI expert group selected core CS subject areas (CS-CSA), where students' should have achieved competences according to the CS-AKT model at the end of their Bachelor programs.

The subject areas were derived from the knowledge areas of the ACM/IEEE curriculum without considering the different aspiration levels.

(4): For each of those CS-CSA, competences will be described according to the CS-AKT. In order to illustrate this procedure, we show an example for using the adapted CS-AKT:

Fig. 3 instantiates the CS-AKT scheme of Fig. 2 for the subject area of *Programming Languages and Programming Methods*, which is one of about 20 subject areas to cover Bachelor programs in Computer Science. The stated competences have been cross-checked against the ACM Computer Science Curriculum (ACM/IEEE, 2013), the (FTI, 2004), and a particular Bachelors Program at the University of Paderborn (UPB, 2009).

	P1	P2	P3	P4
Knowledge Dimension	***Understand*** Understand constructs, properties and typical programming techniques for more than one widely used programming language. Understand the characteristics of the programming paradigms: imperative, object-oriented, functional, and logic programming. Understand fundamental concepts of programming languages, like syntax, binding of names, type systems, storage structures, function calls and parameter passing.	***Apply*** Be able to develop programs for algorithmic tasks in several programming languages of different programming paradigms; apply suitable and adequate programming techniques.	***Analyze*** Be able to learn any new programming language on the base of language documents.	
Type and Complexity of Context		**P2a: Transfer** Be able to develop a software system for a specific application task of medium complexity in a language that is suitable for the task. Make sure to apply adequate programming methods, and to achieve good quality of the resulting software.	**P3a: Evaluate** Be able to analyze programming paradigms and programming languages with respect to their applicability for the solution of a particular application task.	

Figure 3: CS-AKT applied for Programming Languages and Programming Methods

In the *Knowledge* dimension the overall objective is formulated in field *P3*: At the end of a Bachelors program students shall be able to learn any new language autonomously on the base of published language documents. That competence is essential, because several new languages will be developed and some paradigm shifts will happen during their professional life. The competences formulated in *P1* and *P2* are preconditions for that of *P3*. The example Bachelor program (UPB, 2009) addresses these competences by a sequence of lectures: *Foundations of Programming* in the first and second semester, where students are introduced into programming in one particular language (Java, in this case), *Foundations of Programming Languages* in the second semester, where they learn to understand constructs and properties of different languages and programming paradigms, and finally in the fifth semester an advanced course on *Programming Languages and Compilers*, where they achieve a deeper insight in language development and implementation.

In the *Complexity and Context* dimension the competences stated in field *P3a* also build on those stated in *P2a*. Both address the ability of the students to use their understanding of programming languages and methods to develop software of good quality in projects of increasing complexity and with decreasing guidance of supervisors. In our example Bachelor program (UPB, 2009) the competences of *P2a* are to be achieved in a practical course in the third and fourth semester; *P3a* is usually a result of the implementation developed in context of the Bachelor thesis.

In the cognitive process dimension *P5: Create* usually does not apply for Bachelor programs. It is kept in the adapted CS-AKT to be used for the description of competences achieved in Masters programs. For this subject area one would for example describe in the knowledge dimension the competence to develop and implement new domain-specific languages. Of course, it requires that more advanced competences are achieved by Masters students in the lower levels of the process dimension.

(5): Finally the competence profiles of CS study programs can be described according to the expected competences students' should achieve during their studies. A classification of types of scientific activities in the modules, seminars and lectures of the study program will be provided, that applies the CS-AKT. The classification of scientific working will be conducted on the basis of those competences, which are expected that they are mediated primarily in the respective course or module. The profile of a program is then obtained from the frequency of occurrence of types of scientific work and its distribution in the average view on a program. By providing this differentiated profile of a study program the CS-AKT also contributes to resolve problems of the bipolar categorization of 'scientific-oriented' and 'application-oriented' study programs.

4 Empirically Oriented Approach to a CS Competence Model

As a second strategy for developing a CS competence model we now describe the approach that is part of an initiative of our learning center (LZI). This is an empirically oriented approach using content analysis. The result should be a more detailed competence model in a specific topic area in contrast to the model developed by the GI experts. Therefore, we can use this strategy to specify the competences of the experts mentioned before. Since the LZI initiative aims ultimately on affirmative action for students to successfully graduate in CS, the results of the research must be very specific and target group oriented.

This is the reason why we focus in this approach on the areas of Software Development, Software Engineering and Programming. The applied research methodology B (see Fig. 1) consists of the following steps:

1. Content analysis of existing CS curricula of selected Universities on the basis of the topic categories provided ACM/IEEE recommendations (ACM /IEEE, 2013).
2. Conducting expert ratings in order to validate the derived competences in the identified topic areas.
3. Aggregate the results of the expert interviews with existing empirical derived competence models in order to gain a consolidated competence model in the topic area.
4. Development of test instruments for students' self-assessment and the evaluation of seminars and lectures in order to derive affirmative action at the LZI.

In the following sections we focus on the empirical research indicated in (1).

4.1 A Competence Model derived by Content Analysis

To develop a Competence Model by means of content analysis we adapted the strategy used in previous projects like MoKoM and KUI. Fig. 1 provides an overview of this process (Methodology B). Firstly we analyzed several Computer Science Bachelor Curricula of universities from all over the world applying the method of deductive content analysis according to Mayring (2010). Here we start with the ACM/IEEE Curriculum (ACM/IEEE, 2013) in order to firstly derive classification categories. For practical coding we used the software *MaxQDA* (www.maxqda.com).

In the ACM/IEEE Curriculum, the basis of our analysis experts defined a catalog of 18 knowledge areas to cover computer science. Each knowledge area consists of several knowledge units. For each of these knowledge units there is a description of the content and the learning outcomes. In addition, it is defined how many hours should be taken for each knowledge unit during a computer science program. In short the importance of each knowledge unit is shown.

We used these knowledge areas and knowledge units as our categories and subcategories during the content analysis. As the text corpus served different computer science bachelor curricula: The top ten of the QA Ranking[2] and in

2 http://www.topuniversities.com/university-rankings/university-subject-rankings/2013/computer-scienceand-information-systems

addition the top Universities of Switzerland, Singapore, India and eight of the Top Ten of German Universities in Computer Science[3] We decided to use these additional Universities, because we want to gain an overview of different countries. The Top Ten of the QA Ranking contains mostly Universities from the US. Because curricula often have obligatory and non-obligatory courses we decided to analyze only the obligatory courses. Otherwise there are too many variants. Additionally, our observations at the Learning Center and at our institute indicate that students' problems often begin early during the basic courses. Thus, if we want to measure competences according these courses, a look at obligatory subjects is sufficient.

After building the category system and the text corpus the minimal (single word/subject term) and maximal (paragraph) coding units were defined. During the following analysis, the coder assigned the identified coding units to the appropriate categories of the ACM/IEEE curriculum. It was possible to map all code units to our category system derived from the ACM/IEEE curriculum, so there was no need to add other categories to the system during our analysis.

It has to be mentioned that the richness in details of the curricula varies considerably. Some only contain information about the course like title and some organizational information. Other curricula provide detailed descriptions of the course contents and learning outcomes or competences. In addition the representation of the learning contents differs in its form. On the one hand curricula provide only pure lists of keywords, on the other hand some list concrete competences. Fig. 4 shows two examples for the difference in course descriptions.

3 http://www.wiwo.de/ranking-die-besten-unis-und-fachhochschulen/8046582.html

Ludwig-Maximilian University Munich[4] Course: Introduction to Programming	**Indian Institute of Technology Bombay**[5] Course: Computer Programming and Utilization
Qualification Aims: The students will be able to implement solutions for small and manageable problems algorithmically and to realize them with a high level programming language as executable programs. Furthermore, students develop an understanding of the general principles of programming and programming languages. This lays the foundation to ensure that the students (after further experiences in the course of study) may become familiar quickly and accurately with any programming language. Additional there exists a course description with the covered topics.	Description: This course provides an introduction to problem solving with computers using a modern language such as Java or C/C++. Topics covered will include: * Utilization: Developer fundamentals such as editor, integrated programming environment, Unix shell, modules, libraries. * Programming features: Machine representation, primitive types, arrays and records, objects, expressions, control statements, iteration, procedures, functions and basic i/o. * Applications: Sample problems in engineering, science, text processing and numerical methods.

Figure 4: Cuttings from two different curricula

On the basis of these curricula analysis we have got an overview about CS teaching content of several universities and additionally if there exists a kind of core knowledge to be achieved by the students. Because we focus on the areas of Software Development, Software Engineering and Programming we only covered the ACM/IEEE knowledge areas: Programming Languages, Software development Fundamentals and Software Engineering during our curricula analysis.

4.2 Results of the Curricula Analysis

We now present our first results of the curricula analysis after the coding process. As mentioned before the richness in details of the curricula descriptions varies considerably. As a result we do not consider the number of occurrences

4 http://www.uni-muenchen.de/studium/studienangebot/studiengaenge/studienfaecher/informatik/bachelor1/modulhandbuch/16_mhdb_nf_informatik_60_ects_psto_07_10_2010_en.pdf [l.v. 14.02.2014]

5 http://www.cse.iitb.ac.in/page95 [l.v. 14.02.2014]

of a single coding, but only the existence of a coding for a category in a curriculum. The frequencies of curricula that address a category can be seen in Fig. 5. Fig. 5 shows that about half of these subcategories occur in 50 % or more of the curricula. For building our competence model we decide to concentrate on the coding of these subcategories, which are shown in Fig. 6.

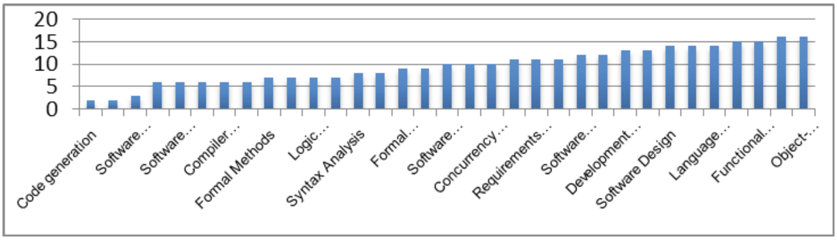


Figure 5: Results of the Curricula Analysis

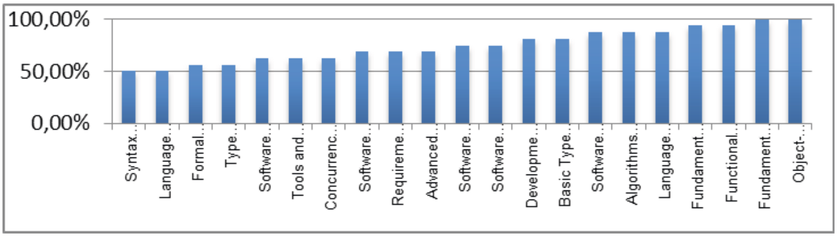


Figure 6: Subcategories occur in 50 % or more of the curricula

In a second step we compared these categories with the ACM/IEEE Curriculum again. In the ACM/IEEE Curriculum all knowledge units are classified by the categories "Core" or "Elective" where core is subdivided into "Tier-1" and "Tier-2". With these categories the ACM/IEEE Curriculum defines the relevance of the included subject areas and knowledge units for CS-curricula. Additionally a curriculum should include significant elective material (ACM/IEEE, 2013).

During the comparison of our results with such categories belonging to Tier-1 or Tier-2 we only found four subcategories, which belong to Tier-2 and which occur in not more than 50 % our curricula. These categories are: Software Reliability (3), Software Evolution (6), Program Representation (2) and Event Driven (6). During our next step, to develop the competence model, we will not consider the codings of these categories. Consequently we have to check, if our later validation of the competence model shows a lack regarding these categories. According to this validation we will add or not add the competences for these categories to our competence model. Nevertheless our results show that many curricula contain the topics the ACM/IEEE Curriculum suggests and we can say that these categories are part of a common computer science knowledge every computer scientist should have. On the basis of to such an empirically derived subject areas we will develop our competence model considering the CS-AKT described above.

4.3 Next steps – Follow up research

According to (2) and (3) a validation of the derived competence model by experts, a comparison with the IEEE Software Engineering Body of Knowledge (SWEBOK) (Society, Bourque and Dupuis, 2004) and other existing research like e.g. the model of Fuller (Fuller et. al., 2007) will be conducted. This aggregated and consolidated competence model will structure the competences in competence components and will graduate levels for each of these components. Finally, according to (4) we want to develop an e-Assessment tool for competence measurement in this specific topic area. E-Assessment will give reason for specific affirmative action at the LZI at our university and analyses e.g.: equality of learning conditions for each tested person, direct interpretation of the data, instant feedback and greater flexibility with respect to location and time. Additional reasons for an e-Assessment regarding to the aims of our Learning Center are that its use provides the opportunity to utilize such an assessment as a ubiquitous self-Assessment tool. That means students can use our assessment tool for their own to check their personal skills and identify their deficits. This is for example useful before they start a master study. An additional advantage is the opportunity to build an adaptive test or an adaptive e-Learning course later, based on the outcomes of the test. With such an instrument we can directly support our students to overcome their deficits and learning barriers with the greatest flexibility regarding location and time. For an assessment we have to create different exercises and items. At the end we need at least one item for each competence component mentioned in our competence model. In addition we have to add the aspiration level for each item.

To measure the competences, a competence model and test items are not enough, additionally a psychometric model is needed. With such a model we can map the answers of a person to the corresponding test criterion (here a competence). Here the *Item-Response-Theory (IRT)* will be applied. The reason for using IRT is, that it gives us the opportunity to measure different levels of competences. Based on our competence model, derived items and an assigned psychometric model e-assessment and measurement of competences at our LZI will be conducted.

4.4 Comparison

In contrast to the GI-experts who developed a competence model for curricula development, the results of the empirically oriented approach should be a more detailed competence model for a specific topic area. Furthermore, the competence model of the experts can be used as a framework for the concrete competences of the empirically oriented approach. For example, the competence: "Understand the characteristics of the programming paradigms: imperative, object-oriented, functional and logic programming." of the experts doesn't name the concrete characteristics of the programming paradigms. Certainly they are not important for curricula development. In contrast this information is important for developing lectures and competence assessments. Here the empirically oriented approach provides a more detailed competence model. Our first results show topics like object-oriented, functional and logic programming. In a next step we will derive competences for these topics. After this our competence model should describe the concrete competences for programming paradigms in object-oriented, functional and logic programming like the characteristics for each. All in all it refines the components of the CS-AKT-model and describes its competences in a more specific way and focused on a specific subject area.

5 Conclusion and Further Work

In this paper we present two methodologies for developing a computer science competence model. The first approach is a normative one where an expert consortium develops the competence model on the basis of existing curricula and references. This approach has the objective to develop national guidelines and recommendations for CS curriculum design in Higher Education. The results of this approach can be seen as a framework for the second methodology we present.

This second approach is an empirically oriented one and starts with a content analysis on different CS-curricula. Afterwards a competence model is build on the basis of the codings of the content analysis. This competence model should be more concrete than the expert one and can be used for the development of an instrument for competence measurement in the subject area of software development and programming. Next steps in this second approach are to derive the competences from the coding's and to validate and revise the resulting competence model. After this step we can start with the development of the assessment tools. This procedure of refinement may be applied also for other subject areas of the CS-AKT model. With our competence models we firstly provide the ability to develop guidelines and recommendations for CS curricula in Higher Education and secondly, in addition with our assessment, the ability to find out the deficits of our students in a more concrete way than by means of observation only. These concrete results will give computer science departments the opportunity to develop competence oriented curricula and specific interventions to help students in overcoming their deficits and in addition to measure the effect of these interventions accordingly.

References

Berges, M., Hubwieser, P., Magenheim, J., Bender, E., Bröker, K., Margaritis-Kopecki, M., Neugebauer, J., Schaper, N., Schubert, S., Ohrndorf, L. (2013). Developing a competency model for teaching computer science in schools. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13) (p. 327)*. ACM, New York, NY, USA.

Denning, P. J. (November 2003). Great principles of computing. In *Communications of the ACM*, 34(11), pp. 15–20.

Fakultätentag Informatik (2004). *Recommendations for Setting Up Consecutive Bachelor's and Master's Programs in Informatics at Universities*. Resolution passed by the Plenary Session of the German Council of Informatics Departments. Retrieved from: http://www.ft-informatik.de/uploads/tx_sbdownloader/BaMaRecommendations.pdf. last access: 17.02.2014.

Fuller, U. et. al (2007). Developing a computer science-specific learning taxonomy. SIGCSE Bull.39, 4, pp. 152–170.

Hambleton, R. K., Swaminathan, H., Rogers, H. J. (1991). *Fundamentals of Item Response Theory*. Newbury Park, CA. Sage Press.

Hubwieser, P., Magenheim, J., Mühling, A., Ruf, A. (2013). Towards a conceptualization of pedagogical content knowledge of computer science. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pp. 1–8.

Koeppen, K., Hartig, J., Klieme, E., Leutner, D. (2013). Competence Models for Assessing Individual Learning Outcomes and Evaluating Educational Processes – A Priority Program of the German Research Foundation (DFG) In Blömeke, S., Zlatkin-Troitschanskaia, O., Kuhn, C., Leutne, D. (Eds.), *Modeling and Measuring Competencies in Higher Education* (pp. 171–192). Volume 1 of Professional and Vet Learning. AW Rotterdam: SensePublishers.

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview, *Theory Into Practice*, 41(4), pp. 212–218.

Linck, B., Ohrndorf, L., Schubert, S., Stechert, P., Magenheim, J., Nelles, W., Neugebauer, J., Schaper, N. (March 2013). Competence model for informatics modelling and system comprehension. In *Global Engineering Education Conference (EDUCON'13)*, pp. 85–93.

Mayring, P. (2010). *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. Weinheim: Beltz Pädagogik, Beltz.

Schaper, N. Kompetenzorientierung in Studium und Lehre (2012). *Fachgutachten für die Hochschulrektorenkonferenz*, Bonn: HRK.

Schaper, N., Magenheim, J., Schubert, S., Hubwieser, P., Bender, E., Margaritis, M., Ohrndorf, L., Berges, M. (2013). Competencies of Teaching Computer Science. In

Blömeke, S., Zlatkin-Troitschanskaia, O. (Eds) *KoKoHs Working Papers no.3,* pp. 32–34.

Society, I. C., Bourque, P., Dupuis, R. (eds.) (2004). *Software Engineering Body of Knowledge (SWEBOK)*. EUA: Angela Burgess.

The Joint Task Force on Computing Curricula Association for Computing Machinery IEEE-Computer Society 2012 (2013). *Computer Science Curricula 2013*

Tuning, Educational Structures in Europe, Competenceshttp://www.unideusto.org/tuningeu/competences.html last access: 14.02.2014.

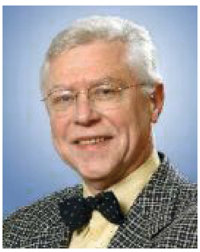
UNESCO (2012). *UNESCO ICT Competency Framework for Teachers*. UNESCO. Retrieved from: http://unesdoc.unesco.org/images/0021/002134/213475E.pdf, last access: 14.10.2013.

Universität Paderborn (2009). *Module Handbook 2009 Version 2-Bachelor and Master Program*. Retrieved from: http://www.cs.uni-paderborn.de/fileadmin/Informatik/Institut/englishPage/ModulHandBook/Module_Handbook_GB_11_02_2013-ModuleMaster.pdf, last access: 17.02.1014.

Weinert, F. E. (2001). Vergleichende Leistungsmessung in Schulen – eine umstrittende Selbstverständlichkeit. In Weinert, F. E. (Ed.). *Leistungsmessungen in Schulen* (pp. 17–31). Weinheim: Beltz

Biographies

Kathrin Bröker is research associate and PhD-Student at the Computer Science Education Group at the University of Paderborn, Germany. There she is responsible for the Computer Science Learning Center. Her research areas are CSE and E-Learning.



Uwe Kastens is head of the Research Group Programming Languages and Compilers at the University of Paderborn. The research interest of this group centers around the design, the compilation, and the application of programming languages and domain-specific languages. He is member of the board of the German Council of Informatics Departments and Chairman of the Study Commission. He is also member of IFIP WG 2.4.

Johannes Magenheim is a Professor for Computer Science Education at the Institute of Computer Science at the University of Paderborn. His primary areas of research and teaching are CSE and E-Learning. He is member of different working groups of the German Informatics Society (GI) and member of IFIP WGs 3.1 & 3.3 and national German representative to TC3.

Informatics Education based on Solving Attractive Tasks through a Contest

Valentina Dagiene[1], **Gabriele Stupuriene**[2]
Faculty of Mathematics and Informatics
Vilnius University
[1]Naugarduko str. 24
LT-03225, Vilnius
valentina.dagiene@mii.vu.lt
[2]Akademijos str. 4
LT-08663 Vilnius
gabriele.stupuriene@mii.vu.lt

Abstract: The paper discusses the issue of supporting informatics (computer science) education through competitions for lower and upper secondary school students (8–19 years old). Competitions play an important role for learners as a source of inspiration, innovation, and attraction. Running contests in informatics for school students for many years, we have noticed that the students consider the contest experience very engaging and exciting as well as a learning experience. A contest is an excellent instrument to involve students in problem solving activities. An overview of infrastructure and development of an informatics contest from international level to the national one (the *Bebras* contest on informatics and computer fluency, originated in Lithuania) is presented. The performance of *Bebras* contests in 23 countries during the last 10 years showed an unexpected and unusually high acceptance by school students and teachers. Many thousands of students participated and got a valuable input in addition to their regular informatics lectures at school. In the paper, the main attention is paid to the developed tasks and analysis of students' task solving results in Lithuania.

Keywords: Informatics Education, Computer Science Education, Tasks, Tests, Contest, Problem Solving, Cognitive Skills, Bloom's Taxonomy

1 Introduction

Competition makes teaching of informatics (computer science, computing) more attractive for children. During contests students have the possibility to test their skills among peers from different schools or even countries and to make friends in a field that they have interests. The contest on informatics and computer fluency named *'Bebras'* (it is a Lithuanian word for 'beaver') may be the key to the potential of informatics science knowledge and an attractive way to bind up technology and education.

Bebras is an international initiative whose goal is to promote informatics and computational thinking especially among teachers and students of all ages, but also to the public at large. The big challenge of *Bebras* is to organise easily accessible and highly motivating online contests in many countries. The contest was established in 2004 by Lithuanian suggestion (Dagiene, 2006). It is involving massively growing numbers of students and countries. Lithuania celebrated its 10th year's anniversary of running the *Bebras* contest in November last year. Since 2004, the *Bebras* contest has quickly spread across Europe and now is a really international motion. Overall, more than 0.7 million students participated in the Bebras contest in 2013 (Table 1).

The *Bebras* contest is design to promote informatics fundamentals for both boys and girls and equally attract their attention. The result is quite good: quite a big number of girls have taken part in last year's contest; some countries even have equal or almost equal participants of both genders (Italy, Japan, Taiwan, see Table 1).

Table 1: Numbers of participants distrusted by country and gender in 2013 contest

Country	Total	Girls	Boys
Austria	**12 154**		
Belgium	**848**		
Bulgaria	**551**	188	636
Canada	**4 229**		
Czech R.	**34 454**	15 386	19 068
Estonia	**3 517**		
Finland	**4 423**	1 846	2 577
France	**171 932**		
Germany	**206 430**		
Hungary	**6 246**		
Ireland	**3 141**	1 375	1 470
Italy	**3 288**	1 644	1 644
Israel	**~2000**		
Japan	**4 371**	2 082	2 289
Latvia	**1 038**	434	604
Lithuania	**25 909**	10 817	15 092
The Netherlands	**12 592**		
New Zealand	**217**		
Poland	**15 933**	11 534	4 399
R. of South Africa	**1 111**		
Russian F.	**17 584**	8 203	9 381
Slovakia	**55 017**	24 217	30 800
Slovenia	**12 040**	5 152	6. 36
Spain	**711**		
Sweden	**1 869**	695	1 446
Switzerland	**9 832**		
Taiwan	**9 526**	4 842	4 684
Ukraine	**86 266**	41 077	45 189
United Kingdom	**21 473**		

In Lithuania, similarly to other participating countries, we strive to implement the contest as a nationwide and efficient event for sending the message about informatics to students and teachers. Under agreements of the involved countries, the second week of November is announced as a *Bebras* week each year.

The contests are made of a set of short questions or tests usually called *Bebras* tasks. These tasks can be answered without prior knowledge about informatics, but are clearly related to fundamental informatics concepts. To solve those tasks, students are required to think in and about information, discrete structures, computation, data processing, data visualisation, but they also must use algorithmic as well as programming concepts. Each *Bebras* task can both demonstrate an aspect of informatics and test the talent of the participant, regarding understanding of informatics.

The *Bebras* initiative is based on two main events: 1) an international workshop which takes place between May and June and is organised in order to discuss the task set for the coming contest; and 2) national contests organised in all participating countries in autumn during the *Bebras* week. Additional activities take place around those two main events. Many countries run a second round for the *Bebras* contest, some countries organise *Bebras*-tasks training workshops for teachers or summer camps for students. Many more activities are set within countries all through the year: participants' awarding celebration, seminars about Informatics concepts, collecting data and writing research papers, etc.

The main aim of the paper is to give a general overview of students' performance in the *Bebras* contest of 2013 in Lithuania and discuss how students (including primary) and upper secondary education cope with it.

2 Contest as a Promoter of Informatics Education

The *Bebras* contest is organised by each participating country locally (Dagiene, Futschek, 2008). Usually there are national committees or organisations established which aim to run the *Bebras* contest. For running the contest, countries are using different technologies mainly based on online contest management systems.

Each country chooses tasks from a *Bebras* task pool approved by the annually organised international *Bebras* task workshop. There are however some mandatory tasks that all countries are obliged to use. There are different task sets for different age students. Five age groups have been used (Table 2).

Table 2: Age groups

Groupe name	Grade, age	Comments
Mini (Little Beavers)	3 and 4, age 8–10	Only few countries have this group: Czech Republic, Finland, Lithuania, Poland, Slovakia, Sweden
Benjamin	5 and 6, age 11–12	Some countries have merged Benjamins and Cadets
Cadet	7 and 8, age 13–14	
Junior	9 and 10, age 15–16	
Senior	11 and 12, age 17–19	Some countries have grade 13 as well

Some countries have been using slightly different distributions of groups. For example, Estonia has run the contest in three age groups: grades 6, 7, and 8 are used for cadets, 9 and 10 for juniors and the rest for seniors. In Lithuania we have all five age groups as it is shown in Table 2. Most participants are from grades 5 to 9, the other grades have a lower number of participants (Fig. 1).

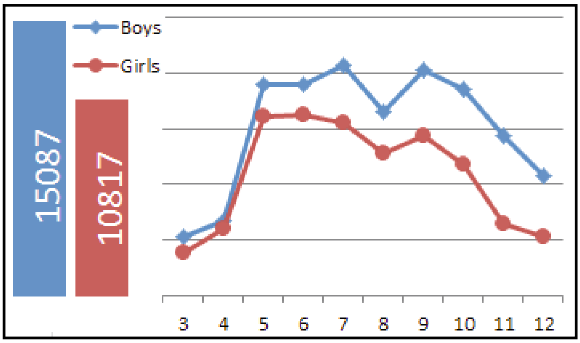


Figure 1: Numbers of contestants distributed by grades (from 3rd to 12th) in Lithuania in 2013

Running contests, however, is used essentially to attract students and teachers. The fundamental goals are to promote informatics as a science among youth, to show how fascinating it is, to think about and to solve informatics problems, and to demonstrate that, on principle, informatics is approachable by everyone. The central tools to achieve these goals are the *Bebras* tasks. Not only are they used in the contests, but also spread among teachers in order to provide them with a wealth of teaching items that can flexibly be incorporated in informatics lessons, school-wide informatics promotional activities or any other occasion to show the attraction of informatics in an entertaining way.

The *Bebras* contest essentially focuses on informatics concepts. Understanding and handling the basics and foundations of informatics is more important than knowing technical details. The use and interpretation of results comes prior to being able to prove results. Controlling computations, calculations and estimations is more significant than being able to do computations by ourselves. A computer has to be understood at many levels, including: as a fundamental culture item and not as a collection of buttons and instructions; as a development of ideas and not a finished work; as an explanation of the concepts, etc. All these topics we keep in mind while organizing contests and working on task preparation.

The informatics curricula in Lithuanian lower and upper secondary schools, the evaluation schemes and even the denominations have been changed; nevertheless Informatics has remained a separate subject, now called "information technologies (IT)". Besides, one of the most important components of IT is to make students of comprehensive schools digitally literate. In Lithuanian lower secondary schools the IT courses are compulsory for the 5th–10th grades (student age 12–17 years) for approximately 1 hour per week, respectively 35 hours per year. There are some optional modules as well (e.g. a programming module in grade 9 or 10). Students of upper secondary schools (11th and 12th grades) can choose advanced optional modules and have to learn the content defined in the course curriculum.

However, there is no common international agreement on an accepted framework for informatics and information technologies courses in general education, although there are several discussions on this issue (Dagiene, Futschek, 2010; Micheuz, 2008; Hromkovic, 2006; Micheuz, 2005; Schubert, 2004). However a number of key concepts arise repeatedly in informatics: languages, machines, and computation; data and representation; communication and coordination; abstraction and design; the wider context of computers (Computing at School Working Group, 2012).

Almost a common opinion is that fundamentals of algorithms and programming are the key concepts in school informatics education. Then, what concepts should we include in informatics education apart from algorithms and programming? What is the ratio of programming concepts and information technology concepts and their application?

The basic concepts of informatics are mentioned in many scientific papers but they are not well defined or commonly accepted. There exist attempts to define the more powerful term "fundamental idea" as an educational principle. Fundamental ideas fulfil the four criteria of the paper (Schwill, 1997):

- Horizontal criterion (applicable in multiple ways in different areas)
- Vertical criterion (may be learned on every intellectual level)
- Criterion of time (observable in the historical development and will be relevant in the longer term)
- Criterion of sense (meaning in everyday life and related to ordinary language)

A. Schwill identified three fundamental master ideas within the software development life cycle: algorithmization, structured dissection and language. In the context of our contest we use the term concept of informatics since we can involve in our short tasks only aspects of fundamental ideas. But we have the four criteria for fundamental ideas in mind to create tasks that involve concepts that are hopefully interesting for a long term, can also be understood without too much pre-knowledge, can be used also in other areas and can be understood at different intellectual levels.

3 Bebras Tasks for Transmission of Informatics Concept to Learners

Interesting, attractive tasks on informatics concepts are crucial for *Bebras* contests. About 200 new challenging tasks are needed each year. Teachers should learn how to explain what is behind one or another *Bebras* task. Also teachers should learn how to develop *Bebras* tasks. So for workshops and conferences the target groups are teachers.

Each countrywide contest is a collection of small, interesting questions that can be answered without prior knowledge about informatics, but are clearly related to informatics concepts and require thinking in and about informational, discrete structures as well as algorithmic, programming concepts.

The key idea behind each task presented to contestants is not to ask for already learned facts but to give problems that allow students to learn something about concepts (on informatics, computer science, computing) that may be new for them.

Every year, new *Bebras* tasks are developed in a cooperative effort of all countries involved: the *Bebras* international Task Workshop. Each country provides a set of task proposals, and the whole pool of proposals is then discussed at the annual International Task Workshop. There, proposals may be rejected, refined, or simply accepted for use in that year's *Bebras* contests. A task pool is the result of this workshop. The national organisers make up their national task set from this pool. However, at the workshop, a subset of the task pool, which has been growing over the years, is determined to be "mandatory" and hence is used in all national *Bebras* contests.

When preparing for the actual year of the contest we drew on the characteristics of appropriate tasks from (Dagiene, Futschek, 2008). To be able to deeply analyse students' solutions and properly interpret resulting observations, we have developed the following six task types:

- ***Information***: conception of information, its representation (symbolic, numerical, graphical), encoding, encrypting;
- ***Algorithms***: action formalization, action description according to certain rules;
- ***Computer systems and their application***: interaction of computer components, development, common principles of program functionality, search engines, etc.;
- ***Structures and patterns***: components of discrete mathematics, elements of combinatorics and actions with them;
- ***Social effect of technologies***: cognitive, legal, ethical, cultural, integral aspects of information and communication technologies;
- ***Informatics and information technology puzzles***: logical games, mind maps, used to develop technology-based skills.

The descriptions of these task types also involve concepts of informatics although this was not the goal of this classification. It gives anyway a rough idea what kinds of problems and what topics of computer science we have in mind for *Bebras* contests.

In the short *Bebras* tasks we can include concepts of informatics like algorithms and programs: sequential and concurrent; data structures like heaps, stacks and queues; modelling of states, control flow and data flow; human-computer interaction; graphics; etc. Using a proper problem statement nearly all aspects of computer science and ICT can be a topic of a *Bebras* task.

While analysing students' solutions of the Slovakian contest in 2009, Kalas and Tomcsanyiova have proposed a new categorization of tasks into four components of informatics education (Kalas, 2009):

1. **Digital literacy**
   - Basic knowledge and concepts of informatics and computers
   - Computer literacy, working with applications
   - Ethical and legal issues, security, history of computing and informatics
2. **Programming**
   - Formal description of a solution, process, behaviour, progress
   - Understanding, analysing, interpretation and assembling such descriptions
   - Algorithms, algorithmic thinking
3. **Problem solving**
   - Logical reasoning, justification, argumentation
   - Puzzles, riddles, problems
   - Strategies for problem solving
4. **Data handling**
   - Representations, coding, patterns, structures
   - Mathematical basics of informatics, combinatorics
   - Data and data structures, information and data processing

The quality of tasks is crucial for the success of all task-based competitions. The tasks must reflect the goals of the competition and should be adequate to the applicants. In educational competitions, the tasks should attract students and drive them to learn and explore as well as to develop skills in the particular area.

When teaching informatics through problem solving, it is very important to choose interesting tasks. Therefore, one should try to present problems from various areas of science and life, with a lot of data. Processing large amounts of data becomes one of the most important aspects when learning programming.

The cognitive domain involves knowledge and the development of intellectual skills (Bloom, 1956). This includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of intellectual abilities and skills. There are six major categories, starting from the simplest behaviour to the most complex. The categories can be thought of as degrees of difficulty.

Let us analyse a task set used in the *Bebras* contest last year in regard to the informatics concepts. Each task is characterised by main informatics concepts, which are included with the aim to bring them to the students (see Annex I). We have classified what kind of cognitive skills must be applied by students for solving each *Bebras* task. Task classification is based on Bloom's revised categories (Anderson et al., 2000) and on Kalas' developed schema

(Kalas, 2009). We label tasks using a first letter according to age groups: M for Mini (Primary), B for Benjamins, C for Cadets, J for Juniors, and S for Seniors. Some tasks were used in more than one age group; these tasks have several letters. *Bebras* tasks are spread in all revised Bloom's categories; most tasks are in domains of Understanding, Applying, Analysing and Evaluating (Table 3).

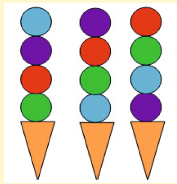
Table 3: Classification of tasks used in Lithuanian Bebras contest 2013

Cognitive skills applied	**Tasks**
Remembering general facts, basic concepts	M2; B7
Understanding (simple) given language and commands, comprehending the meaning	M11+B5; B12+C5; M3; M4; J12
Understanding (complex) description of processes, rules of behaviour and methods	M6+B2; M17; B6; C11; M7; B11; B15; J11; S14; S21
Applying given generative rule(s) or method(s) to an initial state, input or situation	1; M5+B1; B17; C12+J3; C14+J5; C21+J15; B9; M13; M18+C4; B21+S8; S20
Applying – interpret given instructions or program	M10+B4; B19+C9; C19+J9; J21+S13; M14; B14; J17+S15; S7; S17
Analysing situation and processes	M9+B3+C2; J13+S5; J19+S11; B10; S6
Analysing – matching several descriptions with several behaviours	M12+B8+C3; C15; J18+S10; M16; C6; C17+J6; S18
Evaluating – comparing different situations or solutions by certain criterion	C20+J10+S4
Evaluating – deducing possible result, final state or final product	M8+C1; B16; B20+C10+J8+S2; C7+J2; C13+J4; J16; J20+S12; B13+C8+J1+S1; C16; S16; S19
Creating - compiling information together	B18; J14+S9; M15; C18+J7+S3

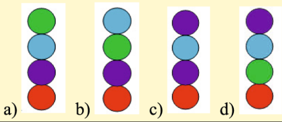
So that the reader could better understand our conception of informatics education and also the analysis of the tasks offered in the Annex, herein we present complete wordings of three tasks. The first of them fits into programming and was solved by both Mini and Benjamin groups; the second one belongs to data handling and operation abstraction and was mandatory for all age groups except the youngest (Mini); the third one was assigned to Juniors and Seniors and focuses on top-down analysis.

Task 1: Ice cream machine

The ice cream machine always put scoops in the same order. In the picture (on the right), you can see three examples for it:



Which order could come from the machine?



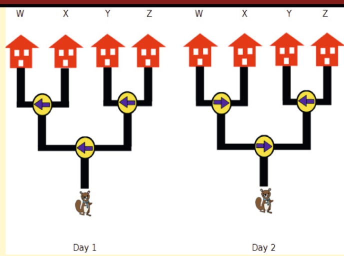
Task 2: Spinning toy

Beavers discovered a piece of wood into which worms had made a system of tunnels and pits. A handy father used it to make a toy.
To start we put a marble in the middle. The goal is to get the marble out by turning the wheel to the left (L) and right (R). By each turn the marble either runs to the next pit or (at the end) out of the wheel.

By which of the following sequences will the marble reach the exit?
a) LRRLR b) RLRLL c) LRRLRL d) LRRRRL

Task 3: Visiting friends

Mr. Beaver has 4 friends living in different villages, and he plans to visit one of these friends every afternoon. Initially, all arrows point to the left road. When passing the intersection, Mr. Beaver would switch the arrow to the opposite direction. For example, on day 1, Mr. Beaver takes the road on the left at the first intersection, takes the left road on the second intersection, and reaches Village W. On day 2, Mr. Beaver turns right at the first intersection, then left at the second intersection, and he arrives in Village Y.



Which village will Mr. Beaver visit on day 30?
a) Village W b) Village X c) Village Y d) Village Z

Figure 2: Three task examples taken from the *Bebras* contest 2013

4 Analysing Solutions of Contestans

We have studied the differences among the informatics tasks at the level of cognitive skills, which students had to apply while solving them. We will show the results of different age and gender groups. We analysed the data, which we obtained before and during the contest. We recorded which tasks were solved by each student and which of four given choices they indicated as correct.

Last year 25909 students took part in the contest in Lithuania, out of them 2176 Mini, 7022 Benjamins, 6550 Cadets, 6490 Juniors, and 3671 Seniors. Figure 4 shows total numbers of boys and girls in these age groups, together with the distributions of their total scores. Horizontal axis represents all possible scores (between 0 and 90 for Mini group, and between 0 and 105 for all others); vertical axis represents numbers of boys and girls who got corresponding score.

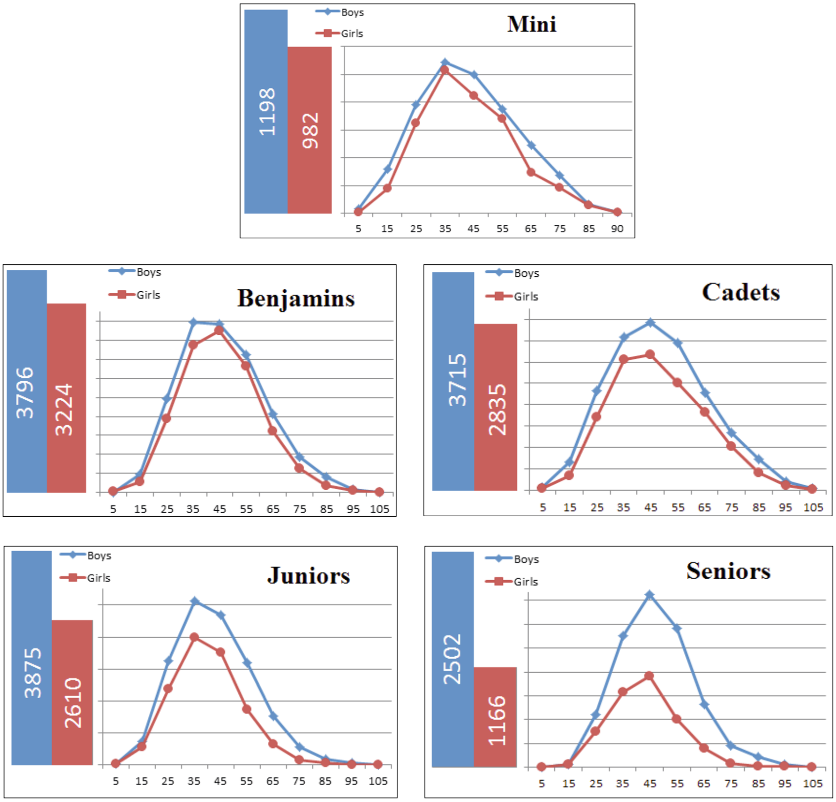


Figure 3: How successful were boys and girls when solving tasks – the distributions of their scores in all five age groups

We consider the total scores excellent: 62.35 % of contestants got more than 1/3 of the points; 22.92 % of contestants got more than half of the points; 5 % got more than 2/3 of the points; 0.33 % got more than 90 points (or 80 for the Mini group). This proves that the main goal of the event was accomplished – to provide an attractive opportunity to deliver informatics education to a group of students as wide as possible, without any preference of any particular group(s).

Attendance of the girls in these two categories significantly exceeded our expectations. As we can see from the charts the girls of all groups except

Senior are doing very well with minimal difference between boys and girls. In our opinion these results disprove the misconception that informatics is a boyish subject.

Figures 5 and 6 show the dependency of the number of correct and incorrect solutions (separately for boys and girls) on student age for tasks presented in section 3.

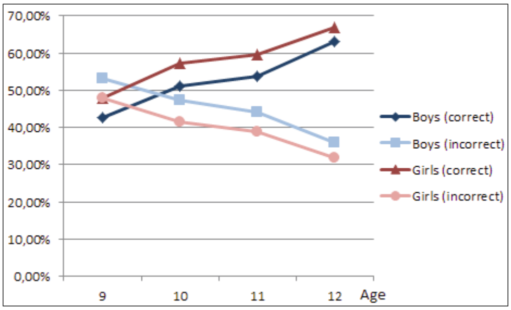
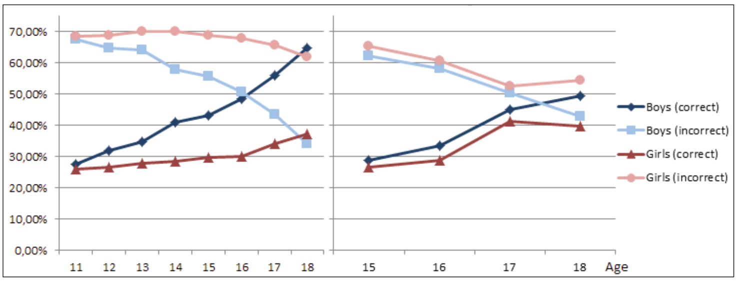


Figure 4: Distribution of solution of the task “Ice cream machine”



Figures 5: Distribution of solutions of the tasks “Spinning toy” (left) and “Visiting friends” (right)

Figure 5 shows that girls of the ages 9 to 12 did slightly better than boys for the task “Ice cream machine”, their scores are better and there are less wrong answers (guessing). However the task “Spinning toy” was exceptionally hard for girls of all ages from 11 to 18. Why? This task requires deep abstract thinking and imagination. Abstraction is one of the main three components of computational thinking. Our schools should focus more on developing abstract thinking of students and especially girls.

The task “Visiting friends” is very hard for both boys and girls (Figure 6 (right)). In order to solve this task students need to be able to do top-down analysis and observe the periodicity from the simulation, also abstraction thinking is needed.

5 Conclusion

Competitions play an important role as a source of motivation students to learn informatics (or computer science, or computing) in a non-formal way. Our ten-year experience running the *Bebras* contest has shown that both students and teachers can gain deeper skills and understanding of informatics concepts. Well-organized informatics contests with conceptual-based, exciting, playful tasks invite students to use computer reasoning and to explore understanding of technology.

The international task workshop is organized annually for developing informatics tasks and producing a task pool, from which each country is obliged to choose tasks for their national contest. Preparation and selection of tasks are very important processes. Lithuania is using the same task set as Austria, Germany, Switzerland, The Netherlands, and almost overlapping with tasks in Finland and Sweden.

It is not easy to estimate how difficult a task will be for a particular age group when developing the task. Our analysis has shown that last year's task set was balanced well enough at least for Lithuanian students: we got a distribution of scores very close to the normal distribution (the Bell curve). A few students do very well and a few do very poorly. A bunch of scores end up clumped around the mean score.

The large and multifaceted data collected in the *Bebras* contests make it possible to analyse many interesting aspects related to e.g. students' understanding, difficulties and misconceptions based on different factors. In this paper, we have looked into tasks and assign them to cognitive skills domains according to the revised Bloom's taxonomy. We found that the *Bebras* tasks are well-balanced according the cognitive skills' domains: at the most tasks are in the high categories Understanding (15), Applying (20), Analysing (12) and Evaluating (12).

An international contest on informatics *Bebras* involves more than twenty countries, cultures and languages. Clearly, these are all factors that make it challenging to create unambiguous and clear tasks.

Acknowledgements

The authors want to explicitly thank all members of the international *Bebras* contest on informatics and computer fluency community that took part in task development and influenced in this way the outcome of this paper.

References

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., … Wittrock, M. C. (2000). A Taxonomy for Learning, Teaching, and Assessing: A revision of Bloom's Taxonomy of Educational Objectives. *New York: Pearson, Allyn & Bacon.*

Bloom, B. S. (1956). *Taxonomy of Educational Objectives*. Handbook I: The Cognitive Domain. New York: David McKay Co Inc.

Computing at School Working Group. (2012). *A curriculum framework for Computer science and Information Technology*. Retrieved from http://www.computingatschool.org.uk

Dagienė, V. (2006). Information technology contests – introduction to computer science in an attractive way. *Informatics in Education,* 5(1), pp. 37–46.

Dagienė, V., Futschek, G. (2008). Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. *LNCS, 5090*, pp. 19–30.

Hromkovic, J. (2006). Contributing to General Education by Teaching Informatics. *LNCS, 4226*, Springer, pp. 25–37.

Kalas, I., Tomcsanyiova, M. (2009). Students' Attitude to Programming in Modern Informatics. *Informática na Educação: teoria & prática, Porto Alegre*, 12(1), pp. 127–135.

Micheuz, P. (2005). 20 Years of Computers and Informatics in Austrian's Secondary Academic Schools. *LNCS, 3422*, pp. 20–31.

Micheuz, P. (2008). Harmonization of Informatics Education – Science Fiction or Prospective Reality? LNCS, 5090, *Springer*, pp. 317–326.

Schubert, S., Taylor, H. (eds.). (2004). Secondary Informatics Education. *Special issue of Education and Information, Technologies*, 9(2), Kluwer Academic Publishers, Boston.

Schwill, A. (1997). Computer science education based on fundamental ideas. In Proceedings of Information Technology – Supporting change through teacher education. Edited by D. Passey and B. Samways. *Chapman Hall*, pp. 285–291.

Biographies

Valentina Dagienė is a professor at the Vilnius University Institute of Mathematics and Informatics and head of the Department of Informatics Methodology. She has published over 150 research papers and the same number of methodological works, has written more than 60 textbooks in the field of informatics and information technologies for secondary schools. She has been working in various expert groups and working groups, organizing the Olympiads in Informatics among students.

Gabriele Stupuriene is a young researcher in the Vilnius University Institute of Mathematics and Informatics at the Department of Informatics Methodology. She is involved in a National project on localisation of education software (*Mahara, Claroline*). Since 2010 she has been working on Informatics contest *Bebras* tasks. She has developed and defended her Master thesis "Conceptualisation of Informatics Fundamentals through Tasks" in 2011.

Annex

Short number	Idea from	Title	What student can learn from the task
M1[5]	Slovakia	The Necklace Machine	algorithm; programming; sequence; repeat; pattern
M2	Slovakia	Tools	understanding a tool
M3	Slovakia	Beings	logics
M4	Slovakia	Train	logics
M5+B1	Slovakia	In the Forest	finding a path; graph; tracing; finding a solution backwards
M6+B2	Hungary	Ice cream machine	detecting an algorithm; machine work; loop
M7	Lithuania	Towns	graph theory
M8+C1	Czech R.	Rotation tool	understanding a tool, what it is able to do and what not, rotate a tool, transformation
M9+B3+C2	Canada	More Candy	longest common subsequence; dynamic programming
M10+B4	Slovakia	Bee Hive	algorithm; robot navigation; follow sequence of instructions
M11+B5	Slovakia	Jeremy in the Bushes	algorithm; robot navigation; tracing
M12+B8+C3[6]	**Russia**	**Balls Trigger**	logics; trigger; logical gate
M13	Bulgaria	Follow the squirrel	turning; instructions; sequences of instructions
M14	Slovenia	Labyrinth	route planning
M15	Latvia	The making of a panoramic view picture	panorama view; puzzle
M16	Russian	Beavers in an elevator	optimization problem
M17	Slovakia	Ladybug Dotty	program; condition; tracing
M18+C4	Germany	Loading trucks	optimization
B6	Japan	Drumming	iteration; repetition; loops; following instructions
B7	Germany	Homework	e-mail etiquette
B9	Lithuania	Cities	representation of information: linking several types of information
B10	**Japan**	**Zebra Tunnel**	to follow instructions; algorithm analysis; data structures: FIFO (queue) and LIFO (stack)

5 Underline font indicates interactive task.

6 Bold font indicates *Bebras* mandatory tasks which must be included by all countries in their contests.

B11	France	Swapping	implicit, directed, graph
B12+C5	Sweden	The importance of an instruction	instruction; human machine instruction
B13+C8+J1+S1	**Japan**	**Signal Fire**	graphs; shortest path problem; breadth-first search
B14	Lithuania	Taking pictures	panorama view
B15	Sweden	Frog trouble	shortest path; breadth-first search
B16	Austria	The takeaway	memory; management of data structure; stack
B17	Belgium	Rescue action	tree traversal; recursive definition; optimisation problem
B18	Germany	Soda Machine	finite stet automata; coding
B19+C9	Slovenia	The Highest Tree	search algorithm; local optimisation; global optimum
B20+C1+J8+S2	**Slovenia**	**Spinning Toy**	binary tree representation; tree traversal; operations abstraction
B21+S8	Switzerland	Build the bridges!	minimum spanning tree, Kruskal's algorithm, Prim's algorithm, graph theory
C6	Slovenia	Gossiping	graph theory
C7+J2	**Slovenia**	**Necklace**	shortest path to reach the end
C11	Hungary	Gift boxes	algorithm; recursion; breaking the problem down into smallest problems
C12+J3	**Austria**	**Airport**	applying rules; structure; scheduling; limited resources
C13+J4	Japan	Bebras Rowing	binary number; bit; numeral system
C14+J5	Austria	Helping grandpa beaver creating his password	e-mail; security; password enforcement; applying rules
C15	Netherlands	Triangle code	encryption; decryption; description algorithm
C16	Canada	Putting people in line	Bubble-sort; sorting techniques; algorithm running time
C17+J6	France	Sort by weight	sorting algorithm
C18+J7+S3	Germany	Movie seating	graph theory; optimal; relation
C19+J9	France	Beaver the hobbit	graph; shortest paths; brute force approach
C20+J10+S4	Switzerland	Serial Transmission	RS232; serial transmission; bits; bytes
C21+J15	Switzerland	Flowchart computing	flowchart; computer program representation; visualization
J11	Japan	Storehouse	Binary search
J12	**Slovakia**	**Dice**	following a list of commands; procedure; imperative programming

J13+S5	Switzerland	Domino circles	Eulerian path; graphs; largest Eulerian subgraph; modelling graph
J14+S9	Germany	Random Pictures	computer graphics; non-determinism; programming; variables
J16	Japan	Shortest Path	division a task in smaller parts; dynamic programming
J17+S15	Netherlands	Turn the cards	logic reasoning implication
J18+S10	**Netherlands**	**River inspection**	algorithm; flow problem; planar directed graph; maximal cut; sweeping line
J19+S11	**Taiwan**	**Visiting Friends**	counting; top-down analysis; modulo operations; patterns; observing the periodicity from the simulation
J20+S12	Austria	No turning left!	graph; shortest path; algorithm; determine a path with minimum effort
J21+S13	Austria	From A to C	perform instructions; algorithm
S6	Taiwan	Delicious Dinner	job scheduling
S7	Austria	Apple in the basket	patterns; invariants
S14	Netherlands	Treasure hunt	Binary search, divide and conquer
S16	Belgium	Old computing machine	programming; assembly language; abstraction
S17	Germany	Colored Necklaces	syntax diagrams
S18	Netherlands	Hotel key	encoding; combinatorics
S19	Belgium	The magic machine	Petri net; graph; algorithm
S20	Italy	Beaver Student back home	algorithms; constraints; programming
S21	Latvia	Raid arrays	Raid array; data redundancy

Student Perspectives of Social Networking use in Higher Education

Jason Brent Ellis[1], **Carla Reis Abreu-Ellis**[2]
[1]Department of Foundations and Instruction
Ashland University
Ashland, Ohio, United States of America
jbellis@ashland.edu
[2]Department of Inclusive Services and Exceptional Learners
Ashland University
Ashland, Ohio, United States of America
cellis1@ashland.edu

Abstract: Social networks are currently at the forefront of tools that lend to Personal Learning Environments (PLEs). This study aimed to observe how students perceived PLEs, what they believed were the integral components of social presence when using Facebook as part of a PLE, and to describe student's preferences for types of interactions when using Facebook as part of their PLE. This study used mixed methods to analyze the perceptions of graduate and undergraduate students on the use of social networks, more specifically Facebook as a learning tool. Fifty surveys were returned representing a 65 % response rate. Survey questions included both closed and open-ended questions. Findings suggested that even though students rated themselves relatively well in having requisite technology skills, and 94 % of students used Facebook primarily for social use, they were hesitant to migrate these skills to academic use because of concerns of privacy, believing that other platforms could fulfil the same purpose, and by not seeing the validity to use Facebook in establishing social presence. What lies at odds with these beliefs is that when asked to identify strategies in Facebook that enabled social presence to occur in academic work, the majority of students identified strategies in five categories that lead to social presence establishment on Facebook during their coursework.

Keywords: Social, networks, higher, education, personal, learning, environments, Facebook

1 Background and Literature Review

Technologies are present in the in-class and out-of-class experiences of students more than they have ever been before (Public Broadcasting System, 2013). In the fall of 2008, the National Center for Educational Statistics (NCES), in the Institute of Education Sciences, conducted a survey. Questionnaires were mailed to 2.005 public schools in the 50 states and the District of Columbia; a response rate of 79 % was returned. “The survey weights were adjusted for questionnaire nonresponse and the data were then weighted to yield national estimates that represent all regular public elementary and secondary schools in the United States” (National Center for Educational Statistics, 2010, p. 2). At this point in time, the NCES found that the ratio of students to handheld devices in schools was one device (which included Palm OS, Windows CE, Pocket PC, BlackBerry) per 21 students nationally, whereas the number of mobile laptops (for distribution without a fixed location) were found to be one device per 14 students nationally (National Center for Educational Statistics, 2010). In the past five years schools have been looking at one-to-one computer initiatives, where schools loan equipment to students for a semester, year, or more or a “bring your own device” initiative where it is hoped that the majority of students will bring in their own purchased, leased, or loaned equipment whether it be a laptop, tablet, or handheld device. In 2013 the Public Broadcasting System’s (PBS) LearningMedia division published national survey results of 503 teachers, finding that:

> A growing number of educators have access to and are adopting new technologies and platforms to support instruction. Ninety percent of teachers surveyed have access to at least one PC or laptop for their classrooms, and six in 10 teachers (59 %) have access to an interactive whiteboard. Tablets and e-readers saw the biggest increase among technology platforms available for classroom instruction. More than one-third (35 %) of teachers said they have access to a tablet or e-reader in their classroom, up from 20 % a year ago. Among teachers with access to tablets, 71 % cite the use of educational applications as the most beneficial for teaching, followed by educational websites (64 %) and educational e-books/textbooks (60 %). (Public Broadcasting System, 2013, para 5)

When asked about the use of handheld devices, including cell phones and smart phones, 36 % of teachers surveyed, responded that they were available in the classroom (Public Broadcasting System, 2013).

There have been two camps of thought on cell phone use in schools; turn it off or turn it in or use it for educational means when directed to do so (Prensky, 2008). The reality of what actually happens in schools on a daily basis may be a little more nebulous. Charles (2012) performed a qualitative study composed of classroom observations and interviews with high school youth and teachers on the use of technological devices in schools and concluded that "schools and teachers set rules and protocols that define appropriate behaviors with social digital tools and discourses. Nevertheless, students and teachers frequently negotiate the boundaries through relationships founded on trust and respect" (p. 15).

When it comes to the use of technology for social networking there are other substantial barriers that schools need to contend with. The press has been particularly adept at discussing the use of social networks in schools as synonymous with cyberbullying (Topping, Coyne, 2013), athletes losing National Collegiate Athletic Association (NCAA) eligibility (Federico, 2013), and teachers having inappropriate communications and interactions with students via social networks (Matthews, 2012). As such, districts, and at times states, have created policy and laws to persuade disuse of social networking in K-12 schools. For instance, Varlas (2011) observed that:

> Schools and districts are getting noticed for what they don't allow. Two common practices – blocking sites and restricting teacher-student social media contact – have made headlines lately. For example, Missouri's Senate Bill 54 (or the Amy Hestir Student Protection Act, named for a student who was repeatedly victimized by a teacher on social media) prohibits direct social media contact between teachers and students, unless it's deemed appropriate, education-related contact in a public setting. S.B. 54 takes the common "no 'friending'" policy a step further by applying it to both current and former students, indefinitely. (para 16)

It is expected that these safety and privacy related tensions will remain in K-12 education, with the only real possibility for remedy lying with centralized district monitoring and through closed or dedicated social networks.

Higher education, on the other hand, allows for more freedoms in terms of teacher-student interactions, including communications via social networks, since most of the students are legally adults (over 18 years of age in the USA).

What is perhaps more interesting is that the demographic of social network user is getting older, so the use of social networks by students across the age spectrum is more congruent than previously thought. Breener and Smith (2013) reported on their study sponsored by the Pew Research Center's Internet & American Life Project that online adult's social networking use has grown substantially since 2005. It was reported that currently 72 % of adults use social networking sites. Further, they noted that:

> Although younger adults continue to be the most likely social media users, one of the more striking stories about the social networking population has been the growth among older internet users in recent years. Those ages 65 and older have roughly tripled their presence on social networking sites in the last four years – from 13 % in the spring of 2009 to 43 % now. (para 1)

But then how are these users using social networks in higher education? The concept of social presence in online learning may hold the first clue.

Social presence by any means is not a new idea. In fact, we as human beings have been referred to as social animals. Aristotle's (350 B.C.E.) Politics observed that, "A social instinct is implanted in all men by nature" (para 14). It would not be too far of a stretch to think that since we seek out social interaction in everyday life that we want to do the same thing while learning. Social learning theory has a long and entrenched place in education. In the 1970s theories from social psychology and developmental theory came forth with the idea is that students can learn from each other while developing. Bandura (1977) noted that children could learn from observation and modelling behaviours from others. Further, Vygotsky (1978) coined the term Zone of Proximal Development, which explained that a child would be able to accomplish a task independently in the future if he/she receives assistance performing it in the present. If we take this base premise as valid, which has been illustrated through research since that time and bring it to the current discussion we are left with the question of how then do students learn from each other, and how does that mix with online social presence?

In online interactions, social presence is defined as "'a sense of being with another' in the virtual environment" (Biocca et al., 2003, p. 460). Thus, social presence "acts to 'humanise' the experience of online learning" (Kehrwald, 2010, p. 48). Tu and McIsaac (2002) theorized that there were three dimensions of social presence; social context, online communication, and interactivity. They proposed that these three components were integral to create a sense of community among online learners. They further observed that, "an increase in the level of online interaction occurs with an improved level of social presence" (p. 131). Jeremić, Milikić, Jovanović, Brković and Radulović (2012) took the notion further by discussing the interaction social presence has with the adaptability of Personal Learning Environments (PLEs):

> The notion of PLE assumes personal selection and aggregation of different, often web-based tools and services into a learning environment customized to the needs and preferences of an individual learner. In a PLE, learning activities are not confined within the "walls" of one system/tool, thus enabling learners to make use of a wide diversity of digital resources (content, tools, and services) available on the Web (p. 28).

Abreu-Ellis et al. (2013) observed similar findings when using Computer Mediated Communication (CMC) for language acquisition in that "participants noted that they tended to migrate to communication technologies they believed that their peers would frequently use or check; for instance using Facebook for urgent communications when they needed to reach their teletandem partner rather than e-mail (for fear their partner would not check for messages in a timely manner)" (p. 366).

Jeremić, Milikić, Jovanović, Brković and Radulović (2012) further clarified the relationship between PLE's and online or social presence in that:

In a PLE, the notion of global online presence, i.e., student's online presence expressed on different tools integrated into his/her PLE, could be especially important. By giving students insights into their class-mates' activities, availability for chat, information about work overload, emotional state, likes and dislikes, and all of that regardless of the particular tool they are using in the given moment, students' global online presence can provide those missing nonverbal cues typical for face-to-face interaction. This further increases students' awareness of each other and positively affects their willingness to collaborate (p. 28).

This study sought to observe how students perceived PLEs, what they believed were the integral components of social presence when using Facebook as part of a PLE, and to describe student's preferences for types of interactions when using Facebook as part of their PLE.

2 Methodology

This study used mixed methods to analyse the perceptions of graduate and undergraduate students on the use of social media, more specifically Facebook, as a learning tool. Participants selected to participate in this study were students who had been required to join a closed Facebook group managed by the researchers in their respective classes. This does not however, screen out the fact that participants may have reflected on their use of social networks or Facebook used outside of those classes under the direction of other professors. An online survey was developed and an introduction letter with the survey link was sent by e-mail to 77 undergraduate and graduate students who were enrolled, at a four-year private university located in central Ohio. Researchers used the university Learning Management System (LMS) to contact current and previous students and to request their participation in the study. Fifty surveys were returned representing a 65 % response rate. Survey questions included both closed and open-ended questions.

Survey data was analyzed in terms of frequencies and correlations. Additionally, a content analysis of narrative responses was performed in order to identify recurring themes. Patton (2002) noted, “content analysis is used to refer to any qualitative data reduction and sense-making effort that takes a volume of qualitative material and attempts to identify core consistencies and meanings” (p. 453).

3 Results

For the undergraduate participants who returned the survey, six (12 %) were sophomores, fifteen (30 %) juniors, and eleven (22 %) seniors. A total of nine (18 %) M.Ed. students answered the survey and nine (18 %) were M.Ed. students who had already graduated by the time of the survey. In terms of gender, forty-two (84 %) participants were female and eight (16 %) were male.

Participants were asked to rate their technology skills on a Likert-type scale from poor to expert. Three (6 %) participants considered themselves as slightly better than poor, 13 (26 %) as average, 30 (60 %) as less than expert, and four (8 %) as expert in technology use. When asked how often they used Facebook, participants rated themselves on frequency of use on a Likert-type scale from never to several times a day. Two (4 %) participants identified that they never used Facebook. Four (8 %) identified using it rarely, 11 (22 %) identified using Facebook sometimes, 13 (26 %) identified using Facebook often, and 20 (40 %) identified using Facebook several times a day. Forty-seven (94 %)

noted that they used Facebook most often for social use while three (6 %) used it primarily for academic use.

Participants were asked to indicate on a Likert-type scale how often they used social networks as a learning tool during their undergraduate and graduate programs. Two participants indicated that they never used social networks as a learning tool. Seven-teen (34 %) participants indicated that they rarely used social networks. Sixteen (32 %) participants indicated sometimes using social networks. Twelve (24 %) participants indicated using social networks often as a learning tool and three (6 %) indicated using social networks very frequently in this capacity.

Participants were asked to indicate how often they were required to create video postings for class on Facebook on a Likert-type scale (from never to very frequently). Six (12 %) participants indicated they had never been required to do so. Twenty-five (50 %) indicated that they had rarely been required to create a video posting on Facebook for class. Twelve (24 %) students indicated that they had been required to do video postings on Facebook sometimes. Seven (14 %) noted that they had been often required to create video postings while no participates noted that they were required to post video postings frequently on Facebook for classes.

When asked how they found the experience of creating video postings on Facebook for class, they indicated their responses on a Likert-type scale (from very difficult to very easy). Two (4 %) of the participants indicated that the process was very difficult. Four (8 %) indicated that the process was difficult. Thirteen (26 %) of the participants were neutral about the process, finding it neither difficult nor easy. Fourteen (28 %) of the participants found the process of creating video postings easy and 16 (32 %) of the participants found the task very easy.

When asked about how they felt about their privacy when using Facebook as an educational tool, participants indicated their responses using a Likert-type scale (from felt that privacy was compromised to felt that privacy was protected). Three (6 %) of participants felt that their privacy was compromised. Eight (16 %) of the participants noted that they felt that their privacy had been somewhat compromised. Sixteen (32 %) of participants noted a neutral response in which they felt their privacy neither compromised nor protected. Fourteen (28 %) of the participants felt that their privacy was somewhat protected, while ten (20 %) of the participants felt that their privacy was protected.

When asked what was the most important aspect educationally to using Facebook, participants were provided with the following choices; text-based communications, video-based sharing activities, and prefer to not use Facebook educationally. Fourteen (28 %) of the participants indicated that they believed that text-based communication were the most important quality. Twenty-five (50 %) of the participants indicated that video based sharing activities were the most important quality of using Facebook educationally. Eleven (22 %) of the participants indicated that they preferred not to use Facebook educationally.

If participants noted that they preferred not to use Facebook educationally, they were asked to provide narrative data regarding that choice. Overarching themes were formed from the narrative data provided by the participants in this study. Several participants noted that they felt that there were better alternatives to using Facebook such as the university's Learning Management System (LMS) or other providers that are limited more directly for academic use. This also overlaps with the notion of having to manage multiple education portals rather than a one-stop-shop for all course requirements. A participant noted that, "I prefer not to use Facebook educationally because many of the assignments such a video posts or discussions that are being done through Facebook can be done through Angel [LMS] where it is more private and less chaotic." This theme was further observed in a participant noting:

> I think that the abilities that Facebook gives academically can be out-done by other academic sites. To my knowledge, there isn't anything that Facebook has above other academic sites other than its popularity as a social media site. There are other learning portals that allow blogs, threads, and posts and I prefer to keep my social life separate from my academic life. While using Facebook was easy to do it was also another site that I had to remember to check. Since we use Angel for class as well I think it would be easier to keep track of if it was all on one site. I am not on Facebook all of the time, if I were then it would be easier. I think the way we used it was effective and beneficial it was just another source to keep track of.

This also alluded to participants wanting to compartmentalize their academic lives from their non-academic lives; "I try and keep my personal life VERY separate from my educational and professional life."

Participants also noted concern about privacy risks, equating a larger social presence on Facebook being synonymous with less security and posing a privacy risk:

> Facebook is about the sharing of personal information and can be publicly searched. Having a large social presence is a potential security risk. While social interaction is a part of the classroom experience, it is not the main goal. The people I interact with on Facebook are already my friends, I do not use it to make new ones.

Furthermore, some participants felt that using social media educationally was incongruent with current school policy and practices (since participants were in teacher-education programs in this study):

> We are not permitted to use Facebook educationally in my district, and although I made a request to use Twitter as a tool for parent communication, it was denied. I believe that this is due to misuse [by students and staff during personal time], and an inappropriate situation between a teacher and students last year. We have been told that we should not use social media, and to be aware that our personal accounts are monitored.

When asked about using Facebook for education and establishing social presence online, participants indicated their responses on a Likert-type scale (from very poor to very good). Four (8 %) participants noted that a very poor social presence was established by using Facebook. Four (8 %) participants noted that a poor social presence was established by using Facebook. Twenty-four (48 %) participants noted that a social presence was neither poor nor good when using Facebook. Eleven (22 %) participants noted that a good social presence was established by using Facebook. Seven (14 %) participants noted that a very good social presence was established by using Facebook.

Participants were provided with a list of strategies and were asked to check all that applied which they identified as helping to establish social presence when using Facebook as a learning tool. Strategies included; *"liking" other students' posts; commenting on other students' posts; "liking" professors' posts; commenting on professors' posts; posting your work for people to see; professors "liking" your work; professors commenting on your work; reading comments on other students' work; watching video posts from classmates; and watching video posts from professors.* Participants were provided with the opportunity of adding additional strategies; however, no additional strategies were listed.

Twenty-eight (56 %) of the participants noted that social presence was established by *"Liking" other students' posts*. Nineteen (38 %) of the participants noted that social presence was established by *"Liking" professors' posts*. Twenty (40 %) of the participants noted that social presence was established by *commenting on professors' posts*. Twenty-seven (54 %) of the participants noted that social presence was established by *posting your work for people to see*. Twenty-three (46 %) of the participants noted that social presence was established by *professors "liking" your work*. Twenty-seven (54 %) of the participants noted that social presence was established by *professors commenting on your work*. Twenty-three (46 %) of the participants noted that social presence was established by *reading comments on other students' work*. Thirty-four (68 %) of the participants noted that social presence was established by *watching video posts from classmates*. Twenty-nine (58 %) of the participants noted that social presence was established by *watching video posts from professors*.

4 Discussion

The majority of students described themselves as having average to less-than expert technology skills and most students disclosed that they used Facebook often, to several times a day. This is not a surprising outcome as the landscape of education has change greatly. Prensky (2010) observed that children come to K-12 schooling as digital natives and to meet these students on common ground "technology is becoming an important part in students' education. But just how to use it in school is not yet, completely clear, and most educators are at some stage of figuring out … how to use technology meaningfully for teaching" (p. 3).

To define how students use Facebook, participants were asked to describe whether they most often used the social network for social versus academic use. Overwhelmingly, Facebook was described as used for social interaction, with 94 % of participants using the social network in this manner. This could be influenced by the fact that social networks are still underutilized for academic purposes in higher education. Sánchez, Cortijo, and Javed (2014) noted "Facebook is the most popular Social Network Site (SNS) among college students. Despite the popularity and extensive use of Facebook by students, its use has not made significant inroads into classroom usage" (p. 138). In this study only 6 % of the participants identified using social networks very frequently for academic purposes.

It does not appear that lack of requisite technology skills in using social networks for academic use seems to be an issue for students themselves. This came to light when asked several questions on the survey. When asked how often they were required to produce and post videos on Facebook for academic work, 50 % of participants noted rarely being required to perform such a task, but since this group was purposely sampled and had all been required to perform such a task at least once previous to taking the survey they were asked to rate the difficulty in producing and posting video on Facebook. Sixty percent of the participants found the task of creating and posting video to Facebook easy to very easy. This evidence supports the claim of student-expertise in technology skills, what seems to be lacking is the skill to be required of them in higher education settings and their comfort or motivation in using social networks as a platform for learning.

There was a spectrum of beliefs about the privacy of using social networks toward learning outcomes. Only 48 % of the participants believed that their privacy was protected in some manner by using Facebook. It is important to note that for the purpose of this study, closed Facebook groups were used in which only members of the group could see each other's posts and the professors moderated who were accepted into the groups. 32 % of participants felt their privacy was neither compromised nor protected by using Facebook. To clarify the issue of privacy, narrative data was collected. When asked if they had responded that they would prefer not to use Facebook educationally, participants noted that:

- Social presence on social networks poses a potential security risk because personal information can be publicly searched
- Since school policy and practices afford little space for social media and social networks, teacher education programs in higher education should mirror these practices
- Direction is given by K-12 school administration not to use social media and teachers are informed that accounts are monitored accordingly.

These summative points reflect the thematic beliefs of 44 % of the participants regarding privacy and security issues in utilizing social networks for educational purposes.

To focus on social presence and interaction modality, participants were asked their preference in using text-based or video-based sharing activities. 28 % preferred text-based interaction whereas 50 % preferred video-based sharing activities. This alludes to the need of flexibility when planning for PLEs in general as there seems to be distinct preferences regarding modalities of interaction.

Perhaps the most interesting finding of this study was in asking participants to rate Facebook in the establishment of social presence online. 36 % of participants rated Facebook as good to very good in establishing social presence online. 48 % of participants indicated that they felt that Facebook neither established good nor poor social presence; in essence they could not identify social presence while using Facebook educationally, having been given a definition of social presence online. What is interesting is that when asked what lead to social presence on Facebook, given a list of action items, the majority of participants identified strategies in five categories that lead to social presence establishment on Facebook during their coursework; social presence was established by *"liking" other students' posts*; social presence was established by *posting your work for others to see*; social presence was established by *professors commenting on your work*; social presence was established by *watching video posts from classmates*; and social presence was established by *watching video posts from professors*. Notably, there is a mix of the validity of modalities from simple action as "liking" to text commentary and video postings. What is of interest is that even though students were not able to identify social presence in using Facebook educationally in general, having been given a definition of social presence online, participants still validated several actions as contributing to the establishment of social presence on Facebook when using it educationally.

The question then returns to how students view Personal Learning Environments in regards to using social networks. Although using Facebook as a PLE was a forced issue for students in the fact that they did not have a choice in utilizing the social network or choosing another viable means to showing their work, they were not given direction on how to interact with their peers once they had posted their work. This did provide a framework for the "aggregation of different, often web-based tools and services into a learning environment customized to the needs and preferences of an individual learner" (Jeremić, Milikić, Jovanović, Brković, Radulović, 2012, p.28) in the fact that they chose how to provide feedback to one another in terms of simply "liking," posting commentary, or follow up videos. This allowed students to engage socially by providing space:

Regardless of the particular tool they are using in the given moment, students' global online presence can provide those missing nonverbal cues typical for face-to-face interaction. This further increases students' awareness of each other and positively affects their willingness to collaborate (p. 28).

The tension seemed to arise around taking, what students interpret as a tool for recreational or personal uses and repackage it toward teaching and learning and their beliefs about the privacy of using Facebook groups, even if they are closed groups.

5 Conclusion

What is built here is a characterization of the profile of college student use and beliefs about using social networks as a learning and teaching tool in higher education. Even though students rated themselves relatively well in having requisite technology skills and 94 % of students used Facebook primarily for social use, they were hesitant to migrate these skills to academic use because of concerns of privacy, believing that other platforms could fulfil the same purpose, and by not seeing the validity to use Facebook in establishing social presence. What lies at odds with these beliefs is that when asked to identify strategies in Facebook that enabled social presence to occur in academic work, the majority of students identified strategies in five categories that lead to social presence establishment on Facebook during their coursework.

References

Abreu-Ellis, C., Ellis, J. B., Carle, A., Blevens, J., Decker, A., Carvalho, L., Macedo, P. (2013). Language Learning: The Merge of Teletandem and Web 2.0 Tools. *Journal of Interactive Learning Research*, 24(4), 353–369. Chesapeake, VA: AACE. Retrieved January 10, 2014 from http://www.editlib.org/p/39333.

Aristotle. (350 B.C.E.) *Politics*. Retrieved from http://classics.mit.edu/Aristotle/politics.1.one.html

Bandura, A. (1977). *Social Learning Theory*. Englewood Cliffs, NJ: Prentice Hall.

Biocca, F., Harms, C. (n.d.). *Defining and measuring social presence: Contribution to the networked minds theory and measure*. East Lansing, MI: Media Interface and Network Design (M.I.N.D.) Labs., Dept. of Telecommunication, Michigan State University.

Breener, J., Smith, A. (2013). *72% of online adults are social networking site users*. Pew Research Center. Retrieved from http://www.pewinternet.org/Reports/2013/social-networking-sites.aspx

Charles, A. S. (2012). Cell phones: Rule-setting, rule-breaking, and relationships in classrooms. *American Secondary Education*, *40*(3), 4–16.

Federico, N. (2013, July). Swimming with the sharks: The dangers of social media for athletes. *Wired Magazine*. Retrieved from http://www.wired.com/insights/2013/07/swimming-with-the-sharks-the-dangers-of-social-media-for-athletes/

Jeremic, Z., Milikic, N., Jovanovic, J., Brkovic, M., Radulovic, F. (2012). Using Online Presence to Improve Online Collaborative Learning. *International Journal of Emerging Technologies in Learning (iJET), 7*(2012). Retrieved January 10, 2014 from http://www.editlib.org/p/44965.

Kehrwald, B. (2010). Being online: Social presence as subjectivity in online learning. *London Review of Education, 8*(1), 39–50.

Matthews, K. (2012, April 18). School Social Media Policies: Should Teachers And Students Be Facebook Friends? *The Huffington Post*. Retrieved from http://www.huffingtonpost.com/2012/04/18/should-teachers-and-stude_0_n_1435728.html

National Center for Education Statistics. (2010). *Educational Technology in U.S. Public Schools: Fall 2008*. Washington: U.S. Department of Education.

Patton, M. Q. (2002). *Qualitative Research & Evaluation Methods*. Thousand Oaks, California: Sage Publications, Inc.

Prensky, M. (2008). Turning On the Lights. *Educational Leadership, 65*(6), 40–45.

Prensky, M. (2010). *Teaching digital natives: Partnering for real learning*. Thousand Oaks, CA: Corwin, A Sage Company.

Public Broadcasting System. (Feb. 4, 2013). *Press release: PBS survey finds teachers are embracing digital resources to propel student learning*. http://www.pbs.org/about/news/archive/2013/teacher-tech-survey/

Sánchez, R. A., Cortijo,V., Javed, U. (2014). Students' perceptions of Facebook for academic purposes. *Computers & Education, 70*, 138–149.

Topping, A., Coyne, E. (2013, August 6). Cyberbullying on social networks spawning form of self-harm: Charities warn of vulnerable youngsters encountering pernicious abuse online but unable to cease using sites. *The Guardian*. Retrieved from http://www.theguardian.com/society/2013/aug/06/cyberbullying-social-networks-self-harm

Tu, C., McIsaac, M. S. (2002). The relationship of social presence and interaction in online classes. *The American Journal of Distance Education, 16*(3), 131–150.

Varlas, L. (2011). Can social media and school policies be "friends"? *ASCD Policy Priorities, 17*(4). Retrieved from http://www.ascd.org/publications/newsletters/policy-priorities/vol17/num04/Can-Social-Media-and-School-Policies-be-%C2%A3Friends%C2%A3%C2%A2.aspx

Vygotsky, L. (1978). Interaction between learning and development. In M. Gauvain and M. Cole (Eds.) *Readings on the Development of Children* (1997). New York: W. H. Freeman and Company.

Biographies

Dr. Jason Brent Ellis, Associate Professor, joined the Ashland University faculty in 2007. A native of Canada, his research focuses on technology use by students with disabilities in K-16 education, ways of levelling the digital divide through policy and pedagogical interventions, ways of providing access to under-represented and non-traditional students, the creation of sustainable models of education in emerging global economies, and scaffolding in distance education.

Dr. Carla Abreu-Ellis, Associate Professor, holds a Ph.D. in Higher Education Administration from Bowling Green State University. Dr. Abreu-Ellis teaches undergraduate and graduate courses in the Dwight Schar College of Education at Ashland University. Her research interests include College students with disabilities, technology and disabilities, access to education and retention of college students, Universal Design, and diversity.

Teaching Data Management: Key Competencies and Opportunities

Andreas Grillenberger, Ralf Romeike
Computing Education Research Group
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3
D-91058 Erlangen
{andreas.grillenberger, ralf.romeike}@fau.de

Abstract: Data management is a central topic in computer science as well as in computer science education. Within the last years, this topic is changing tremendously, as its impact on daily life becomes increasingly visible. Nowadays, everyone not only needs to manage data of various kinds, but also continuously generates large amounts of data. In addition, Big Data and data analysis are intensively discussed in public dialogue because of their influences on society. For the understanding of such discussions and for being able to participate in them, fundamental knowledge on data management is necessary. Especially, being aware of the threats accompanying the ability to analyze large amounts of data in nearly real-time becomes increasingly important. This raises the question, which key competencies are necessary for daily dealings with data and data management.

In this paper, we will first point out the importance of data management and of Big Data in daily life. On this basis, we will analyze which are the key competencies everyone needs concerning data management to be able to handle data in a proper way in daily life. Afterwards, we will discuss the impact of these changes in data management on computer science education and in particular database education.

Keywords: Data Management, Key Competencies, Big Data, NoSQL, Databases, Data Privacy, Data Analysis, Challenges

1 Introduction

Nowadays, data take a key position in nearly everyone's daily life. Enormous amounts of data are generated and organized every day, for example when storing documents or music, when using social media, but also in a less obvious way during activities like using public transport (by using electronic tickets), when consulting a doctor (by using electronic health insurance cards), and so on. Therefore, handling data in daily life has many facets: data may be captured wittingly or unwittingly, it may be stored locally or in online (cloud) stores, it may have different structures, and so on. As these data are captured almost everywhere, data analysis enables the reconstruction of large parts of the daily routine with high statistical relevance. For example, this is the case when taking together data coming from public transportation with the payments with credit cards and the data captured by a smart meter used in the private household. With these information, the daily routine like working hours, shopping habits but also which devices are used at home may be reconstructed by analyzing the times one uses public transportation, where someone goes shopping and by analyzing the power consumption information captured by the smart meter (Molina-Markham, Shenoy, Fu, Cecchet, Irwin, 2010). While this example deals with data collected about a person by third parties, in rarer cases people also collect data on themselves: Participants in the "Quantified Self" movement (also known as "life logging") actively gather and analyze data on their own life for different purposes, sometimes for improving health, for improving well-being, or for improving own productivity, in other cases just out of curiosity.

Today, data have a clear influence on daily life and this influence is continuously increasing. A central task is handling these data in a responsible way, as storing, modifying, deleting and using data are typical aspects concerning everyone's life. Storing large amounts of data not only includes selecting an appropriate data store, but also structuring and organizing this data, deciding whether to create backups (and which backup methods to use), synchronizing data between multiple devices (and perhaps users), and so on. This also involves protecting own data and such about other persons from being manipulated, lost or from being used abusively, but also methods for guaranteeing the authenticity of data. For recognizing the necessity for doing so, people also need to recognize the value of data. This is especially possible when analyzing data on their own. Nowadays, such possibilities are opened up for everyone because various comfortable analysis tools are available for free, but also because of the Open Data movement, which targets on publishing as many data sets as possible in order to allow many-sided usage. These data analysis methods can also enable people to make data captured in daily life valuable for personal use and to extract new information out of known data.

In computer science, the aspects mentioned before are often summarized under the term “Big Data”. This is a topical subject in various contexts, as it not only affects people’s life as well as CS, but also has strong impact on economy, politics and security. Typically, the term Big Data is described as handling large amounts of data with varying structures and high velocity (Laney, 2001). This includes continuously generating data but also fast processing of such in nearly real-time. Various topics that are being discussed in public dialogue today are strongly affected by Big Data and particularly by data analysis, e. g. early data retention or surveillance programs of intelligence agencies. These topics are often hard to understand, as data management and data analysis are complex topics with rising importance, while the knowledge needed for understanding them is not part of current (CS) education.

In the context of these current developments, the relevance of data management for people changes fundamentally: While until now, data management and specifically databases were topics that were mainly considered as relevant for educational and professional use, these topics are now affecting the whole life of everyone. Despite this changing impact, current teaching considers databases as central topic of data management education (cf. e.g. Brinda, Puhlmann, Schulte, 2009; Seehorn et al., 2011). In the future, other topics like data safety and data privacy will gain importance in everyday life but also in education. Thereby, the purpose of data management education changes tremendously. While the current focus of teaching is set on concepts of and knowledge on databases, in future this emphasis needs to be changed to fostering competencies needed for everyone. Hence, current database education needs to be revised and adapted to these new requirements in order to teach sustainable concepts and aspects of data management that are not only relevant in computer science and computer science education, but also in daily life. So, for being able to deal with the new possibilities and threats, new key competencies coming from data management have to be fostered in class.

In order to point out these new and reappraised key competencies, in this paper we will first describe main aspects of data management and Big Data that are relevant for teaching but also for the students’ life. On this basis, we will point out major key competencies, which people need for being able to deal with the new possibilities and threats evolving in context of data management. Finally, we will outline the consequences for computer science education by discussing main challenges computer science education will have to deal with in the future.

2 Data Management in the Context of Big Data

Handling data is an important task today. This includes planning, organizing and utilizing data, methods which are typically summarized by the term “data management” (Bodendorf, 2005). In addition to these aspects, data management also comprises for example evaluating the quality of data, acquiring data, securing access to data as well as data backup and recovery (DAMA International, Mosley, Brackett, Earley, 2009). These aspects are clearly concerning data management in daily life, as described before.

The topic data management changes tremendously under the influence of current developments like Big Data: Although data management has been an important task in daily life for years, people often only consider it as topic in computer science (education), because the influences of data on daily life were only hardly recognizable so far. Nowadays this influence becomes increasingly obvious: While in 2012 approximately a total of 2.7 Zetabytes (10^{21} Byte) of data existed, about 2.5 Exabyte (10^{18} Byte) of additional data were generated per day (IBM, 2012). However, Big Data is not only characterized by the large amount of data and this high rate of data generation, but also by the variety of data: About 80 % of these data are unstructured or of varying structure, as they are especially coming from social media. For example, in 2012 about 100 Terabytes of data were uploaded daily on Facebook and 230 Million tweets were sent on Twitter per day (IBM, 2012). Therefore, Big Data is often characterized by (at least) three Vs: “volume”, “variety” and “velocity” (Laney, 2001), other Vs like the “veracity” of data are added in some descriptions. When dealing with Big Data, new challenges are evolving: usually, the most commonly used relational database management systems are optimized for consistent, durable and reliable storage of data. In contrast, newly emerging databases (often summarized under the term NoSQL[1] databases) set the focus on fast processing of distributed stored data and thus accept limitations, such as of consistency. In addition, other aspects of data management, like data safety and privacy are strongly affected by these new developments.

1 Nowadays, the term NoSQL is interpreted as “Not Only SQL” (Edlich, n.d.), while originally it was used as name for a database management system not supporting SQL at all (Strozzi, 1998).

3 Key Competencies in Data Management

With the increasing impact of data management, CS education needs to analyze which knowledge and skills people need in order to deal with this topic in their life. As data management is a complex demand with strong influence on different fields of daily life, skills necessary for successfully coping with this topic are described as key competencies according to the definition by Rychen, Salganik (2001). These will be derived from all parts of data management – including its main aspects structuring, organizing and utilizing data (Bodendorf, 2005) – but also from discussing the consequences of data management and its influence on data privacy. Therefore, we will hereafter point out the most important key competencies related to the topic data management. These key competencies will be illustrated by examples describing their relevance for people's life.

3.1 Storing data

Nowadays, everybody stores and manages enormous amounts of data every day, e.g. text files, videos, music, e-mails and so on. For storing data, various possibilities exist that strongly differ especially in the following aspects:

- storing data offline or in the cloud,
- storing data as files in a file system or as entries in a database,
- using specialized stores like media stores or not,
- storing data in a structured or unstructured way.

In most cases, data are stored as files in file systems, locally or in the cloud, e.g. when dealing with documents, photos, music and so on. In contrast, there are also data stores that are specialized on only a few data types. For example, e-mail clients are data stores for e-mails but also for contacts and in some cases calendar entries, while music could instead be stored in media libraries, often together with videos and pictures. By using an application specialized for concrete use cases, dealing with specific types of data will be distinctly simplified. As each time when storing data the requirements differ, it is not possible to decide in general if to use specialized stores or not, as these stores also have disadvantages: for example, they often use proprietary file formats and thus comprise the threat of a "vendor lock-in"[2]. Also, all other aspects mentioned

2 The term "vendor lock-in" describes the dependence on a single vendor of products. This is for example the case, when data are stored in a proprietary file format, so that using them in another vendor's product is hardly possible.

above have to be decided from case to case. These decisions are summoned by the question: *"Which data should be stored – and where and how?"* Therefore, handling data implicitly involves knowledge on the different possibilities for storing data.

The decision which data store with which functionalities should be used in a concrete use case has to be made as the case arises. So, for being able to deal with data, and especially large amounts of such, in a proper way, it is necessary that students *understand and apply different ways for storing data.*

3.2 Handling metadata

All these large amounts of data that are generated every day bring additional information with them, the so-called metadata. These are for example visible as attributes of a file (e.g. creation as well as last-modified date, author/creator…), as log files (e.g. in cloud storage services: "file 'example.txt' was deleted on 2014-02-01 09:07 by 'user1'") or as tracked changes in a document. Although most metadata are actually accessible by the user, they are often disregarded: While final versions of a document are typically cleaned from e.g. comments and notes, often metadata are not revised. These data may contain information not supposed to be contained in the finalized document, or they may have been generated from old content, like old or wrong keywords. In addition, information that may be confidential, like the concrete author of a document or the time span between creation and last modification is usually included. There are various examples, where confidential data were disclosed by metadata included in published documents. For example, in 2005, the United Nations published a report on Syria's involvement in a murder; this document not only contained the visible information, but also tracked changes. These annotations were only hidden and contained, i. a. names of persons involved in this plot that were not supposed to be disclosed (Zeller, 2005). Another example for the hidden generation of metadata is the automated geotagging of photos taken by smartphones: typically, such devices add the current GPS position as metadata to all photos taken with this device. When sharing such photos, the user then probably shares more information than intended without noticing it. Therefore, when handling data in daily life, people need the key competency to *note that additional data may be included in data sets as metadata.*

On the other side, these metadata may simplify dealing with data: by adding additional information as metadata, locating data is simplified and accelerated. In particular, metadata are necessary when searching for information by substantial criteria, as most file contents cannot be interpreted directly, so searching by content might only be possible for pure text files. For example, when dealing with photos, adding metadata that describe the place the photo was taken at, or by marking persons who are visible on it, locating this photo afterwards will obviously be simplified in comparison to searching without such information. As metadata are typically considered by search engines of operating systems, but also within most of the currently used database management systems, being able to deal with metadata simplifies daily data management a lot. However, also the disadvantages of using metadata should be kept in mind: not only the effort of assigning and maintaining them may be relevant for the decision whether such information should be added, because even the usefulness of them strongly depends on the concrete use case. Additionally, while the existence of metadata strongly benefits reading data from the data store, typically writing operations are slowed down, because not only the data but also the metadata need to be updated in order to ensure consistency. Therefore, another key competency in data management is to *understand the purpose of metadata and use them in a proper way.*

3.3 Handling redundancy and consistency

When managing data, people will always have to deal with redundancies and inconsistencies: for example, for files related to multiple topics it often seems reasonable to store copies of the file in multiple folders of the file system. This leads to inconsistent data when one file is being updated while at least one copy is (accidentally) left untouched. Summarizing, data stored redundantly tend to become eventually inconsistent. Since this problem, needing a duplicate copy of a file in another location/folder, is not unusual when saving data, students need to understand the consequences of storing data redundantly. In addition, they have to deal with this requirement, e.g. by creating a link to the data at the second location instead of saving a real copy of it. Therefore, to *understand the consequences of storing data in a redundant way* as well as to *save data in a proper way in order to prevent inconsistencies* are key competencies of data management.

Today, another common cause for inconsistencies is the synchronization of data between multiple devices. Nowadays, one person carries in average 2.9 (mobile) devices including laptops, smartphones and tablets (Truong, 2013), and the overall number of devices used by one person may be even higher. Data are often synchronized between two or more of these devices, and not only read but also modified on these. This leads to inconsistencies when modifying data that was earlier changed in another location, but not successfully synchronized to the other devices yet. This leads to different possible consequences, dependent on the application used for synchronization and on the type of synchronized data: while only in special cases (such as pure text files) conflicts may be automatically resolved, in most cases duplicate data will come up or in the worst case data will be lost. So, another key competency in this topic, which is needed to be able to understand threats when synchronizing data, is to *understand the consequences of synchronizing data and deal with synchronization conflicts*.

While commonly redundancies and inconsistencies should be avoided, there are also use cases where both concepts are used intentionally: for example, backups are redundant copies of the original files and will become inconsistent as soon as the original files are modified again. However, in this case redundancy occurs by design, because backups serve as fallback copies, especially for the case that data are accidentally deleted or changes must be reverted. So, they need to be redundant to the original file (for restoring) and need to become inconsistent when the original file is being modified (for reversing changes). Today, as in most operating systems different backup functionalities exist, people also need to be aware of the different ways for creating backups: continuous backup vs. backup at discrete points in time, incremental vs. complete backup, hot vs. cold backup. These possibilities clearly differ in used hard disk space, in the speed of the backup and restore processes, and in the typical frequency of backups. For each use case, it is necessary to decide, which aspects are required – a decision which has to be done in context of the value of the concrete data. Therefore, another key competency in this field is to *create redundant data sets for backup/data safety in a proper way.*

3.4 Data safety and encryption

Nowadays, a great part of one's personal life is captured as data. With smartphones and other mobile devices, an increasing amount of moments is immediately captured, for example in form of posts in social media, photos, but data is also in background, e.g. as position data, log files of sensors and so on. Often, the data everyone manages and generates are not only stored on stationary desktop PCs, but also on mobile devices as well as portable USB drives without any security measures, and they are often transferred via insecure communication channels. This results in privacy issues, but also enables even more problematic threats like identity theft. In professional context, financial losses may occur. Storing data on mobile devices or storage media enlarges the risks of unauthorized usage of these data, of manipulations and of data theft.

Consequently, it is an important task to store and transfer private or confidential data in a secure way. This may be reached by different ways. A typical method for securing data in case of theft of the device, on which the data are stored, is to restrict access to the device. This is often accomplished by using password protection or similar authentication methods. Although this will increase data safety, as accessing data becomes more difficult, data safety cannot be guaranteed by this method, because data may yet be accessible stored on the device. A simple approach for overcoming such authentication methods is reading the data store (for example the hard drive) using another device that does not enforce the authentication. Users need to be aware, that usually typical authentication methods cannot suffice to secure their data, as they are only a hurdle for accessing these. Therefore, people must differentiate between restricting the access to devices and to the data itself. To (relative strictly) ensure data safety, it must be prevented that the meaning of data is recognizable without the required permissions. This is the goal of data encryption: While encrypted data might still be read from the hard disk, they have no value for anyone until being decrypted using the right key. So, another key competency when managing data is to *understand the difference between restricting access to a device or service and protecting the data stored on it* as well as to *encrypt data and communication* in order to prevent unwanted access to these data.

Another aspect concerning data safety is to decide whether to confide specific data or not. As for example, the attribute *author* of files, e-mails and so on is typically not protected against changes nor is the content itself; these data carry the risk of being manipulated. As in various use cases it is necessary to be able to trust data, everyone needs to be aware of methods for checking the authenticity of data. For example when reading e-mails, today most people keep in mind that these messages may contain non-authentic content, as they may be somewhat junk or phishing mails. However, with an increasing quality of such messages, it would be harder to figure out if an e-mail is authentic or not. Other data than e-mail are often less questioned, because threats are less obvious and less discussed in public dialogue. Therefore, methods for proving the authenticity of data will become more relevant in the future, especially because an increasing amount of legally relevant tasks is done via electronic communication methods. One technique for guaranteeing the authenticity of the sender information as well as the validity of the content is by digitally signing the data. This enables the recipient to check if data were manipulated. Therefore, it is necessary that people *know methods for guaranteeing the authenticity of data and use them in a proper way.*

3.5 Using methods of Data Analysis

Today, various sets of information and data are available and accessible for free. However, only few people are able to use these data in another way than by just looking at them or analyzing them manually. For example, by combining data from various sources, interesting new use cases may be found as well as new information may be extracted. Today, this is possible for everyone: different simple tools for analyzing data are provided for free by the large Big Data companies like Google or IBM. In addition, there are simple tools for creating mash ups, a form of integrating multiple data and especially media. An example for using open data sets is evaluating whether to book a hotel in a concrete neighborhood in another way than by reading the opinions of former visitors. For example, the City of New York offers many data they capture daily as open data sets.[3] This includes calls to the service number 311, which are concerning complaints on noise, street or sidewalk conditions and so on.[4] While the direct results of analyzing these data are relatively obvious, they can also be combined with other data, such as restaurant inspection results[5], in order to gain prediction factors, for example if the noise conditions in a borough and the ratings of the restaurants in this part of a town correlate or not. Doing such data analysis is possible with simple techniques, for example included in spreadsheet applications or available as online tools. Typical data analysis methods are grouping of data ("clustering", in this case by neighborhood), categorizing them by different characteristics ("classification", in this case e.g. the types of service calls but also the restaurant grades) or by determining interdependencies (if-then-relations) between data ("association", e.g. the described analysis if the restaurant grades and noise conditions correlate). So, another key competency in data management is to *use, find and combine data in order to gather new information*.

By analyzing data themselves, people learn to recognize the threats for data privacy coming from data analysis. With the ability to combine data from different sources, it is only a small step into discovering that the same methods may be used when analyzing personal data. Even data strongly anonymized or pseudonymized according to data privacy acts may be deanonymized – so data privacy acts would be bypassed. This was for example the case, when AOL released a set of search data, which included a user's search terms as well

3 NYC Open Data: https://data.cityofnewyork.us.

4 Analyzation of New York City 311 Service requests: http://opendatabits.com/new-york-city-311-servicerequests-open-data.

5 Meshup of NYC 311 calls together with restaurant inspection results: http://opendatabits.com/nyc-restaurantinspections-results-open-data.

as a unique person ID related to the user, but without revealing personal data of this person. By analyzing these data, some data analysts were rapidly able to recognize some persons out of these data and discovered their real name, contact data, as well as their search habits at AOL's search engine (Barbaro, Zeller Jr., 2006).

Therefore, another key competency of data management, which involves not only data analysis but also data privacy, is to *know the threats for data privacy and analyze data with keeping ethical demands in mind.*

3.6 Being aware of Data Traces and Data Privacy Issues

With the possibility to store and analyze huge amounts of data, different threats for data privacy are evolving. As mentioned before, metadata may be harmful if the user does not know about them, or when handling them in an inappropriate way. In addition, a lack of data safety and encryption strongly affects data privacy. This threat is even intensified when dealing with modern devices, applications and services, as various types of data on this usage and on the user are captured continuously. While the main aim of some services is capturing data in a relatively obvious way, such as in social networks, in other cases they are generated in a hidden way besides the intended use, for example as log files. Additionally, applications supposed to generate data, like the mentioned social networks, tend to store more data than actually needed for the service to work. While in some cases, the user is aware of this data generation and actively decided for capturing these data, such as when participating in the "Quantified Self" movement, this is usually not the case. However, by combining different sources of such data, large parts of daily life may be reconstructed. As nowadays, everyone uses different kinds of data stores, but also applications using these data, one leaves digital data traces everywhere. Hence, the question if one trusts an application/service or not becomes increasingly important for data privacy, as when using an application there is usually no possibility to decide whether it is allowed to capture data.

Examples are typical chat applications that offer the possibility to display "last online" times to contacts: by having a look on these times (and perhaps comparing them to the ones of other persons) other people can digitally trace persons with simplest methods. Depending on the concrete application, this tracking is even possible without prior contact to a person, only by adding them on the contact list without a need for approval of this request by the person added. Therefore, the decision to disclose own data, like these online times, but also the uploaded photo and the status message, is implicitly made when registering with this service and using it.

So, raising students' awareness on such abilities and threats is an important aspect when discussing data privacy. As such methods for collecting data are typically hard to discover and in most cases cannot be prevented, users need to be aware of these possibilities in order to be able to recognize hints on such issues, e.g. the "last online" times in chat applications mentioned before. This is especially, when confiding large amounts of data to one provider, e.g. when storing data on cloud storage services.

Therefore these aspects of data privacy are summoned by the questions *"Who stores which data on me? Who has which information on me? Who can I confide data about me?"* So, to *note own data traces* but also to *know the possible threats of using data storage services* are important key competencies, which are hard to foster, because tracking such traces is mostly done in an invisible way, as well as threats when using data storage services are typically hard to discover.

3.7 Overview

As described before, several key competencies in data management could be identified. These will be summarized in order to get a complete overview:

People…

- understand and apply different ways for storing data
- note that additional data may be included in data sets as metadata
- understand the purpose of metadata and use them in a proper way
- understand the consequences of storing data in a redundant way
- save data in a proper way in order to prevent inconsistencies
- understand the consequences of synchronizing data and deal with synchronization conflicts
- create redundant data sets for backup/data safety in a proper way
- understand the difference between restricting access to a device or service and protecting the data stored on it
- encrypt data and communication
- know methods for guaranteeing the authenticity of data and use them in a proper way
- use, find and combine data in order to gather new information
- know the threats for data privacy and analyze data with keeping ethical demands in mind

- note own data traces
- know the possible threats of using data storage services

It is clearly visible, that these competencies face different aspects of data management. But at the same time, they are all strongly related to daily life. Additionally, most of these key competencies have one central aspect in common: they represent newly occurring decisions, which are necessary in order to deal with data management in a proper way. This mirrors the current developments in computer science: while until the last years, mainly one database system was leading and used for most use cases, there is nowadays a great variety of such systems which evolved since the development of the NoSQL databases. This makes it necessary to choose the database depending on the use case.

4 Challenges for Computer Science Education

In contrast to their relevance for everyday life, the key competencies described before do not yet receive sufficient attention in current data management education. With the increasing relevance of data management, current curricula for computer science education need to be revised with keeping the new requirements and possibilities in mind. By considering these aspects in class, computer science education changes tremendously: Especially, while current data management education mainly focuses on databases, in the future the relevance of various additional topics will increase clearly, while other current topics may then be less important. The key competencies developed above, need to be considered in computer science education, since no other subject in general educational schools can foster these, because more than basic knowledge on these topics is required.

In contrast to these new requirements, current computer science education mainly focuses on relational database management. Since the topic databases was intensively discussed in the context of computer science education in between the years 1986–1998, only occasionally papers and articles on this topic were published. While in the earlier of these years the relevance of (mainly relational) databases in class was the main topic of research, in the later of these years and now on, the focus of publications is set on supporting the teaching of databases. Therefore, various tools were discussed, especially for teaching SQL, e.g. by Grillenberger, Brinda (2012) and by Sadiq et al. (2004). Since 1998, only few research results on this topic were published, in particular there are no such publications on current developments like Big Data or the increasing relevance of data management in daily life, yet. In addition, currently no compilation of key competencies concerning this field exists. Only different educational standards, like the K-12 Computer Science Standards by the Computer Science Teachers Association (Seehorn et al., 2011) or the German Educational Standards for Computer Science in Lower Secondary Education (Brinda, Puhlmann, Schulte, 2009), mention some competencies on this topic.

In the following, we will outline some of these competencies for comparing them with the key competencies in data management described above. By having a look on the educational standards, important competencies on data management/database education are found particularly in the topics "structuring data", "data safety" and "data privacy". The former includes aspects of creating and using data structures, e.g. students "know principles for structuring documents and use them in an appropriate way"[6], they "know and use tree structures, for example directory trees"[6] or they "navigate in directory trees and manipulate directory trees in a proper way"[6] (Puhlmann et al., 2008). The latter ones – "data safety" and "data privacy" – need to be distinguished, even though they are strongly related to each other. Data safety focuses on the technical aspects, like preventing prohibited access to confidential data, encrypting such data and so on, while data privacy focuses on using data (and especially personal data) in a proper way. So, competencies needed concerning data safety are for example to *"explain the principles of security by examining encryption, cryptography, and authentication techniques." (*Seehorn et al., 2011), while a typical competency on data privacy is to *"evaluate situations in which private data should be shared"*[6] (Puhlmann et al., 2008). Also, an important competency considering data management is to *"use data analysis to enhance understanding of complex natural and human systems."* (Seehorn et al. (2011)).

By comparing these competencies currently considered as important in computer science education with the ones described before, a clear difference is visible: Although most competencies of current database education will remain relevant in future, various additional ones are supplemented. The competencies fostered today are strongly related to computer science, while in future key competencies of data management will especially face handling data in daily life.

In addition, the topic databases will change clearly in context of Big Data and the newly emerging NoSQL databases. In order to meet the main requirements of storing Big Data, the management of large amounts of data with high variety and high velocity, new types of databases arose. These non-relational

6 Original in German, translation by authors.

databases are typically summarized under the term NoSQL[7] Various concepts of databases that were so far assumed as being fundamental to this topic are dropped by these databases, in order to speed up access to distributed stored data. For example, while consistency is a main requirement of relational databases and part of the ACID[8] criteria, this concept is dropped in NoSQL databases, because they are only "eventually consistent" according to their main requirements summarized as BASE[9]. Therefore, in order to teach sustainable concepts and aspects of data management and databases, the concepts fundamental to databases – and not only for relational database management systems or for NoSQL databases – need to be analyzed.

5 Conclusions

As described in this paper, by considering the new aspects coming from current developments like Big Data and because of the increasing importance of data management for daily life, data management education changes tremendously. By discussing the new aspects coming from these topics in class, various key competencies of data management will be fostered. Although these new key competencies are becoming increasingly relevant, they are mostly not yet fostered in current education on the topics data management or databases. Especially aspects coming from data privacy and data analysis will increase in importance in future data management education, but also all the other key competencies described before need to be fostered, as key competencies are "of prime importance for a successful life and effective participation in different fields of life" (Rychen, Salganik, 2001).

This will also ensure a better fit between education and daily life, because today most people use applications involving Big Data analysis multiple times daily, but without being able to notice the collection of their data or its analysis – and often without even knowing about possible consequences. Additionally, as everyone handles and generates large amounts of data continuously, also the importance of data management in daily life increases continuously: such as when storing data in different data stores (like the file system, media libraries and so on), when synchronizing data between multiple applications and/or devices or when creating backups of data.

7 NoSQL nowadays is interpreted as "not only SQL" (Fowler & Sadalage, 2012). In original, by Carlo Strozzi (1998), this term was used as name for a database not supporting SQL.

8 ACID is the abbreviation for Atomicity, Consistency, Isolation and Durability, the four main requirements on relational databases (Elmasri & Navathe, 2011).

9 BASE is the abbreviation for Basically Available, Soft-State, Eventually Consistent, the three main requirements on NoSQL databases (Edlich, n.d.).

In addition, Big Data has strong impact on current and newly emerging professions. Especially, various professions are changing when considering aspects of Big Data as well as of data analysis. Also, at the moment new professions are evolving, like the data scientist (Davenport, Patil, 2012). In this profession, aspects coming from informatics, like data analysis, are combined with mathematical ones, especially coming from statistics. Therefore, knowledge on fundamental concepts and methods of handling Big Data will have sustainable influence.

References

All electronic sources were retrieved on 17th April 2014.

Barbaro, M., Zeller Jr., T. (2006, August). A Face Is Exposed for AOL Searcher No. 4417749.

Bodendorf, F. (2005). *Daten- und Wissensmanagement [Data and Knowledge Management]*. Springer.

Brinda, T., Puhlmann, H., Schulte, C. (2009). Bridging ICT and CS: Educational Standards for Computer Science in Lower Secondary Education. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 288–292). New York, NY, USA: ACM.

DAMA International, Mosley, M., Brackett, M. H., Earley, S. (2009). *The Dama Guide to the Data Management Body of Knowledge Enterprise Server Edition*. Technics Publications Llc.

Davenport, T. H., Patil, D. J. (2012). Data scientist: the sexiest job of the 21st century. *Harvard Business Review, 90*(10), pp. 70–77.

Edlich, S. (n.d.). NOSQL Databases. Retrieved from http://nosql-database.org

Elmasri, R. A., Navathe, S. B. (2011). *Fundamentals of Database Systems*. ADDISON WESLEY Publishing Company Incorporated.

Fowler, M., Sadalage, P. J. (2012). *NoSQL Distilled – A Brief Guide to the Emerging World of Polyglot Persistence* (1. ed.). Amsterdam: Addison-Wesley.

Grillenberger, A., Brinda, T. (2012). eledSQL: A New Web-based Learning Environment for Teaching Databases and SQL at Secondary School Level. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 101–104). New York, NY, USA: ACM.

IBM. (2012). The Flood of Big Data. *Infographic*. Retrieved from http://ibmdatamag.com/2012/04/managing-the-big-flood-of-big-data-in-digital-marketing

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., Irwin, D. (2010). Private Memoirs of a Smart Meter. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building* (pp. 61–66). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1878431.1878446

Puhlmann, H., Brinda, T., Fothe, M., Friedrich, S., Koerber, B., Röhner, G., Schulte, C. (2008). Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I [Principles and standards for computer science in schools: educational standards for computer science lower secondary]. *Supplement to LOG IN, 150/151*.

Rychen, D. S., Salganik, L. H. E. (2001). *Defining and selecting key competencies*. Hogrefe & Huber Publishers.

Sadiq, S., Orlowska, M., Sadiq, W., Lin, J. (2004). SQLator: An Online SQL Learning Workbench. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (pp. 223–227). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1007996.1008055

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., … Verno, A. (2011). K–12 *Computer Science Standards*. Computer Science Teachers Association, Association for Computing Machinery.

Strozzi, C. (1998). NoSQL: a non-SQL RDBMS. Retrieved from http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home Page

Truong, K. (2013). INFOGRAPHIC: Users weighed down by multiple gadgets – survey reveals the most carried devices. *Sophos Naked Security*. Retrieved from http://nakedsecurity.sophos.com/2013/03/14/devices-wozniak-infographic

Zeller, T. J. (2005, November 7). Beware Your Trail of Digital Fingerprints. *The New York Times*. Retrieved from http://www.nytimes.com/2005/11/07/business/07link.html

Biographies

Andreas Grillenberger is research associate at the Faculty of Engineering and doctoral candidate at the Computing Education Research Group of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany.

Ralf Romeike is the head of the Computing Education Research Group at the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany.

The Student Learning Ecology

Halvdan Haugsbakken
Department of Sociology and Political Science
Norwegian University of Science and Technology
Dragvoll 7491
Trondheim, Norway
Halvdan.Haugsbakken@svt.ntnu.no

Abstract: Educational research on social media has showed that students use it for socialisation, personal communication, and informal learning. Recent studies have argued that students to some degree use social media to carry out formal schoolwork. This article gives an explorative account on how a small sample of Norwegian high school students use social media to self-organise formal schoolwork. This user pattern can be called a "student learning ecology", which is a user perspective on how participating students gain access to learning resources.

Keywords: Learning ecology, social media, high school, Norway

1 Introduction

How do students evaluate and use social media to organise formal schoolwork? This case study attempts answering that question, by connecting traits of youth's web consumer culture to Barron's (2006) concept of *learning ecology* and recent educational research on social media. The paper argues that social media is used by students beyond socialisation and informal learning. The paper offers a case study on how students blend formal schoolwork into a sphere normally associated with pastime activities. This user behaviour suggests being characterised by reflective decision-making processes, showing selective user participation. Participating students are part of a self-organised web practice, which happens beyond the instruction of their teachers. The paper verifies that out of a data sample of 26 Norwegian high school students, 12 reported using different Web 2.0 tools in the mentioned way. Some students modelled a network learning environment, which I suggest can be called a *student learning ecology*. The term is an attempt to apply and expand on Barron's

(2006) concept. To empirically describe this, we can look at the paper's content and structure. First, I take in hand the research perspective I will use. Second, I account for the applied methods and the study's data sample. Third, I perform the data analysis and present findings. Finally, I provide concluding remarks and address the study's limitations.

2 Research Perspective

The arrival of Web 2.0 (O'Reilly, 2005) has involved the introduction of several technical definitions. Boyd and Ellison, for example, define Social Network Sites (SNSs) as "web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other user with whom they share a connection, and view and traverse their list of connections and those made by others within the system" (2007, p. 211). Such an understanding involves that web services like Facebook and Twitter are SNSs, while "old" web pages, like blogs, are not (Aalen, 2013). Kaplan & Haenlein, on the other hand, have classified social media as a "group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User Generated Content" (2010, p. 61). They also (2010) suggest that there are six types of social media software: (1) Collaborative projects, e.g. Wikipedia, (2) blogs, (3) content communities, e.g. YouTube, (4) SNSs, e.g. Facebook, (5) virtual game worlds, e.g. World of Warcraft, (6) and virtual social worlds, e.g. Second Life.

Such definitions are useful. They give directions and clarify what social media "is", and what it "is not". On the other hand, they pose analytical challenges. They are technical and challenging to use, in order to capture the social side of new technologies. Applying them to explain web mediated phenomena, like Internet meme, the cyber currency Bitcoin, hacktivism, for example, could prove difficult. One needs to apply other approaches. Barron's (2006) concept of "learning ecology" suggests to be beneficial in this sense. Learning ecology is defined as "the set of contexts found in physical or virtual spaces that provides opportunities for learning" (Barron, 2006, p. 195). Inspired by socio-cultural, activity and situative learning theories (Engeström, 1987; Lave, Wenger, 1991; Vygotskij, 1978), learning ecology assumes that individuals are involved in many settings, create activity contexts within and across settings (Barron, 2006, p. 199). According to Barron (2006), learning ecology assumes the involvement of several learning processes, and the creation of activity contexts in a new setting, or, that the pursuits of learning are found outside the primary learning setting. Barron (2006) argues that it accepts informal learning and recognises the variety of literacies, practices, and forms of knowledge, which are used by youth when they interact with new technologies. Learning ecology also considers that the boundaries between different settings are permeable and that youth uses multiple cultural forms in pursuing knowledge (Barron, 2006).

Learning ecology can analytically reduce the constraints on technical definitions, recognise informal learning, stress that several social media applications are used independently of each other, to support forms of learning processes, for example. Learning ecology can bring attention to forms of network organisations, and the meaning of transactions taking place between ties in social networks. Making a distinction between *informal* and *formal* schoolwork can help further. Establishing to what extent an activity follows a learning objective or is given by an educational authority, can bring to light how a learning ecology "works" (OECD, 2014). Exact attention on how students use Google Docs to collaborate on project assignments, how they establish Facebook groups to inform each other on homework assignments, how they share files with fellows student to get feedback, for example, can be one way to answer questions raised in recent educational research on social media. Over the years, it has been documented that youth use SNSs to socialisation, personal communication, and informal learning (Madge, Meek, Wellens, Hooley, 2009). On the other hand, it seems that educational researchers to a little degree explore the *"whys"*, on why students use SNSs to create content, share, interact and to collaborate, in order to self-organise formal schoolwork (Hamid, Chang, Kurnia, 2009). Researchers are prone to argue that we need to know more about uses, practices, and user patterns (Ellison, Steinfield, Lampe, 2011). It is difficult to identify a student user perspective, which asks why certain students participate in a web mediated participative culture (Jenkins, 2006), while others refrain from being part of one. Barron's (2006) concept can act as such a bottom-up user perspective.

Educational research on SNSs, however, appears to be shaped into different trajectories. Studies still favour university students as main research subject. There are certain topics that reoccur as focal point. Studies bring closer attention to the new literacy practices, forming as students communicate in new ways (e.g. Drouin, 2011; Greenhow, Robelia, 2009). Other studies have concluded that Facebook is a tool for effective collaborative learning (Irwin, Ball, Desbrow, Leveritt, 2012), while Lurkin et al. (2009) found that students' use of Web 2.0 brought little evidence of critical reflection. Studies have claimed that social media can have positive effect on English training (Kabilan, Ahmad, Abidin, 2010), while Maragaryan et al. (2011) found that engineering students followed more lecturers' teaching approaches than using digital technologies. We find a body of studies which has explored user patterns (e.g. Robelia, Greenhow, Burton, 2011; Silius, Miilumäki, Huhtamäki, Tebest, Pohjolainen, 2010). Researchers find that SNSs used in educational context is more about student socialising than following course objectives (e.g. Junco, Cotten, 2012; Madge et al., 2009; Nykvist, Daly, Ring, 2010; Price, 2011; Wodzicki, Schwämmlein, Moskaliuk, 2012). Other studies have investigated how students manage different types of online identities, and its associated politics and practices (e.g. Mallan, Giardina, 2009; Mazman, Usluel, 2011; Selwyn, 2009).

Some studies have analysed how students use web 2.0 applications as part of their studies. Hrastinski and Aghaee (2012) found that university students used very few social media tools that could support their learning. They used social media to ask general questions, coordinate group work, and share work files. Hung and Yuen (2010) found that use in classroom teaching indicated the development of a stronger sense of connectedness among students, but had its basic role as a *supplementary tool*. Veletsianos and Navarrete (2012) found that students enjoyed using ELGG, but that participation to course-related and graded activities, showed little degree of networking, sharing, and collaboration. Grosseck et al. (2011) found in their study that the majority of the students spent significant time on Facebook. They engaged more into private matters than concentrating on the academic tasks at hand, even if they took part in discussions about their assignments, lectures, and shared information about research resources. In other words, we can infer that these studies yield the limited success of social media's usefulness, in terms of enhancing students' learning ability and user-acceptance in education, for example.

3 Methods and Data Sample

All Norwegian youth between 16 and 19 are entitled to attend high school education, which normally follows a three-year study programme. Future students can choose between general studies and vocational studies. General studies is a three-year education that prepares for university studies. In vocational studies, students can choose between different sub-programs. It follows a so-called "2+2 model", involving that the two first years are theory orientated, while the two following ones are organised around apprenticeship in a company. The study's data sample, however, was collected at a rather large high school in Trondheim, Norway, from January to March 2012, which offers both general and vocational studies. The high school has digital competences as a priority area. The students were recruited from one class in general studies and another in vocational studies. The students were digitally informed and were well-versed in use of computers and social media software. The sample is *not* a representative population, reflecting all Norwegian high school students.

The research design followed an explorative approach. It is rooted in a qualitative research method. 26 students were interviewed by use of qualitative indepth interviews, 17 boys and 9 girls. I completed 12 interviews, 10 in groups consisting of pairs to 4 students. Two interviews were completed individually, meaning a face-to-face conversation between me and the student. All interviews were conducted at the premises of the high school. The interviews lasted from 20 minutes to an hour. All interviews were semi-structured and explorative, following a guide with predefined questions. I asked the students about their user experiences. I asked if they used social media to organize themselves in online communities, in order to share formal school-work or to carry out informal learning. After I completed the interviewing, however, I transcribed them. I started looking for patterns. To complete this data analysis strategy, I was inspired by the sociological technique *constant comparative method* (Strauss, Corbin, 1990; 1998). I performed an open-ended approach, where I coded and grouped the students' answers into larger themes. The results from my coding are the five themes, which constitutes the data analysis. The study's informants are listed in table 1.

Table 1: The case study's informants.

Interview no.	Form of interview	Informant no.	Gender	Age	Approx. Facebook Friends	Member school FB-group	Active bloggers	Google Doc	Skype	Study program	Subject	Level in school	Date
1.	Group	1.	M	17	900	-	-	-	Y	Voc	Eng	2nd	Jan 2012
		2.	M	17	600	-	-	-	Y				
2.	Group	3.	M	17	350	-	-	-	-	Voc	Eng	2nd	Jan 2012
		4.	M	17	50	-	-	-	-				
3.	Group	5.	M	17	-	-	-	-	-	Voc	Eng	2nd	Feb 2012
		6.	M	17	800	-	-	-	-				
4.	Group	7.	M	17	-	-	-	-	-	Voc	Eng	2nd	Feb 2012
		8.	M	17	-	-	-	-	-				
5.	Group	9.	M	17	400	-	-	-	-	Voc	Eng	2nd	March 2012
		10.	M	17	-	-	-	-	-				
6.	Ind.	11.	M	17	300	-	-	-	-	Voc	Eng	2nd	March 2012
7.	Group	12.	M	16	600	-	-	-	Y	Gen	Spa	1st	Feb 2012
		13.	M	16	700	-	-	-	Y				
		14.	M	16	-	-	-	-	Y				
8.	Group	15.	F	16	1.000	Y	Y	Y	-	Gen	Spa	1st	March 2012
		16.	F	16	300	Y	Y	Y	-				
9.	Group	17	F	16	700	-	-	-	-	Gen	Spa	1st	March 2012
		18.	F	16	400	-	-	-	-				
		19.	F	16	800	-	-	-	-				
		20.	F	16	1.000	-	-	-	-				
10.	Group	21.	M	16	700	Y	-	-	-	Gen	Spa	1st	March 2012
		22.	M	16	-	-	-	-	-				
11.	Ind.	23.	M	16	200	Y	-	-	-	Gen	Spa	1st	March 2012
12.	Group	24.	F	16	200	Y	-	-	-	Gen	Spa	1st	March 2012
		25.	F	16	200	Y	-	-	-				
		26.	F	16	300	Y	Y	-	-				

4 Data Analysis – Findings

The data analysis builds on the user experience of 12 students, which covers seven males and five females. These are informants 1, 2, 12, 13, 14, 15, 16, 21, 23, 24, 25, and 26. 10 attended general studies and 2 vocational studies. Based on their personal user experiences, I have categorized their answers into five larger themes. Each theme outlines how they evaluate and organize schoolwork, moreover, if they use social media to cooperate, share, and get feedback on their formal schoolwork from peers. The themes are aimed at answering the article's main question; how do students evaluate and use social media to organise formal schoolwork? Each theme also aims at showing a conformist user behaviour, suggesting to be characterised by reflective decision-making processes, exposing selective user participation.

The first theme explores how they evaluate their online ties, reflecting that students are rather sceptical regarding who they bond with on SNS. The second connects to how they establish Facebook groups, which works as a type of "class bulletin boards". The third shows how students produce learning tools, and how they actively decide not to share them with co-students. The fourth scrutinises how students use Skype, as a way to cheat on their homework. The fifth theme tells the story of how Facebook groups take on a larger role. It is a discussion and coordination site, to complete larger project work submitted in the collaborative tool Google Docs.

4.1 Theme 1: The social selection of online ties

The first theme characterising the student learning ecology, concerns ideas and practices related to social selection processes in social networks. How social actors choose their ties, for example, has implications on access to potential resources. The students interacted between several social media software, some that are "social", like Facebook, while others are mere content pages, like blogs and YouTube. The latter ones did not give access to new ties. Facebook was widely used, however, and "faceworking" was not new. It is an ongoing reflection process. Requests and ties are continuously up for review. Students had large Facebook networks, on average between 400 to 500 ties, working as a standardisation. It seems that personal Facebook networks were "normalised" around there. Some had as many as 1.000 connections, but admitted they did not know everyone. Many claimed they knew all their ties, but some had reversed this. One female student had unfriended 700 ties from 1.000 to 300.

The students reported putting on a conformist "guard", as they were now more concerned with rejecting than including new ties.

In other words, inclusion and exclusion to social networks, and the blurred role between the on-line and off-line worlds, was a factor. It worked as a significant precondition for participation, as well as creating multiple divisions between students. These followed the lines of independent variables, like age, gender, and study programs. The students in vocational studies, for example, took on a very "conservative" stand. They did not share any type of assignments with co-students, involving very low prospects for student collaboration. Only two male students did so. If they shared, it happened in small networks, consisting of two or three ties, often within the limit of a one-to-one relation. The male students in vocational studies preferred submitting assignments on the school's Learning Management System. They expressed considerable scepticism to share schoolwork on social media. Privacy was an issue. They had an individualised approach. They considered that sharing should only be carried out under the strictest confidentiality, mainly as an off-line relationship between student and teacher in a private physical space.

Students in general studies had a different attitude. They used Facebook and Skype to goal-orientated activities. This applied to at least 10 students, implying higher probability for student collaboration. Yet, there are user patterns showing layers of division and low degree of transparency. All Facebook groups, for example, were closed. They were established for different reasons. Some were class-based, others were created as part of project work in distinct subjects. Facebook groups has also been created around distinct subjects they studied. The students published different content too, ranging from practical information, to take on the role as discussion forums. Resourceful students created them and took on the role as administers. Many students explained that they had been added without their consent, but somehow started using them regularly. There were at least four to five Facebook groups.

The Facebook groups were in fact off-limit area to teachers, involving that none of teachers had taken any role in creating them. The students had very clear opinions, on who should have access. If the teachers, for example, took on a very active role in orchestrating how they should work and what type of content should be shared, it would involve lower probability of use. The students needed an "online backstage", a site where they can do their school work and not having their teacher peaking over their shoulder. As this female student explains:

I-24: "They could have written that, this was something you should have paid attention to in class. And, you have to be friends with the teacher, if they are to be member of the group. And I don't think that there are many who are friends with the teachers."

4.2 Theme 2: Facebook as a "class bulletin board"

The second theme in the student learning ecology, nevertheless, is to consider what role Facebook groups can take. Facebook groups are often framed as a "class bulletin board". Once groups were established, they took on a practical and coordinating role. Sharing was not based on discussions of assignments, such as increasing knowledge on a distinct topic, but to keep oneself updated. Students in general studies used the groups in this way. None of the students in vocational studies reported using or being member of anyone. The data suggests four to five closed groups, where at least three were class-based. Students emphasized that they were useful. The class-based were mainly used within three areas: (1) as bulletin boards, (2) to inform about homework, and (3) to share cram sheets as part of preparation for tests. These female students explain:

R: Are you member of a Facebook group?
I-24: Yes. We have a class group. There we talk about what homework we have and what tests we are going to have, stuff like that.
R: Are you active in one of those? I have understood that it is not created by a teacher, but by you guys?
I-24: Yes, to remind each other that we have tests. It is very smart.
R: Is this a bulletin board or do you have discussions about assignments?
I-25: No, not about topics.
I-24: It is like that, if someone has homework, and has forgotten what pages we are supposed to read for a class, then you post what page we are supposed to read in science, and then there is someone who writes it if they know it.

The transcript indicates that sharing is about obtaining practical information as part of preparations for future classes. Students share information on what they have in homework for the next class, which pages they are supposed to read for a particular lesson, for example. Sharing is not based on a motivation to participate in a reflective process with the aim of turning data to knowledge on a distinct topic. Sharing is individual and rarely based on collaboration. The Facebook groups are a sort of a "student answering service", where communication is individual, but public, with the expectation of a short answer. There is a low threshold for sharing. Anybody can post anything without having the risk of being bullied. The exchange is a supplement to regular reminders students do face-to-face. This aspects, perhaps, reminds much of the old "work plan", a sheet, which teachers handed out to students at the beginning of each week describing designated workload. Cram sheet, however, is a popular digital item:

> *I-21: "We have a class group, we have an own Facebook group. When we have tests, for example, we can share cram sheets. If there is someone who has not done their homework, then we can share, so we can talk to each other, what is our homework for the next day, what is the work for the next week. In that sense, it is very convenient."*

4.3 Theme 3: Production of learning tools – the cram sheet

The third theme of the student learning ecology, however, concerns the creation and sharing of a popular user-generated item, the cram sheet. The creation and sharing of cram sheets, reflects how students embed or transfer a learning strategy, which aim at reproducing formal knowledge and carry out a goal-driven activity in the online world. Cram sheets can be classified as a concise set of notes of compressed knowledge used for quick reference. Students use them as part of their preparations for tests and exams, inasmuch as a method to memorize formal knowledge in any given subject they are enrolled in. Creating them is also an exercise, as learners have to perform some degree of work by themselves. Modern students often turn to the Web and retrieve them there. But there is a catch. The Web's complexity means that there are unknown quantities of cram sheets in global circulation. Students will often face a reoccurring problem: cram sheet overload. The relevant and accurate one, which covers the exact material for the test at hand, can be hard to find. If not found, they must be produced and shared by someone, a piece of workload which someone has to complete. This male student explains:

R: What's going on there?
I-23: Everything about what we have in homework, when classes start, cram sheet, tests, and what the tests are about.
R: Do the students share their schoolwork very actively there?
I-23: Yes, a lot. It is mostly those who don't bother studying and who don't bother do well at school, who ask if others can post cram sheets. I do not post my cram sheets there.
R: What is a cram sheet?
I-23: We often have a topic related to our tests. Everything that we have in a specific topic, we write down on a sheet, which is important to know. So, it is almost like a summary of what we are going to have on the next test.
R: Is this a method that you created or developed by yourself?
I-23: It is almost as taking notes in class, where you write what you feel is important to know. I use cram sheets a lot. Mostly, I use when I browse through what we have read in the textbook, I read through it, and write down what's important.
R: Is this a method you learned in school?
I-23: Yes.
R: Are you careful about sharing cram sheets on Facebook?
I-23: Yes, I think it is too easy. I think that they ought to figure it out by themselves and organise their own cram sheet. They only dodge work.
R: Because you are really doing the work for them, right?
I-23: Yes. I will not do the free work for them. I have worked hard on this and I will not just give it away.
R: Are there many asking for cram sheets?
I-23: It is the same who ask. They rarely post cram sheets themselves.
R: There is somebody doing that?
I-23: Sometimes there is.
R: Are there anyone who are more active in this Facebook group than others?
I-23: Yes. Those who don't pay attention in class, those who need more info.

The transcript shows that non-publishing is a moral belief and a decision, identifying rigid distinctions and labelling of co-students. In our case, "those who need more info". Students requesting such items, are ascribed the social identity or the role as "free riders", a type of student who attempts benefiting from a learning resource without repaying own requests. They try profiting from others' work, a type of "student opportunist", and are more or less understood as disloyal. They would seldom repay a social gift and try to escape responsibilities and obligations. Non-sharing does not encourage to constructive student interaction or collaboration. One can easily sympathize with the student. Non-sharing displays defined norms or values commonly seen when items are exchanged. If one is to share, the student has an awareness that formal learning should imply a symmetrical value in a relationship. If something is being given away, it creates an expectation of reciprocity, or that something is returned. The male student knows that blind sharing is to make it easier for a student type category, who breaks with the acceptable standard for good student collaboration. If he gives away his cram sheet, he will probably get little in return. Consequently, it is not better to share.

4.4 Theme 4: Using Skype to cheat on homework

The fourth theme shows how social media is used for what can be classified as a non-constructive learning activity, in terms of possessing good learning strategies. Social media is used to *cheat* on homework, but also reveals students' ingenuity and creativity in reengineering online resource management practices. Such web practices are seldom intended at retrieving information from the web for critical reflection, in order to create indepth understanding of a topic. They are merely collaborative practices, where students use online ties from Facebook or Skype, to quickly manufacture and reproduce a digital item with as little work as possible. This applied especially when students needed doing their homework in a hurry, a practice they referred to as "last minute work". These male students explain:

R: Do you use Skype to do schoolwork?
I-12: That too. To send files.
R: What kind of files are that?
I-13: Homework.
I-12: Among other homework. Collaboration assignments, for example. One writes something on one computer and then sends it to the others.
R: Is there a Word file?
I-14: Yes. Anything, really.
R: Is there someone who writes a document, and then circulates it?
I-13: It happens sometimes.

R: Who starts writing the document?
I-12: It varies.
I-13: It is often those who are quite structured.
R: Is it you guys?
I-13: Yeah.
I-14: Yes, you might say that.
R: Is that within your network? Is it so that one starts to write, and then some other adds more? Is that how it works?
I-12: No.
I-13: It's like "last minute work". If it happens that your teacher is going to check your homework, then you get it one minute before you have to show it.
R: But can't the teacher identify this?
I-14: No.
I-13: No. They just look at the assignment.
R: Is this something you learned here, at this school?
I-14: We got the laptop this year.
I-13: PC was not as that "cool" in junior high school.
R: It wasn't?
I-12: No. It was first in high school that we got our own laptops.

It is commonplace that students manipulate homework; it is a type of an ancient and well-played educational "ritual game of deception" between students and teachers. Teachers are familiar with that students try to deceive them, into believing that they have completed their homework assignment. Skype is a form of student collaborative "back stage", to perform superficial cosmetic work on formal schoolwork. The male students circulate one similar digital item on their "backstage", which on their "front stage", is portrayed to be the individual work of one student, when it is fact not. Skype is used as a tool to perform a type of impression management strategy, a social role play, that the students have done their "job". The practice is doubtfully constructive in fostering good study habits, but shows that students use ties in the student learning ecology to modify "cut and paste" practices. We see that students abide to some sort of code, norm or value, which still questions if homework has an educational value.

4.5 Theme 5: Facebook and Google Docs for learning

The fifth theme illustrates an advanced web mediated practice. It can be classified as innovative, and shows a way on how students should work in the interaction between learning and new web technologies. The practice is a blended approach, where formal knowledge is supposedly formed in the intersection between face-to-face interaction and digital space. It is a constructive learning practice and is not based on investing minimal efforts in order to get the "job done". It reflects that online exchange is part of interaction, collaboration, and reflection, where web content is retrieved and transformed into some type of formal knowledge, or, perhaps, students are connecting pieces of sources in order to create formal knowledge outside the mind. The intent is to create, sensemake depth, and process data to some sort of formal knowledge through social interaction. It can be argued that it constitutes a practice where students attempt expanding their knowledge on an already *established* socio-cultural experience.

This trait becomes clearer when looking on what type of digital content is retrieved, processed, and produced, and what role social media plays in this regard. Very few students used social media to this purpose. Female students in general studies, for example, established temporary Facebook groups, which were part of a larger cross-disciplinary project, which covered the work realms of different teachers and subjects. The groups were operational as long as the projects lasted. The female students reported that they sent links to each other, and actually discussed the project's purpose, thus relating social web to a goal driven learning activity. Facebook updates are answered with comments, where one gets the glimpse of a participatory culture, involving that SNSs are used as *discussion forum*. This transcript from one of my interviews shows the point in case:

I-15: We had a group project, "2050 Trøndelag", on how Trøndelag is going to be in the year 2050. There we had a Facebook group, where we discussed what we were writing, what we should put in our project, what was relevant to have, and things like that.

R: Who was most active in that group? Was it you?

I-16: No, it was not a big group. All contributed. We were five students, but the fifth did not do much. We were contributing all together. And we used Google Docs.

Facebook is used in combination with the web-based office suite software Google Docs, which means that we see a type of parallel processing of two distinct web practices. Facebook groups act as a *work* and *coordination* site, while Google Docs is the tool that documents the assumed transformation of info to knowledge, as well as being the end-deliverable for grading. There is a hidden parallel web practice; they used SNSs and collaborative real-time online writing and edit software together. Other students demonstrated similar user patterns, but instead of writing in Google Docs, they sent working documents on e-mail to each other. Students demonstrating this user pattern, however, are autonomous and very self-organised. They appear mastering the complexity and chaos of the current Web, and are well-versed in writing and reading texts, beyond the mere firm reproduction of scattered information. They possess a reflective and critical skill, which aid them to tell the differences in quality of what information is relevant and not. They manage the conversion of data to the logics of formalised knowledge, or, comply with the intent of goal-driven deeds. They can modify and interpret web content, beyond "cut and paste" or retrieving. This collaborative web practice, for example, suggests to have helped one of my informants in her learning. She explains:

> I-15: "And when all of us were going to contribute in the written part, I was very nervous, because I'm not so good in writing Norwegian. And then I sent it to the people in the group, so that they could see through it, what I should write more about or what was wrong. Just to be sure it was correct what I had done. So I got good feedback. It helped me a lot that we had a Facebook group. I got to hear 'it was awesome, but I could imagine that you wrote a bit more about fish farming on Salmar too.' And then I wrote a bit more about that. And the other would look at it and then it was time to hand it in."

5 Conclusion

Recent educational research on students' use of social media appears producing contradictory results, especially regarding the question if it represents a constructive learning resource in formal learning. Greenhow and Robelia (2009), for example, found in their research that students used it for such purposes, while others argue it might be a positive asset, foremost as a supplementary in classroom training (Hung, Yuen, 2010). On the other side of the axis, researchers have uncovered little solid evidence that social media fosters collaborative learning (e.g. Madge et al., 2009; Selwyn, 2009). Social media struggle to be perceived as a potential learning resource, a tentative belief that gets more legitimacy when students report that they prefer the "old way" (Hrastinski, Aghaee, 2012). Research has also indicated that "to get social media to work", instructors are forced to instruct students and oversee that it is used at all (Veletsianos, Navarrete, 2012). This implies that use of social media in formal learning still has a long way to go, implying that some researchers question if youth as "digital natives" is a myth or reality (Margaryan et al., 2011).

The application of Baron's (2006) "learning ecology" is an attempt to introduce a user perspective on students' use of social media. I have attempted emphasising that students exercise reflective decision-making processes, showing strong selective user participation. Students apply different strategies in how they choose to involve themselves in digital social learning environments. There are certain internal dynamics in students' user behaviour, which are reflected in the five themes, that future research should perhaps address. Social media is used for constructive and non-constructive learning. As only half of the data sample uses it to fulfil a learning objective, this case study only contributes to reconfirm what previous research has taught us; there is still a long way to go, in order to get social media to be a tool that fosters collaborative learning. Only a few does, female students in general studies.

There are obvious research limitations with this case study. One cannot gather valid conclusions from one single case study. The study does not claim to be representative for how all Norwegian high school students use social media in formal learning either. It is only an explorative. The study supports tendencies seen in current research, but poses some indication to where futures studies should set their focus. There is need for longitudinal research, in addition to address gender issues. My main concern has been to introduce a more solid user perspective, which can theoretically can cast light on the subject matter. And, perhaps a place to start is to go further into Barron's (2006) "learning ecology", as I have done here.

References

Aalen, I. (2013). *En kort bok om sosiale medier*. Bergen: Fagbokforl.

Barron, B. (2006). Interest and self-sustained learning as catalysts of development: A learning ecology perspective. *Human Development*, *49*(4), 193–224.

Boyd, D., Ellison, N. B. (2007). Social network sites: Definition, history and scholarship. Journal of Computer-Mediated Communication, *13*(1), 210–230.

Drouin, M. A. (2011). College students' text messaging, use of textese and literacy skills. *Journal of Computer Assisted Learning*, *27*(1), 67–75.

Ellison, N. B., Steinfield, C., Lampe, C. (2011). Connection strategies: Social capital implications of Facebook-enabled communication practices. *New Media and Society*, *13*(6), 873–892.

Engeström, Y. (1987). *Learning by expanding: an activity-theoretical approach to developmental research*. Helsinki: Orienta-Konsultit.

Greenhow, C., Robelia, B. (2009). Old communication, new literacies: Social network sites as social learning resources. *Journal of Computer-Mediated Communication*, *14*(4), 1130–1161.

Grosseck, G., Bran, R., Tiru, L. (2011). Dear teacher, what should I write on my wall? A case study on academic uses of Facebook. *Procedia Social and Behavioral Sciences*, 15, 1425–1430.

Hamid, S., Chang, S., Kurnia, S. (2009). Identifying the use of online social networking in higher education. In Atkinson, R. J., McBeath, C. (Eds.), *Same places, different spaces. Proceedings ascilite Auckland 2009. 26th Annual ascilite International Conference, Auckland, 6–9 December 2009* (419–422). The University of Auckland, Auckland University of Technology, and Australasian Society for Computers in Learning in Tertiary Education.

Hrastinski, S., Aghaee, N. M. (2012). How are campus students using social media to support their studies? An explorative interview study. *Education and Information Technologies, 17*(4), 451–464.

Hung, H. T., Yuen, A. C. Y. (2010). Educational use of social networking technology in higher education. *Teaching in Higher Education, 15*(6), 703–714.

Irwin, C., Ball, L., Desbrow, B., Leveritt, M. (2012). Students' perceptions of using Facebook as an interactive learning resource at university. *Australasian Journal of Educational Technology, 28*(7), 1221–1232.

Jenkins, H. (2006). *Confronting the challenges of a participatory culture: Media education for the 21th century*. Retrieved from www.digitallearning.macfound.org

Junco, R., Cotten, S. R. (2012). No A 4 U: The relationship between multitasking and academic performance. *Computers and Education, 59*(2), 505–514.

Kabilan, M. K., Ahmad, N., Abidin, M. J. Z. (2010). Facebook: An online environment for learning of English in institutions of higher education? *Internet and Higher Education, 13*(4), 179–187.

Kaplan, A. M., Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons, 53*(1), 59–68.

Lave, J., Wenger, E. (1991). *Situated learning: legitimate peripheral participation*. Cambridge: Cambridge University Press.

Luckin, R., Clark, W., Graber, R., Logan, K., Mee, A., Oliver, M. (2009). Do Web 2.0 tools really open the door to learning? Practices, perceptions and profiles of 11-16-year-old students. *Learning, Media and Technology, 34*(2), 87–104.

Madge, C., Meek, J., Wellens, J., Hooley, T. (2009). Facebook, social integration and informal learning at university: 'It is more for socialising and talking to friends about work than for actually doing work'. *Learning, Media and Technology, 34*(2), 141–155.

Mallan, K., Giardina, N. (2009). Wikidentities: Young people collaborating on virtual identities in social network sites. *First Monday, 14*(6).

Margaryan, A., Littlejohn, A., Vojt, G. (2011). Are digital natives a myth or reality? University students' use of digital technologies. *Computers and Education, 56*(2), 429–440.

Mazman, S. G., Usluel, Y. K. (2011). Gender differences in using social networks. *Turkish Online Journal of Educational Technology, 10*(2), 133–139.

Nykvist, S., Daly, C., Ring, J. (2010). Everyone wants web 2.0. In Wong, S. L. et al. (Eds.), *Proceedings of the 18th International Conference on Computers in Education*. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

O'Reilly, T. (2005). *What Is Web 2.0? Design patterns and business models for the next generation*. Retrieved from http://oreilly.com/web2/archive/what-is-web-20.html

OECD. (2014). *Recognition of Non-formal and Informal Learning*. Retrieved from http://www.oecd.org/education/skills-beyond-school/recognitionofnon-formaland-informallearning-home.htm

Price, J. (2011). That's what friends are for: Creating an online community of, and for, first year student to increase retention. In Williams, G., Statham, P., Brown, N., Cleland, B. (Eds.), *Changing Demands, Changing Directions. Proceedings ascilite Hobart 2011*. (1032–1038).

Robelia, B. A., Greenhow, C., Burton, L. (2011). Environmental learning in online social networks: Adopting environmentally responsible behaviors. *Environmental Education Research, 17*(4), 553–575.

Selwyn, N. (2009). Faceworking: Exploring students' education-related use of Facebook. *Learning, Media and Technology, 34*(2), 157–174.

Silius, K., Miilumäki, T., Huhtamäki, J., Tebest, T., Pohjolainen, S. (2010). Students' motivations for social media enhanced studying and learning. *Knowledge Management and E-Learning, 2*(1), 51–67.

Strauss, A., Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: Sage Publications, Inc.

Strauss, A., Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: Sage Publications, Inc.

Veletsianos, G., Navarrete, C. C. (2012). Online social networks as formal learning environments: Learner experiences and activities. *International Review of Research in Open and Distance Learning, 13*(1), 144–166.

Vygotskij, L. S. (Ed.). (1978). *Mind in society: the development of higher psychological processes*. Cambridge, Mass.: Harvard University Press.

Wodzicki, K., Schwämmlein, E., Moskaliuk, J. (2012). "Actually, I Wanted to Learn": Study-related knowledge exchange on social networking sites. *Internet and Higher Education, 15*(1), 9–14.

Biography

Halvdan Haugsbakken is PhD Candidate in Sociology at Dep. of Sociology and Political Science, at the Norwegian University of Science and Technology. He is M.Sc. in Social Anthropology, and has worked as Research Scientist at SINTEF Technology Society.

ICT Competencies for School Students

Anthony Jones
Graduate School of Education
University of Melbourne, Australia
a.jones@unimelb.edu.au

Abstract: This paper discusses results from a small-scale research study, together with some recently published research into student perceptions of ICT for learning in schools, to consider relevant skills that do not appear to currently being taught. The paper concludes by raising three issues relating to learning with and through ICT that need to be addressed in school curricula and classroom teaching.

Keywords: Learning with ICT, student perceptions, student experience

1 Introduction

> In my vision, *the child programs the computer* and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building. (Seymour Papert, 1980, p. 5)

When this author began introducing secondary school students to computer technology through using a computer program to perform calculations, the students quickly learned the consequences of illegible or unclear handwriting, and illogical or incomplete planning. The year was 1968 and the Year 10 mathematics students were being introduced to the FORTRAN programming language. Students did not see or physically interact with the computer as it was located at a nearby tertiary institution. Instead they wrote instructions on a coding sheet that the teacher delivered to data entry staff at the tertiary institution who converted the written lines of FORTRAN into code that could be processed by the computer. The data entry staff were not programmers, but instead were trained to use a keyboard to accurately copy the handwritten symbols, words, lines and spaces from a coding sheet and enter them into a machine that that turned each line of a coding sheet into a punched card, with a printed copy of the entered instructions at the top of the card. These cards, together with appropriate first and final cards, were placed in line to be processed by the computer. Several days later the teacher would collect small bundles that consisted of cards wrapped in a printout for each individual student program.

For the first two or three coding sheets that each student created the teacher painstakingly proof-read and suggested corrections before submitting them to the data entry staff. One mathematics period each week was set aside for programming, and this always began with print-outs and punched cards being returned to the students. During the initial lessons the problems to be coded into the programming language were developed on the blackboard through class discussion and then copied onto the coding sheets. This approach resulted in most students successfully completing the set task. However as soon as students were asked to work individually to solve a problem by converting it into FORTRAN code and running it on the computer the number of errors increased exponentially. When students received a print-out indicating that the computer could not process a line of code, they were taught to identify the type of error – initially categorising errors as either theirs or data entry.

While it is not the intention in this paper to concentrate on the historical aspects related in the previous paragraphs, some of these aspects, together with the quotation from Seymour Papert, will be used to illustrate and suggest some key competencies and skills that students and many teachers do not appear to associate with educational use of ICT in 2014.

Many researchers have investigated the use, lack of use, and misuse of a variety of ICT in school classrooms. A large proportion of these studies have considered class-room use of ICT from the perspective of teachers, but fewer have considered educational use of ICT from the perspective of students. This paper reports on a small-scale study conducted towards the end of 2011 in a single school, and considers the actions and attitudes of approximately one hundred and seventy students in grade 4 or 6 at an Australian urban primary school.

2 The Study

In mid 2011 all the grade 4 and grade 6 students at a state government funded primary school in Melbourne, Australia, participated in a study that was intended to focus on a comparison of results from two mathematics tests – one presented and completed on paper and the other presented and completed online. The results and comparisons have been reported previously (author) and will not be detailed here. Over a period of two weeks following the completion of both tests, students in each of the seven classes involved were interviewed. The interviews were conducted in a classroom by a researcher and with the class teacher present. Students were asked to think about the two mathematics tests and express their opinions and feelings about them individually and collectively. The interview sessions were video recorded.

While the school followed the curriculum set by the state education authority, teachers were given no explicit instructions about what or how to use ICT for teaching and learning. While the curriculum used in 2011 has been replaced, the approach to ICT in the new version is the same. The following quotation from the online introduction to ICT shows the general approach.

> ICT, an interdisciplinary domain, focuses on providing students with the tools to transform their learning and to enrich their learning environment … Learning in this domain enables students to focus on the task to be accomplished rather than on the technology they are using to do the work. (AusVELS, 2013)

This curriculum has three subdivisions: ICT for visualising thinking; ICT for creating; and ICT for communicating. The extended descriptions for each subdivision use generic terms such as 'graphic organisers' and 'ICT tools' and provide teachers with no substantial assistance or direction. If every classroom teacher was adequately trained and practiced in using ICT with students, such a lack of curriculum detail might not be critical. However the Journal of Australian Educational Computing (2012, 27(2)) published a special issue devoted to a major Australia wide research project into pre-service teachers and ICT. This showed that across Australia there are concerns about the deficiency in many ICT key competencies demonstrated by teachers and those preparing to become teachers.

Data being reported came from a research project in which primary school students in grade 4 and grade 6 were interviewed in groups about two mathematics tests that had recently completed, one paper-based and the other online. Four of the 7 classes sat for the paper-based tests simultaneously, working in their usual classrooms. However the online tests were completed one class at a time in a computer room because the school did not have a sufficient number of computers to allow simultaneous testing. On the day that 4 classes attempted the paper-based test, the other 3 classes did the online test. A fortnight later the roles were reversed and each student attempted the alternative version of the test.

Whenever testing was underway a researcher moved between the rooms to observe and to respond to any questions that arose. Both the paper-based and online tests were supposed to be treated as parts of the formal assessment process by students and staff. However video recordings made while students sat the online test clearly indicate a difference in attitude among both students and teachers between the two types of test. While a full comparison of the results of the two tests is not relevant to this paper, the time taken by students to complete them is.

Table 1: Times for tests

Test	Allocated time	Average time spent on task
On paper	45 minutes	45 minutes
Online	45 minutes	14 minutes

One of the major reasons for not attempting to analyse and compare results between the two tests is provided in the following paragraph. Grades 4 and 6 were chosen for this study because they had completed the Australian national testing (NAPLAN) in the previous year, and it was hoped to compare the mathematics scores from this testing with results from this research. For most students there was a strong correlation between their NAPLAN mathematics score and the paper-based test score. However the online test results were significantly lower than the other test scores.

When researchers analysed the video recordings of students taking the online test it was noted that students spent much less time completing the test, and did not appear to go back and check answers before logging off. Almost all students took the full 45 minutes allocated for the paper-based test, but the same students averaged only 14 minutes for the online version. One complicating factor is this comparison of the time spent on task is the fact that there are fewer items in the online version, but not enough to account for such a dramatic time difference. The producer of both tests, the Australian Council for Educational Research, set the allocated time and stated that the tests were equivalent in that they assessed the same mathematical concepts at the same levels.

Following analysis of the video recorded class discussions and some focus group interviews with teachers and one researcher, it appears that both students and teachers had very different attitudes to the two types of test. Many of the students, together with several teachers, commented that they saw the online test as of having less importance and value than the paper-based test. Some students indicated that they considered the online test as a type of computer game, and so they went through it as quickly as possible and did not bother to go back and check because that isn't done in games.

3 Discussion

This section commences with some brief thoughts about equitable access to the internet through various types of digital technology. It then relates findings from several research surveys to some aspects of the current Australian ICT curriculum, before considering issues that arose from the research study.

We are aware that access to digital information and technologies has become ubiquitous for many people because they have access anywhere and anytime. Mobile devices have altered the means and difficulties accessing the internet both in and out of school. This change in access raises different types of issues relating to equity of internet access for all students. One issue that will not be discussed relates to differences between the number and types of technologies students have available to be part of their learning. In the school which participants in the study reported here attended, almost all the students reported having access to digital technology at home. However some had to share a computer and internet connection with older siblings who had preference. Others had their own computer, but without internet access. What were of more interest for this paper were the attitudes, beliefs, and abilities of the students to differentiate between personal use of social media and educational uses of digital technology, including the internet.

A UK study of years 8 and 10 students (Becta, 2008) looked at internet use in the home and in the classroom. Among other things the authors reported that the majority of students used the internet to read, play games, and communicate within a personal social network, and that "relatively few learners are engaging in more sophisticated Web 2.0 activities such as producing and publishing their own content for wider consumption" (Becta, 2008, p. 4). It was also reported that it was possible that the participants did not have the necessary technical experience to create and publish online, and that they were not aware of possible creative uses of the digital tools they were using. It was also suggested "that experience with user-friendly social networking technologies may encourage them to see Web 2.0 applications as services that they consume, rather than as tools that they can use to advance their own aims" (Becta, 2008, p. 4).

In the US, the Speak Up research project, one part of Project Tomorrow, has been surveying teachers and students about education and digital technology for more than a decade. In *From Chalkboards to Tablets*, their report based on data collected in 2012, notes that all forms of social media appear to be used away from the classroom. As has occurred in earlier surveys, this report notes that students comment on, and are dissatisfied with, the manner in which technology is used in classrooms. Specifically, it is claimed that 'dissatisfaction with using technology at their school is not about the quantity or quality of the equipment or resources; it is about the unsophisticated use of those tools by their teachers, which they believe is holding back their learning' (Project Tomorrow, 2013, p. 7). The report notes that this is very different to what was found in the early surveys. In the concluding section it is noted that

> *"Today, while access is still not universal for all students, for the majority of the students across all grades, their attention is on how to use a wide range of digital tools and resources to enable a highly personalized learning experience. This self-initiated evolution from access to personalization provides an interesting model for thinking about the adoption and adaption of emerging technologies within our schools." (Project Tomorrow, 2013, p. 24)*

That this is not only occurring among students in US schools is indicated by a much smaller Australian study reviewed by DERN (Digital Education Research Network). Among other things, the student participants aged 12–18 were asked about their attitudes to ICT learning. As with the Project Tomorrow report, the students perceived teachers to lack competence in using ICT for learning. Interestingly there appear to be differences based on age about teachers using more ICT in the classroom. Among younger students up to 88 % wanted more use of technology by teachers, while only 42 % of the 16–18 year old students wanted this.

Curriculum documents such as AusVELS ICT are vague in terms of curriculum content at grade levels for a variety of reasons. One reason that applies in Australia is that schools are free to choose the type and brand technology they purchase, and the resultant variety of hardware and software make it extremely difficult for education systems and authorities to specify classroom activities for teaching learning with ICT. One consequence of this lack of curriculum specificity is that a range of key competencies that are considered critical in other developed countries, by both education authorities and industry, are not included in the Australian curriculum.

One example of industry concern is the DERN (2013) research review mentioned previously, which was commissioned not only to investigate student perceptions of ICT in schools, but also their perceptions of ICT as a career path. In spite of almost every participant using social media for personal communications, and the general level of perceived personal competence with using ICT at school, only 31 % had considered ICT as a career and only 38 % of the 18 year olds thought studying IT would be interesting. It would appear that students sharply differentiate between out of school personal use of digital technology and educational applications of the same digital technology.

4 Conclusion

The purpose of this paper has been to argue that research appears to indicate that we are moving away from Papert's belief at the beginning of the paper. It is likely that for student users to control the technology for learning they require both instruction and practice. This implies a level of ICT knowledge and skills that many current teachers, and those preparing to become classroom teachers, do not possess. The small research project reported here, combined with the research reports discussed, give rise to several issues the teaching and learning of ICT in schools.

Over the past few years schools across the developed world have introduced tablets and iPads into the classroom. The author's experience with these devices is restricted to their application in primary schools with children aged 5 to 11 years. While a general observation has to be that students appear to enjoy using tablets, closer inspection suggests that often what students are doing on these devices is mindless, repetitive, and difficult to connect to stated curriculum learning goals, other than that technology is being used.

For both school-provided and BYOD (bring your own device) technology the programs (apps) that are used are determined by the school or education system. For many students this means that what they use and do on devices at home does not correspond with what they use and do in their classes. This is first and most important issue arising from this discussion: students lack meaningful experiences in using ICT for learning.

Records show that over time the number of students in a class has varied greatly. Aristotle appears to have interacted with one student at a time, something that still occasionally occurs in special circumstances. Teachers are trained to work with groups of students – a whole class or sub-group of a class. The smaller these groups become, especially when each group or individual is doing something different, the less capable and confident teachers become. For example, from the perspective of a teacher there are enormous differences between pairs of students in a laboratory conducting the same science experiment, and students working at a computer in pairs or on their own while they solve a problem or create an artefact of their own choosing. This is a second issue: the majority of classroom teaching occurs in classes or groups, but when a student is expected to learn using ICT it often needs to be an individual activity. It appears that currently neither students nor teachers are successfully managing this transformation between modes of learning.

A third issue is one that is beyond the control of both students and teachers. The digital technology available for learning, both hardware and software, changes constantly. Students and teachers begin using a particular piece of ICT and almost inevitably it changes – either through the release of an updated version or through a new product that is different even though it might offer the same options. It can be disconcerting to be learning theorems of Euclidean geometry that have been known for thousands of years, or to be studying a play by Shakespeare that is centuries old, through making use of digital technology that is constantly changing.

In summary, it has been argued that in schools there are key competencies of ICT that have been, and are still being, lost because of the lack of specificity in all curriculum areas that invoke teaching and learning with and through ICT.

References

AusVELS (2013). *The Australian curriculum in Victoria*. Retrieved from http://ausvels.vcaa.vic.edu.au/Information-and-Communications-Technology/Overview/Introduction

Becta (2008). *Learners' use of Web 2.0 technologies in and out of school in Key Stages 3 and 4*. Retrieved from http://dera.ioe.ac.uk/1476/1/becta_2008_web2_learnersuse_report.pdf

DERN (2013). *Research reviews: Australian teenagers and ICT*. Retrieved from http://dern2.acer.edu.au/index.php/dern/research-reviews/P20

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. The Harvester Press Ltd: Brighton, UK.

Project Tomorrow (2013). *From chalkboards to tablets: the emergence of the K-12 digital learner*. California: Project Tomorrow. Retrieved from http://tomorrow.org/speakup/pdfs/SU12-Students.pdf

Biography

Anthony Jones is Senior Fellow of the Graduate School of Education at the University of Melbourne and Deputy Director of the International Centre for Classroom Research. He has taught in primary and secondary schools, and has been involved in initial teacher education and teacher development for more than three decades. After commencing computer use with a Year 10 Mathematics class in 1968, he has continued exploring learning and teaching with and about ICT.

Prior to becoming a teacher educator he taught in primary and secondary schools. Since the late 1960s he has focussed on both teaching and learning with and through ICT, with a particular interest in mathematics.

Supporting the Development of 21st Century Skills through ICT

Cathy Lewin, Sarah McNicol
Manchester Metropolitan University, UK
{c.lewin, s.mcnicol}@mmu.ac.uk

Abstract: The growing impact of globalisation and the development of a 'knowledge society' have led many to argue that 21st century skills are essential for life in twenty-first century society and that ICT is central to their development. This paper describes how 21st century skills, in particular digital literacy, critical thinking, creativity, communication and collaboration skills, have been conceptualised and embedded in the resources developed for teachers in iTEC, a four-year, European project. The effectiveness of this approach is considered in light of the data collected through the evaluation of the pilots, which considers both the potential benefits of using technology to support the development of 21st century skills, but also the challenges of doing so. Finally, the paper discusses the learning support systems required in order to transform pedagogies and embed 21st century skills. It is argued that support is required in standards and assessment; curriculum and instruction; professional development; and learning environments.

Keywords: 21st century skills, primary education, secondary education, pedagogy, innovation

1 Introduction

Developing 21st century skills and competencies in schools demands pedagogical shifts away from didactic approaches together with the embedding of ICT. Whilst the majority of school heads and teachers recognise the importance of ICT use for developing 21st century skills, ICT use in classrooms is still limited (EC, 2013). Indeed, 20 % of students in secondary education across Europe rarely use ICT during lessons with the majority of European teachers using ICT primarily for lesson preparation (EC, 2013). Moreover, where ICT is used it does not always lead to changes in pedagogical practices (Law, 2009; Shear et al., 2010; Luckin et al., 2011).

Innovative Technologies for an Engaging Classroom (iTEC) is a 4-year project on supporting the scaling up of pedagogical and technological innovation in classrooms across 19 European countries. The project has developed a process for creating educational scenarios and accompanying learning activities which describe new pedagogical approaches supported by a range of digital tools. The development of 21st century skills, such as collaboration, creativity and critical thinking, is a core aim.

This paper describes how 21st century skills have been conceptualised and embedded in the resources developed for teachers in the iTEC project. The effectiveness of this approach is considered in light of the data collected through the evaluation of the pilots. Finally, the paper discusses the role of ICT in supporting the development of 21st century skills.

2 21st Century Skills and the Role of ICT

21st century skills, sometimes referred to as 21st century competencies, is a complex term which encompasses skills that may be required to be successful in learning, in the workplace and to live effectively in the 21st century (P21, 2009; Binkley et al., 2012). Although frameworks and definitions of 21st century skills exist (e.g. ISTE, 2007; P21, 2009; Binkley et al, 2012), most refer to the same list of competences which includes collaboration, communication, ICT, creativity, critical thinking and problem solving (Voogt, Pareja Roblin, 2012). Voogt and Pareja Roblin (2012) suggest that as well as supporting teachers to change their pedagogy they also need to understand better how ICT can facilitate 21st century learning. As the development of 21st century skills can be enhanced through the use of ICT (Ananiadou, Claro, 2009), it is argued that their development should be cross-curricular, demanding changes in pedagogical practices (Voogt, Pareja Roblin, 2012).

It is recognised that the, "dramatic shift toward the economic aims of education" (Rutowski et al., 2011, p. 191) is one of the key drivers for the development of 21st century skills. Very few frameworks draw on educational research as part of their justification (Voogt, Pareja Roblin, 2012). This is acknowledged as a weakness, but a detailed discussion of these issues falls outside the scope of this paper. Furthermore, critics of the 21st century skills movement argue that the frameworks take no account of content (knowledge) and that the skills cannot be taught in isolation, proposing that the curriculum should be emphasised rather than the teaching of skills (see for example http://commoncore.org/mission; Mathews, 2009). However, Voogt et al. (2013) argue that, "most frameworks recognize the complex and cross-disciplinary nature of 21st century competencies and thus recommend integrating them across the curriculum" (p. 407). Indeed, many countries claim that the skills are already integrated although in some countries ICT skills are taught separately (Ananiadou, Claro, 2009). Furthermore, in a review of 10 frameworks, Mishra and Kereliuk (2011) suggest that developing sound content knowledge is included as an essential component of 21st century skills.

It is noteworthy that more recent frameworks attempt to broaden the notion of skills, referring, for example, to 'learning literacies' (Beetham et al., 2009); or to a 'Frame-work for 21st Century Learning', which incorporates skills, content knowledge, expertise and literacies (P21, 2009). This demonstrates that the importance of teaching 21st century skills as a holistic cognitive process, rather than as a set of discrete functional skills is becoming widely recognised.

The following section outlines each of the skills normally included within definitions of 21st century skills.

Critical thinking has been theorised, debated and defined by many (Kennedy et al., 1991) but broadly refers to making informed decisions on the basis of analysing, synthesising and evaluating information. When employed in a practical context, it is often referred to as problem solving. Both concepts are closely related to reflection, which is 'central to critical thinking and deeper learning' (Quinton, Smallbone, 2010, p. 126). The potential of portfolios and social media tools such as blogs supporting student reflection in school contexts has been noted (Crook et al., 2010).

Communication skills have been important in education for centuries (Voogt et al., 2013). Students need to have the ability to "exchange, criticise, and present information and ideas" (Ananiadou, Claro, 2009, p. 10). ICT has become an important tool for supporting communication both in education and also in a wide range of social practices (National Research Council, 2012) making it easier to reach a wide audience and communicate at a distance, faster and more ubiquitously. Students need to have well-developed communication skills in order to collaborate and work in teams. Collaboration is one of the skills clearly demanded by the twenty-first century workplace, particularly with the shift away from manual work (Dede, 2010). Team working is increasingly being facilitated by digital tools, which allow geographically dispersed team members to collaborate.

Creativity has long been considered an important component of education, however, since the 1990s, there has been a growing recognition of the importance of learner creativity. ICT can support learner creativity in many ways including developing ideas, making connections, creating and making (Loveless,

2002) with more recent developments such as web 2.0 increasing opportunities for creative activities.

Digital literacy does not simply refer to technical skills, but to "the ability to use information and communication technologies to find, evaluate, create, and communicate information, requiring both cognitive and technical skills" (ALA, 2011). It includes, for example, internet safety and an understanding of the ethical and legal issues relating to the access and use of ICT. Digital literacy is sometimes conflated with the closely related terms, information literacy and media literacy. In addition, other 21st century skills, for example, communication, collaboration and creativity are included as components of some definitions of digital literacy.

While the integration of these 21st century skills in classrooms is encouraged by theorists and policymakers, in practice, teachers often lack the skills and the space to teach their students 21st century skills (Voogt et al., 2013). Furthermore, their development requires substantial changes to pedagogical approaches and assessment practices (Binkley et al., 2012; Voogt, Pareja Roblin, 2012). The iTEC project has endeavoured to provide support and resources for teachers to enable them to develop their students' 21st century skills through new pedagogical approaches that make substantial use of technology. The resources have been designed to be generic and applicable to a wide range of curriculum areas, thus enabling the development of 21st century skills to be integrated within curricula.

3 How 21st Century Skills have been conceptualised and embedded in ITEC

iTEC was conceptualised as a project focusing on 'Learning in the 21st century' and innovation in the classroom through effective uses of technology. A process of visioning, development and piloting was put in place, developed over four cycles and subsequently produced as toolkits which stakeholders (including classroom teachers) used themselves in a fifth and final cycle of activity, thus shifting from a supplyled approach to a demand-led one. As part of the visioning process developed within the iTEC project to support teachers, socio-economic, technological and educational trends likely to impact on learning and teaching were identified and prioritised. The trends identified include references to 21st century skills both as a key trend but also embedded in the descriptors of other trends (for example, collaboration, problem-solving) (Cranmer, Ulicsak, 2011). The trends informed the development of educational scenarios (narratives of innovation in the classroom). The educational scenarios generated were prioritised, and a subset developed further. This prioritisation process was formalised through the development of scenario selection criteria to ensure that those put forward for further development were likely to lead to innovation in the classroom. One of the six dimensions of the selection criteria is 'Does the Scenario (or Learning Story) provide sufficient opportunities for learners to develop and demonstrate 21st Century Skills?' This exemplifies the perceived importance of 21st century skills in the project's conceptualisation of what makes a scenario innovative.

Next, through participatory design workshops with teachers and desk-based analysis of the scenarios, 'design challenges' were identified. Design challenges are potential barriers and issues that might prevent new pedagogical approaches and/or digital tools from being implemented in the classroom. For example in cycle 1, several design challenges such as 'Team work is not familiar' were identified in relation to team work/collaboration (Keune et al., 2011). Further analysis revealed a number of design opportunities or ways of overcoming the design challenges. Learning activities were then developed to address the design challenges and harness the design opportunities. These were pre-piloted and then a coherent package of 6–8 learning activities put forward for testing in large-scale pilots.

ICT is an integral part of each learning activity. Each package of learning activities presented to teachers is exemplified through a learning story, a narrative describing how the learning activities might be used in the classroom, supported by ICTs. Each participating teacher was presented with the package of learning activities and accompanying learning stories. They selected and adapted the ideas including the specific digital tools used, ensuring the implementation met their individual needs.

The learning activities created for cycle 4, exemplify the use of ICT and the underpinning 21st century skills.

- 'Dream': Introduce, understand and question a design brief using ICT to support reflection, team formation, collaborative editing, publishing and blogging.
- 'Explore': Collect information in relation to design brief using ICT to support browsing, social bookmarking, collaborative editing and media recording.
- 'Map': Create a mindmap to understand the relations between collected information using mindmapping software.
- 'Reflect': Record audio-visual reflections and feedback using ICT to support reflection.

- ‘Make’: Create a design using ICT to support media editing, programming, construction, 3D editing and 3D printing.
- ‘Ask’: Perform workshops with end-users using ICT for media recording.
- ‘Show’: Publish and present designs to an audience using ICT to support video editing, media recording, video publication and media sharing.
- ‘Collaborate’: Form ad-hoc collaborations with learners from other schools using ICT to support online discussion, media publication and blogging.

This is further exemplified in the following vignette from Spain in cycle 3. Firstly, the tools were set up, for example blogs and Dropbox[1] accounts, and students were instructed in their use, thereby developing their digital literacy skills. Then students searched for relevant games on the Internet, and evaluated each to gauge its advantages and disadvantages. This activity required both critical thinking and digital literacy skills. Based on this information, teams of students designed their own games, first on paper and then electronically using SMART Notebook. Working in teams required collaboration and communication skills; designing a game required creativity skills; and presenting their game in SMART Notebook required digital literacy skills. Students presented their games to the class for feedback, making use of communication skills and digital literacy as they used the interactive whiteboard to present. They used the feedback to revise their designs, using critical thinking and creativity skills, as well as collaboration. Throughout the project, students maintained a blog to reflect on, and evaluate, their experiences, developing critical thinking, digital literacy and communication skills.

4 Methodology

Each country has a national coordinator who oversees the project, supports teachers and co-ordinates data collection in their country. The evaluation was led by a team from a UK university.

At the end of each cycle, teachers completed an online questionnaire about their experiences, focusing on their use of the iTEC technologies, enabling factors, challenges encountered and potential for innovation. Teachers were only asked to comment on the impact of iTEC on 21st century skills directly in the cycle 4 questionnaire although there were questions on collaboration,

1 https://www.dropbox.com/home.

communication, creativity and ICT use in the questionnaires administered in cycles 1–3.

National coordinators conducted one or more case studies in their country each cycle, involving lesson observation and interviews with teachers, head teachers, ICT coordinators and students. They provided case study reports (cycles 1–3 only, two per cycle) or transcripts (all cycles, one per cycle) to the evaluation team. In cycle 4, national coordinators also conducted a focus group with a sample of teachers from their country. In addition, members of the evaluation team gathered data through observing project activities (e.g. training sessions, webinars). The focus of the evaluation altered during each cycle to adapt to the needs of the project, so the precise questions asked within the survey and interviews have changed, meaning direct comparison between cycles is not possible for all measures.

Qualitative data were analysed using Nvivo. Transcriptions were coded thematically using a conceptual framework from the SITES2 study (Kozma, 2003), modified to incorporate new codes to reflect emerging themes. The surveys comprised both open-ended and closed questions; the open-ended questions were translated into English using Google Translate and then coded, while the closed questions were analysed using SPSS.

5 The Realisation of 21st Century Skills in the Classroom

Drawing on data from the evaluation of each cycle, we now consider the effectiveness of this approach for embedding 21st century skills in the classroom, focussing in particular on critical thinking and problem solving, communication and collaboration, creativity, and digital literacy.

5.1 Critical thinking and problem solving

As exemplified above, many of the learning activities provide opportunities for developing critical thinking and problem solving. In addition, reflection has been central to all four cycles, originally developed as a learning activity called ‘Team newsflash’ and subsequently labelled Reflection/Reflect.

TeamUp, is a prototype tool developed initially in response to the design challenges noted by teachers in participatory design workshops. Groups of students can record 60-second audio ‘newsflashes’ responding to specific questions: what they have done, what they will do next and any problems encountered. Recordings are available for other groups and teachers to access at any time. In cycle 4, a second prototype tool, ReFlex, was offered which enables students to build up a series of reflections about their learning activities which are then displayed on a timeline. Alternative widely available audio/visual reflection tools, blogs, mindmapping tools and note-taking tools (online sticky notes) were also suggested to teachers.

The evaluation results suggest that the use of technology to support critical thinking, problem solving and reflection had a positive impact. In cycle 4, 73 % of participating teachers (n=326) agreed that their implementation of iTEC improved their students critical thinking skills (linked to improvements in self-reflection skills) whilst 80 % agreed that their students' problem solving skills had improved. Facilitating student reflection, supported by tools such as TeamUp, ReFlex and blogs, was seen to be particularly innovative among case study teachers. The perceived benefits included enabling teachers to monitor progress, developing students' metacognition and self-evaluation skills, and supporting peer learning. Case study data supported these findings in various ways although teachers did not mention critical thinking or problem solving specifically very often.

> [The learning story has] fostered the development of metacognitive processes, critical thinking and autonomy. (head teacher, Italy, cycle 3)

> [Being able to record reflections using technology] forces students to think about their work, become aware of the work we have been able to do, and skills they have developed. (teacher, Spain, cycle 3)

Students also perceived benefits to be gained from using technology for reflection. In cycle 4, students from Israel noted that whilst reflection could be achieved using pen and paper, they believed that the act of recording (and listening back to) their reflection made them think about their statement more and encouraged them to make their points more clearly than they might do in a written document.

Of course, supporting student reflection was not without its challenges in relation to time pressures, student attitudes, skills and technical problems associated with the use of prototype tools. In cycle 2 for example, it was noted that students were not always comfortable recording reflections.

5.2 Communication and collaboration

Communication and collaboration are required for many of the learning activities developed in the iTEC project from working in teams to conducting participatory design workshops. Suggestions for appropriate technologies include TeamUp (described above), blogs, Facebook and wikis. For example, in cycle 1 the 'Working with outside experts' learning activity suggests the use of instant messaging or Skype to communicate synchronously with experts. In cycle 2 the 'Ad-hoc collaboration' learning activity suggests the use of Twitter or Facebook to communicate their activities to and collaborate with students at other schools.

Teachers from the first three cycles (n=826) felt that students expressed their ideas in new ways (87 %), and communicated in new ways with each other (81 %), the teacher (78 %) and the wider community (59 %). Most teachers agreed (n=826) that the iTEC process increased opportunities for collaboration (93 %), enabled students to develop new skills in collaboration (91 %) and enabled students to use ICT to support collaboration (87 %). In cycle 4, teachers (n=326) agreed that students improved their communication skills (86 %) and their collaboration skills (87 %).

In cycles 1–3, 71 % of teachers reported using communication tools such as email, conferencing or instant messaging when implementing the learning activities. Indeed, communication tools were in the top 3 tools used in each cycle, perhaps reflecting the ready availability of such tools in European classrooms. Some teachers commented that the way in which they interacted with students changed through the use of technology, for example, they communicated and commented on students' work via blogs instead of providing written comments in exercise books.

Students also used digital tools to communicate and collaborate with other students, both in their class and in other schools. The iTEC Facebook group page was set up to enable teachers and students to find other iTEC participants to collaborate with and to share what they had been doing. During Cycle 2 many students posted links to the artefacts that they had created and received 'likes' and comments from others. Further examples of the use of digital tools for communication and collaboration include the following:

> An unexpected outcome of the blogs is that students from other classes and schools left comments and suggestions for the class' students. All of the posts were encouraging and constructive and students appreciated this feedback. (case study report, France, cycle 2)

> Students have done a lot of work at home, working and communicating together and with me by using Facebook, and documenting the process by using a blog. They were not used to this kind of activity ... and me too. I had already used blogs, but not in the normal curricular activities (only in special extra-curricular activities), and not in this way: it's a new kind of homework, in which students have to reflect on what they do in the schoolwork. (teacher, Italy, cycle 4)

Students came to appreciate the value and benefits of team working:

> We found iTEC more enjoyable than our other lessons because we worked as a group and everybody in our team always help each other and we learned a lot of things and we shared our experiences with each other. (student, Turkey, cycle 4)

Again, this has not been without challenge with some teachers feeling that some students struggled to write to a standard and in a format suitable for a blog whilst others noted that school policies prevented them from using communication tools such as Twitter. An Italian teacher in cycle 2 also noted that it was a challenge to facilitate primary students collaborating with secondary students, attributed to the lack of ability to respond to complex communications.

5.3 Creativity

As the learning stories for each cycle indicate, iTEC supports creativity in a variety of ways. For example, in cycle 4, the learning stories were 'create a game', 'create an object' and 'tell a story', all clearly creative acts. Learning stories from previous cycles involved students redesigning their school; creating a maths game; and creating a guided walk. During the first three cycles of iTEC, 91 % of teachers (n=826) agreed that the process enabled students to both engage in creative activities and to develop creative their skills. In cycle 4, 89 % of teachers (n=326) agreed that their students' creativity skills had improved.

It was both the types of outputs which students were expected to produce (e.g. videos, products, games) and the tools they were expected to use which were frequently thought to promote creative skills. Using a range of digital tools to support the entire design and creation process, rather than simply to undertake research and present findings, was new for a number of teachers and students:

> The devices are being used a lot. What we see now is that students use them more to create things rather than use them to look up or produce texts. (head teacher, Belgium, cycle 3)

Furthermore, some students felt that the digital tools provided them with ways to express themselves, which were not available with more traditional tools:

> It also helps us to be more creative because sometimes a pencil and a piece of paper aren't enough to show what is in my mind in real terms. (student, Turkey, cycle 3)

5.4 Digital literacy

While implementing their iTEC learning stories, teachers were encouraged to experiment with digital tools which they, and their students, had not used before. During cycles 1–3, teachers reported the use of an average of 8.2 (SD=2.7) different types of ICT (for example collaboration tools, communication tools, media recording tools), with 60 % (n=826) agreeing that they used digital tools that they had not used before, thus exposing students to a wider range of digital tools. It was clear that, prior to iTEC, some students had very limited experience of the use of new technologies in a learning context, for example:

> We have made a PowerPoint presentation once or twice, but we don't use computers in our lessons very often. (student, Estonia, cycle 3)

> The differences are that during this project all the students have used a computer not like the rest where only the teacher uses the computer. (student, Spain, cycle 3)

Exposure to a wide range of technologies during iTEC therefore helped them to appreciate how new technologies could be used for learning and the benefits and potential challenges of doing so, along with the technical skills required to use these technologies effectively. In cycle 4, 87 % of teachers (n=326) agreed that their students' digital literacy skills had improved. Students themselves were also clearly aware that their digital literacy skills were improving:

> I knew some basic things on the computer, but since the project I know a lot more about file sharing and GoogleDocs (student, Israel, cycle3)

> I never worked with Google SketchUp before and because of this project I know how to use it and I also learned how to develop my own blog. (student, Slovakia, cycle3)

Crucially, iTEC provided an opportunity for students to develop digital literacy skills in a practical context. This allowed them to learn about the full range of social, inter-personal and ethical issues around the use of digital tools, not simply the technical skills required:

> … before the project, in ICT class, they've used blogs but not with a real use, the project has allowed them to really see what means to publish information and work through a blog, they could see how many people would visit them and really understand that their information was public. (ICT coordinator, Spain, cycle3)

6 Discussion and Conclusion

As Rutowski et al (2011) acknowledge, "significant challenge remains in integrating ICT into the pedagogical practices aimed at developing 21st-century skills". The importance of learning support systems for the transformation of pedagogies and to facilitate the development of 21st-century skills is well-recognised (Cuban et al., 2001; Pelgrum, Law, 2003), although as Voogt and Pareja Roblin (2012) point out, most frameworks pay little attention to these practical considerations. Results from iTEC indicate that support is, indeed required in the four areas identified by P21 as: 21st Century Learning Support Systems, namely, 'Standards and Assessment'; 'Curriculum and Instruction'; 'Professional Development' and 'Learning Environments'. These are considered in turn below.

6.1 Standards and Assessment

In common with other research on ICT in education, rigid summative assessment practices and curricula were identified in iTEC as significant barriers to the types of pedagogies, which support the development of 21st century skills. As the majority of assessment frameworks are subject- and knowledge-based, they do not cover 21st century skills (Dede, 2010) although many countries claim that their assessment policies do address this (Ananiadou, Claro, 2009). This was a challenging area for iTEC to address especially as the project involved a large number of countries each with their own standards and assessment procedures. The evidence from iTEC supports that of others (e.g. Voogt et al., 2013) suggesting assessment frameworks need to be revised to meet the needs of 21st century teaching and learning and ensure skills, or competencies, as well as knowledge are assessed. Teachers were overwhelmingly positive about the impact of iTEC on students' development of 21st century skills, and frequently assessed these skills using formative, peer and self-assessment methods. As one teacher commented, "it allowed me to assess some things which are not always easy to measure in a normal class. For example, autonomy, creativity, critical thinking …" (teacher, Portugal, cycle 3). As this comment indicates, these were not reflected in formal assessments.

6.2 Curriculum and instruction

Curricula need restructuring to accommodate 21st century skills, which are often dis-connected from subject areas (Voogt et al., 2013). However, evidence from iTEC suggests that new pedagogical strategies such as collaboration and creativity can be embedded across the curriculum. The development of generic learning activities and exemplar learning stories supported this. Teachers were able to use the resources as sources of inspiration in order to redesign their classroom pedagogies. Through the iTEC process teachers were involved in the collective design of new learning activities underpinned by 21st century skills and a wide range of ICTs. iTEC provided opportunities to incorporate 21st century skills in subjects where this was not usual practice, in particular, the integration of digital literacy in subjects which traditionally made little use of technology. As a school ICT coordinator commented, "It has been a nice change. The pilot here has been done in maths and usually ICT is not used in maths classes." (ICT Co-ordinator, Spain, cycle 3). Changing the curriculum and instruction methods is far from easy, however. The iTEC process offered teachers ways to adapt and develop their teaching, but also demonstrated that this needs to be undertaken as a whole school process.

6.3 Professional development

ICT technical support and ICT pedagogical support were identified as important enablers for iTEC teachers to support them in the delivery of 21st century skills. In Cycle 1 access to technical and pedagogical support were noted to be essential for main-streaming. Alongside support from iTEC national co-ordinators, informal professional development opportunities such as the support of other teachers (either face-to-face or through online communities) was found to be important to teachers.

Whilst the iTEC process ensures that 21st century skills are embedded in learning activities, it does not address the specific skills required by teachers and students. Key aspects of the delivery of 21st century skills, such as students working in teams, the use of a range of digital tools and supporting student reflection were all areas where teachers felt they required additional support. For example, in Cycle 3, basic technical problems which could have been resolved with adequate technical support were noted in 31 of the 47 case studies. There is a need for additional resources, training and support to help teachers and students to adapt to these new pedagogical approaches, as Dede (2010) indicates. Currently such support is lacking or optional rather than mandatory (Ananiadou, Claro, 2009).

6.4 Learning environments

The development of 21st century skills also requires changes to traditional learning environments, in particular the provision of new technologies and infrastructure. Among schools in the iTEC project, ICT infrastructure, including the provision of reliable and sufficient access to the internet, requires further development in many countries. For example, in Cycle 2, insufficient access to ICT was the second most commonly identified barrier. Changes such as the introduction of Bring Your Own Device policies whereby students are encouraged to bring their own mobile phones and other electronic devices to use in the classroom represent one way in which this problem can be overcome. This development can facilitate not only the development of digital literacy skills, but also collaboration and communication skills as students can work together in new ways, both in the classroom and beyond.

This notion of extending the traditional boundaries of the classroom is critical to the development of 21st century skills. This was evident in iTEC as students communicated and collaborated with peers and experts outside the school, often using digital tools to do so, and devised creative solutions to ‘real life’ problems. Indeed, Voogt and colleagues (2013) argue that the development of 21st century skills can take place outside formal education and could prove useful if teachers are able to capitalise on this. The traditional boundaries should not be extended but made permeable.

The increasing use of social media is another way in which the learning environments are changing to facilitate the development of 21st century skills. As reported above, this can impact on digital literacy, collaboration and communication skills, creativity and critical thinking as students are presented with new ways to work together, create new types of learning output.

7 Concluding Remarks

The approach developed through iTEC, enabling teachers to become learning designers and embed 21st century skills and ICT in their pedagogical practices, has proved to be flexible and adaptable across a range of curriculum areas. As a result teachers and students perceive that there has been a positive impact on students' 21st century skills. However, the process does not currently support teachers who have limited experience of 21st century skills themselves and thus there is a need to provide further support on the specific skills to ensure that teachers do not spend their time solving known problems and that the benefits are maximised.

References

American Library Association (ALA) Digital Literacy Taskforce. (2011). *What is Digital Literacy?* Retrieved from http://connect.ala.org/files/94226/what%20is%20digilit%20%282%29.pdf

Ananiadou, K., Claro, M. (2009). *21st Century Skills and Competences for New Millennium Learners in OECD Countries*, OECD Education Working Papers, No. 41. Paris: OECD Publishing.

Beetham, H., McGill, L., Littlejohn, A. (2009). *Thriving in the 21st century: Learning Literacies for the Digital Age (LLiDA project)*. Retrieved from http://www.jisc.ac.uk/media/documents/projects/llidareportjune2009.pdf

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., Rumble, M. (2012). Defining Twenty First Learning Skills. In Griffin, P., Mc Gaw, B., Care, E. (Eds.) *Assessment and Teaching of 21st Century Skills* (pp. 17–66). Dordrecht: Springer.

Cranmer, S., Ulicsak, M. (2011). D2.1 *Summary report of scenario development process*. Retrieved from http://itec.eun.org/c/document_library/get_file?p_l_id=10307&folderId=37321&name=DLFE-1624.pdf

Crook, C., Fisher, T., Harrop, H., Stokes, E. (2010). *New Modes of Technology-enhanced Learning: Developing successful practice*. Coventry: Becta. Retrieved from http://dera.ioe.ac.uk/1535/

Cuban, L., Kirkpatrick, H., Peck, C. (2001). High access and low use technologies in high school classrooms: Explaining an apparent paradox. *American Educational Research Journal*, 38(4), pp. 813–834.

Dede, C. (2010). Comparing Frameworks for 21st Century skills. In Bellanca, J., Brandt, R. (Eds.) *21st Century skills: Rethinking how students learn* (pp. 51–76). Bloomington, IN: Solution Tree Press.

European Commission (EC). (2013). *Survey of schools: ICT in education*. Retrieved from http://ec.europa.eu/digital-agenda/sites/digital-agenda/files/KK-31-13-401-EN-N.pdf

International Society for Technology in Education (ISTE). (2007). *ISTE Standard for Students: Advancing digital age learning*. Retrieved from http://www.iste.org/docs/pdfs/nets-s-standards.pdf?sfvrsn=2

Kennedy, M., Fisher, M. B., Ennis, R. H. (1991). Critical thinking: Literature review and needed research. In L. Idol, B.F. Jones (Eds.) *Educational values and cognitive instruction: Implications for reform*, pp. 11–40. New York: Routledge.

Keune, A., Toikkanen, T., Purma, J., Leinonen, T. (2011). *D3.1 Report on design prototypes and design challenges for education*. Retrieved from http://itec.eun.org/c/document_library/get_file?p_l_id=10307&folderId=37321&name=DLFE-1641.pdf

Kozma, R. B. (Ed.) (2003). *Technology, Innovation and Educational Change: A global perspective*. Eugene, OR: International Society for Technology in Education.

Law, N. (2009). Mathematics and science teachers' pedagogical orientations and their use of ICT in teaching. *Education and Information Technologies*, 14, pp. 309–323.

Loveless, A. (2002). *A Literature Review in Creativity, New Technologies and Learning: A Report for Futurelab*. Bristol: Futurelab.

Luckin, R., Blight, B., Manches, A., Ainsworth, S., Crook, C., Noss, R. (2011). *Decoding Learning: The proof, promise and potential of digital learning*. London: Nesta.

Mathews, J. (2009, January 5). The Latest Doomed Pedagogical Fad: 21st-Century Skills. Washington Post, Retrieved from http://www.washingtonpost.com/wp-dyn/content/article/2009/01/04/AR2009010401532.html

Mishra, P., Kereluik, K. (2011). What 21st Century Learning? A review and a synthesis. In M. Koehler, P. Mishra (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2011* (pp. 3301–3312). Chesapeake, VA: AACE.

National Research Council (2012). *Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century*. Committee on Defining Deeper Learning and 21st Century Skills, J. W. Pellegrino and M. L. Hilton, Editors. Board on Testing and Assessment and Board on Science Education, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.

Partnership for 21st century skills (P21). (2009). *Framework for 21st century learning*. Retrieved from http://www.p21.org/storage/documents/1.__p21_framework_2-pager.pdf

Pelgrum, W. J., Law, N. (2003). *ICT in education around the world: Trends, problems and prospects*. Paris: UNESCO International Institute for Educational Planning. Retrieved from http://unesdoc.unesco.org/images/0013/001362/136281e.pdf

Quinton, S., Smallbone, T. (2010). Feeding forward: using feedback to promote student reflection and learning – a teaching model, *Innovations in Education and Teaching International*, 47(1), pp. 125–135.

Rutkowski, D., Rutkowski, L., Sparks, J. (2011). Information and communications technologies support for 21st-century teaching: an international analysis. *Journal of School Leadership*. 21(2), pp. 190–215.

Shear, L., Novais, G., Moorthy, S., Langworthy, M. (2010). Innovative teaching and learning research: Executive summary of pilot year findings. Microsoft Partners in Learning. Retrieved from http://itlresearch.com/images/stories/reports/ITL%20Research%20Pilot%20Year%20Executive%20Summary%20Oct%202010%20F.pdf

Voogt, J., Pareja Roblin, N. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies, *Journal of Curriculum Studies*, 44:3, pp. 299–321.

Voogt, J., Erstad, O., Dede, C., Mishra, P. (2013). Challenges to learning and schooling in the digital networked world of the 21st century, *Journal of Computer Assisted Learning*, 29, pp. 403–413.

Acknowledgements

Thanks to all the teachers and students involved in the iTEC evaluation and to the national co-ordinators and their colleagues involved in data collection. Thanks also to Maureen Haldane, iTEC work package 5 lead for cycles 1–3 at Manchester Metropolitan University. iTEC is coordinated by European Schoolnet (EUN) and co-funded by the European Commission's 7th Framework Programme.

Biographies

Cathy Lewin is a Professorial Research Fellow and Co-Director of the Technology, Innovation and Play for Learning research group. Her research interests concern young people and ICT, in relation to both formal and informal learning. She has conducted research for Becta, DCSF, ESRC and European Schoolnet including ImpaCT2, ICT TestBed, and the Primary Schools Whiteboard Expansion Project. She is currently leading the evaluation work package in the FP7-funded iTEC project.

Sarah McNicol is a Research Associate at the Education and Social Research Institute (ESRI) at Manchester Metropolitan University (MMU) in the UK. She is currently working on the iTEC project and has a background of research into information and digital literacies.

Expert Rating of Competence Levels in Upper Secondary Computer Science Education

Johannes Magenheim[1], Wolfgang Nelles[1], Jonas Neugebauer[1], Laura Ohrndorf[2], Niclas Schaper[1], Sigrid Schubert[2]
[1]University of Paderborn, Germany
{jsm, jonas.neugebauer}@uni-paderborn.de,
{wnelles, nschaper}@mail.uni-paderborn.de
[2]University of Siegen, Germany
{laura.ohrndorf, sigrid.schubert}@uni-siegen.de

Abstract: In the project MoKoM, which is funded by the German Research Foundation (DFG) from 2008 to 2012, a test instrument measuring students' competences in computer science was developed. This paper presents the results of an expert rating of the levels of students' competences done for the items of the instrument.
At first we will describe the difficulty-relevant features that were used for the evaluation. These were deduced from computer science, psychological and didactical findings and resources. Potentials and desiderata of this research method are discussed further on. Finally we will present our conclusions on the results and give an outlook on further steps.

Keywords: Competence Modelling, Competence Measurement, Informatics System Application, Informatics System Comprehension, Informatics Modelling, Secondary Education

1 Introduction

As a result of the on-going discussion about educational standards, competence models were developed for many subjects. They structure the particular learning field into different dimensions and sub-dimensions. In the project MoKoM a competence model with main focus on system comprehension and object-oriented modeling was developed.

This was done in several sub steps, which are shortly described in the following:

1. A theoretically derived competence model was created through the analysis of curricula and syllabi.
2. This model was refined with an empirical approach in terms of expert interviews, which were transcribed and analyzed.
3. On base of the empirically derived competence model a test instrument was created which was applied in a study with more than 500 students.
4. The evaluation results will be used to develop a competence level model that includes differentiated proficiency levels.

As a current research step, an expert rating for each item of our test instrument was done. The main research objectives for this are as follows:

1. To identify, describe, and examine empirically difficulty relevant features of the test items of a competence test of informatics competences
2. To develop a basis for the derivation of a competence level model

2 Difficulty-Relevant Features and Feature Levels

To identify and describe difficulty relevant features of the competency test we first defined difficulty relevant features of the competency test items. We derived those features from the literature concerning difficulty relevant features of competency tests in general (e.g. Schaper et al., 2008). Furthermore we analysed the items concerning informatics specific difficulty facets and tried to define and grade them analogue to the more general features. On this basis altogether thirteen features were identified and defined. In this section we describe for each of the thirteen difficulty-relevant features their feature levels in computer science education (CSE).

2.1 Addressed knowledge categories

In a first step we analysed the structure of the knowledge dimension of the revised taxonomy for learning, teaching, and assessing from Anderson and Krathwohl (Anderson, Krathwohl, 2001) as a possible difficulty relevant feature of the test items. We assumed that the knowledge categories, A. Factual

Knowledge, B. Conceptual Knowledge, C. Procedural Knowledge, D. Meta-cognitive Knowledge and its usage for solving the test tasks, would differentiate between different levels of difficulty concerning our test items. So we derived the first difficulty-relevant feature with the following four feature levels:

- WI1: The successful solution of the task requires bare factual knowledge, and conceptual knowledge about the basic elements of computer science.
- WI2: The successful solution of the task requires basic and elaborated conceptual knowledge. The students recognize functional connections among basic elements of computer science within a bigger structure and task formulation.
- WI3: The successful solution of the task requires procedural knowledge. The students understand methods, rules and concepts to actively apply skills of computer science.
- WI4: The successful solution of the task requires meta-cognitive knowledge. The students know which cognitive requirements are needed for the available computer science task and how they can obtain and use the required contents to solve the task.

2.2 Cognitive process dimensions

In a second step we analysed the structure of the cognitive process dimensions of the revised taxonomy for learning, teaching, and assessing by Anderson and Krathwohl (Anderson, Krathwohl, 2001). We could assume that these addressed process categories, 1. Remember, 2. Understand, 3. Apply, 4. Analyze, 5. Evaluate and 6. Create, would also differentiate the levels of our test items. So we got the second difficulty-relevant feature with the following six feature levels:

- KP1: The successful solution of the task requires a memory performance. The students recall relevant knowledge contents from their memory.
- KP2: The successful solution of the task requires a comprehension perfor-mance. The students understand terms, concepts, and procedures of computer science and can explain, present and give examples for them.

- KP3: The successful solution of the task requires an application performance. The students are able to implement known contents, concepts and procedures within a familiar as well as an unfamiliar context.
- KP4: The successful solution of the task requires an analysis. The students are able to differentiate between relevant and irrelevant contents, concepts and procedures. They choose the suitable procedures from a pool of available procedures.
- KP5: The successful solution of the task requires a rating (evaluation). The students are able to evaluate the suitability of concepts and procedures of computer science for the solution of the task.
- KP6: The successful solution of the task requires a creation. The students are able to develop a new computer science product by using concepts and procedures of computer science.

2.3 Cognitive combination and differentiation capacities

In a third step we applied findings of developmental psychology, e.g. of Piaget (Piaget, 1983). We could assume that these addressed combinations, *Reproduction, Application, Networked application*, would differentiate between different levels of difficulty concerning our test items. So we derived the third difficulty-relevant feature with the following three feature levels:

- KV1: Reproduction of computer science knowledge and application of single, elemental terms, concepts and procedures of computer science in close contexts (no cognitive combination capacities required).
- KV2: Application of single terms, concepts and procedures of computer science in bigger contexts, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made.
- KV3: Networked Application of terms, concepts and procedures of computer science in different, especially bigger scenarios, whereas an argumentative and/or intellectual consideration between competitive terms, concepts and procedures (approaches) for example has to be made (multiple challenging cognitive combination capacities required).

2.4 Cognitive stress

In a fourth step we applied findings of cognitive psychology, e.g. of Jerome Bruner (Bruner, 1960). We assumed that these abstraction levels would differentiate the levels of difficulty concerning our test items. So we derived the fourth difficulty-relevant feature with the following three feature levels:

- KB1: For the successful solution of the task little, consecutive processing steps and no transfer performances are required: The degree of abstraction is very low.
- KB2: For the successful solution of the task many, consecutive processing steps and average transfer performances are required: The degree of abstraction is medium.
- KB3: For the successful solution of the task very many, consecutive processing steps and huge transfer performances are required: The degree of abstraction is very low.

2.5 Scope of tasks (necessary materials, reading effort and understanding)

In a fifth step we applied findings of educational psychology, e.g. of Benjamin Bloom (Bloom, Engelhart, Furst, Hill, Krathwohl, 1956). In this case we assumed that the addressed scope levels would differentiate between the levels of our test items.

So we derived the fifth difficulty-relevant feature with the following three feature levels:

- UM1: The task is formulated very short. No additional materials are required.
- UM2: The task is formulated extensive, only less material is required and the reading effort is kept within limits.
- UM3: The task is formulated very extensive. A high reading effort (quantitative and/or qualitative) and extensive materials (e.g. in the form of descriptions, APIs, overviews) are required for the solution of the task.

2.6 Inner vs. outer computational task formulation

In a sixth step we applied findings of didactics of informatics, e.g. of Deborah Seehorn (The CSTA Standards Task Force, 2011). We assumed that these addressed relation between inner and outer computational task formulation would differentiate the levels of competence concerning our test items. So we derived the sixth difficulty-relevant feature with the following two feature levels:

- IA1: For the successful processing of the task, *no* translation in an inner-computational format has to take place. The task is already present in a determined computational format.
- IA2: For the successful processing of the task, a translation in an inner-computational format has to take place. The task is already present in a determined computational format.

Aspects of demands of computer science

For aspects concerning special demands in computer science tasks we utilized dimension K4 of our competency model as a feature. This dimension covers the complexity of systems (Linck et al., 2013).

2.7 Number of components

The amount of components is a feature for the complexity of systems. This does apply to the understanding as well as the development of these. It is important to understand the effects and modes of operations in existing systems. The more components interact together, the more interactions have to be considered. When transferring this to the development of systems it is extended by the decision which components are needed and which tasks they fulfil.

2.8 Level of connectedness

As it might appear at first, the level of connectedness is not restricted to a concrete connection between systems (i.e. a network connection). It also refers to the connection of information used like the handling of data organized in a database. The more connectedness is required to fulfil the task, the more complex it is.

2.9 Stand-alone vs. distributed system

When dealing with distributed systems knowledge of the interaction of components and the connectedness of these is needed. This introduces a further level of abstraction since this involves different systems, which more or less multiplies the elements or parts that have to be considered.

2.10 Level of Human-Computer-Interaction (HCI)

The level of HCI needed is not necessarily given in the definition of the task, but can also be a part of the solution or the path to the solution. In this case learners should be able to decide which level is appropriate to fulfil the requirements. This also includes a decision based on the actual target group, e.g. it is depending on the user of software if it should be implemented as a simple command-line tool or a GUI.

2.11 Combinatorial complexity (mathematical)

The combinatorial complexity addresses the area of software tests with the creation of test cases as the main purpose. This is relevant not only for the actual testing process but also for the development of algorithms, where requirements have to be defined first and then verified. This can only be done by the development of suitable software tests.

2.12 Level of the necessary understanding of systems of computer science

This aspect describes the level of in-depth knowledge of computer science. It starts with a basic knowledge level, which can be build up through an everyday experience with computer science systems. It does not require a lengthy learning process. This aspect transitions through an interim level up to the need of fundamental ideas and concepts in computer science education. Furthermore for these tasks an independent evaluation of the system is needed.

2.13 Level of the necessary modelling competence of computer science

Computer science tasks often require modelling skills, which are covered by this difficulty feature. The feature varies from the basic illustration of tasks with a pseudo code to a complex transition with different UML-diagrams.

3 Research Methodology of Expert Rating

The experts were asked to rate each item of the competence test with reference to the thirteen difficulty features. Therefore a rating scheme and instruction was designed.

Furthermore, to conduct the expert ratings the measurement instrument was split into four parts of roughly equal size. To test the rating process one of these parts was used in a preliminary rating with hessian computer science teachers in the course of a teachers' workshop. The discussions during this test resulted in the addition of a "not relevant" rating level for all features, since the teachers thought some features inapplicable for some of the items. Each of the four instrument parts – including solutions for all items – was presented to two selected experts in the field of didactics of informatics, along with an explanation of each feature and its rating levels. The experts were asked to answer each item on their own, compare the solution with the given sample solution and then rate the item for each of the features. In addition, the experts had to give a subjective rating of the item difficulty on a scale from one to ten.

The resulting two ratings for each item were compared and treated in three ways: 1. Exact matches between the ratings were considered final, 2. Items with big differences for one or more features were transferred to a new rating booklet and 3. Every other rating was discussed within the project group to decide upon a final rating. A "big" difference was considered to be a substantial disagreement in the experts opinion, e.g. one expert rated the feature SG "not relevant" (SG0) for an item while the other saw the need to use high levels of modelling skills (SG3) to solve it. An example for a small difference would be the differentiation between "high levels of modelling activities" and "medium levels of modelling intensity".

After that the new rating booklet was given to two new experts together with the ratings of both previous experts. Then they were asked to go through the same process as the other experts to rate the items. Though they had the two previous ratings for each feature available for orientation, they could rate each item independently from them. The results from this second rating were compared in the same way as explained above. This time all differences in the new ratings were discussed by the members of the project team in order to decide upon a final rating for each item and feature.

The group was composed of seven researchers with background in computer science, computer science education and psychology. Since the group had to thoroughly discuss every feature, the process was done in two sittings. Afterwards each item had been assigned to a distinct rating level for each feature.

4 Results of the Expert Rating

The rating process resulted in a classification of 74 items concerning each of the de-scribed features. The rating levels for each feature were coded as increasing numbers, e.g. coding WI1 as 1 and WI2 as 2. For every feature the "not relevant" rating was coded as 0. This way, we ended up with 13 nominal variables with n+1 categories for a feature with n levels. For almost all features it was reasonable to assume a ranking of the levels in the order they are described above. The assumption is that a higher level correlates with a higher item difficulty. Thus, the variables are considered to have an ordinal scale. Though this presumption does not necessarily have to be true, the order will be reviewed through the analysis of the rating data. This was done using descriptive and explorative methods to determine the relevant features that influence the item difficulty.

Comparing the number of times a feature was rated as "not relevant" for an item implies that the experts hold some features to be less useful in determining the difficulty of an item. Especially the features derived from the fourth competence dimension *K4 Dealing with system Complexity* were mostly considered to be inapplicable by the experts. The number of times each feature was deemed not relevant can be seen in figure 1. Interestingly the two features not derived from K4 with the highest number of "not relevant" ratings were *inner vs. outer computational task formulation* (IA) and the degree of necessary modelling competence (SG).

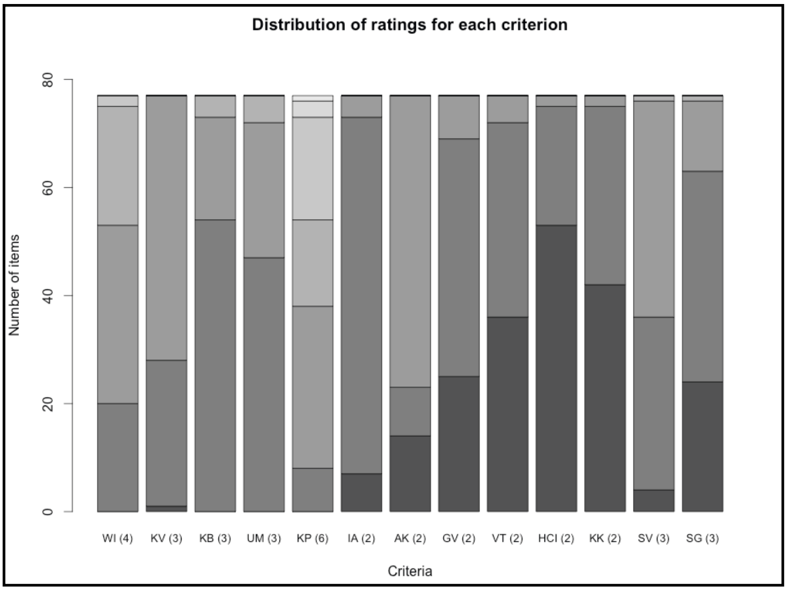


Figure 1: Each feature was rated for 77 items in two to six categories (in brackets) plus the "not relevant" (dark grey) rating.

The ratings for the feature IA suggest that the differentiation between inner and outer computational formats is not that easy in computer science tasks. The experts rated the majority of items as already being in an inner computational format. Judging by this, the feature might not be well suited to differentiate item difficulties. The SG feature was derived from the competence dimension *K3 System Development*, which represents an important part of the competence model. The experts saw no relevance of this feature for 24 items, over 30 % of the instrument. This actually could be expected, since each item was designed with exactly one competence profile in mind. Looking at the items with a relevant SG rating, they include almost all of the items designed for K3. Therefore the rating suggests that the test items were well constructed with regards to their competence profiles. On the other hand though, this raises the question why the feature SV, derived from *K2 System Comprehension*, was considered relevant for all but 4 items, since the items intended to measure these competences were constructed the same way as those for K3. An explanation for this is the need to comprehend system functions on an external and internal level, before being able to design and construct such systems. This is why items that refer to K3 often times also require some form of system comprehension. Figure 1 shows the number of ratings for each category of each feature.

The overall difficulty of the test instrument was subjectively rated by the experts with a mean of 4.2 on a ten-point scale. The distribution of difficulty estimates shows a tendency to the lower ratings, suggesting that the overall item pool might be marginally too easy (see figure 2). Ideally the difficulty ratings would be distributed normally, showing the most items in the medium difficulty range and an equal amount of easy and hard items. Though these ratings are subjective estimates by the experts they substantially correlate with the estimates from the IRT analysis ($r(72) = .553$, $p < .001$) and thus can be seen as an indicator for the validity of the expert ratings.

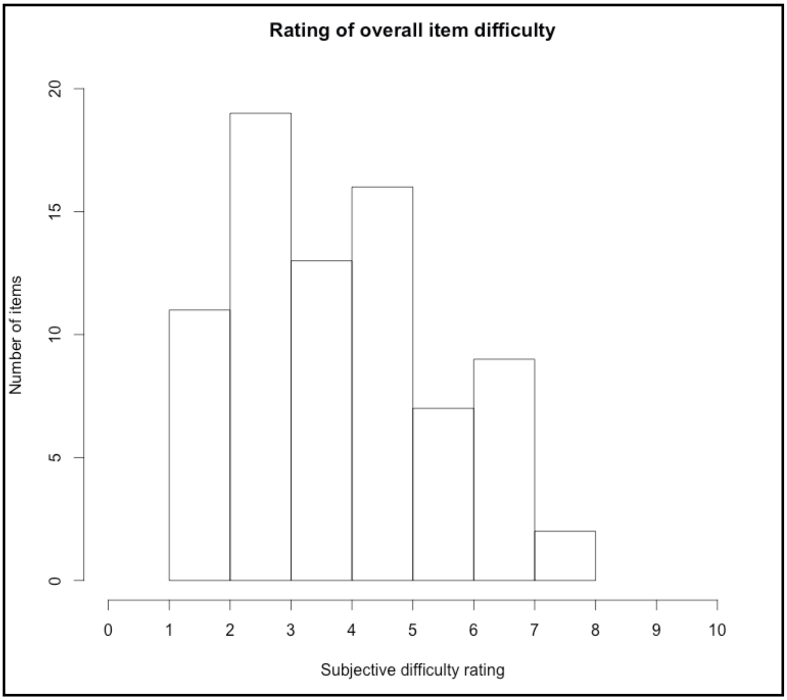


Figure 2: Histogram of subjective difficulty ratings on a ten-point scale.

4.1 Regression analysis

To determine which features have the most influence on the item difficulty, the expert ratings were related to the empirical difficulty estimates that were calculated by means of the Item Response Theory (IRT) (Moosbrugger, 2008; Rost, 2004). The utilization of the IRT allowed for the application of a matrix design, in which not all test subjects have to work on every item (Hartig, Jude, Wagner, 2008). The test instrument was partitioned into six booklets with only parts of the tasks. All together the booklets were distributed to 538 computer science students in upper German high school education. The analysis of the returned data was done with ACER ConQuest, applying a 1PL partial credit model to estimate the item difficulties (Wu, Adams, Wilson, Haldane, 2007). The estimated parameters had a mean of -3.405 and standard deviation of 1.25.

To be able to use regression methods each ordinal variable of *n* levels had to be dummy coded into *n-1* dichotomous variables. Each dummy variable *i* would be 1 if the ordinal variable had the value *I*. The “not relevant” rating was coded as all dummy variables being 0. The dummy coding for the feature IA can be seen in table 1.

Table 1: Dummy coding for IA

IA	IA1	IA2
0	0	0
1	1	0
2	0	1

Since for some features one of the rating levels never was assigned to any item (e.g. "not relevant" for WI) the number of levels for those effectively was reduced by one. Thus, for these features another rating level had to be omitted from the dummy coding. The resulting 32 variables were used as the explanatory variables in a linear regression analysis with the difficulty estimate from the IRT analysis as the dependent variable (Hartig, 2007; Moosbrugger, 2008; Schaper, Ulbricht, Hochholdinger, 2008; Watermann, Klieme, 2002).

To evaluate the regression model the coefficient of determination can be examined. A value of 1.0 indicates that the item difficulty is completely explained by the analysed features, a value of 0.0 means that there is no link between the features and the empirical item parameters (Bortz, Schuster, 2010; Hartig, 2007). The analysed features significantly predict about 71 % of the differences in the item difficulties ($R^2 = 0.717$, $F(32,42) = 3.241$, $p < .001$). Though this is a good result, due to the high number of explanatory variables, this value might be overestimated. The adjusted R^2 takes the number of variables into account and takes a value of $R^2adj = 0.496$. Table 2 shows the regression coefficients, t-values and significance for the regression model.

The significance of the results is low for most of the features. The rating levels with the most significant influence on the item difficulty are AK2, HCI2 and SV3, with the last having the most substantial impact on the difficulty, increasing it by $b = 6.37$ points if the third level of the feature SV was assigned to an item (Hartig, 2007). The number of assignments for all three rating levels was very low (9, 2 and 1 times) respectively, but the features might still be valuable to differentiate item difficulties. The features AK and HCI stem from the competence dimension K4. As mentioned above, those features were considered "not relevant" for large parts of the items. Despite this, they still seem to be relevant for the estimation of item difficulties.

Table 2: Results of the regression analysis

	First regression model			**Second regression model**		
	b	*t*	*p*	*b*	*t*	*p*
Constant	-6,470	-4,746	< .001	-6,574	-5,265	< .001
WI2	-0,358	-0,926	0,3598	-0,274	-0,765	0,4482
WI3	-0,232	-0,416	0,6797	-0,143	-0,283	0,7786
WI4	*-3,732*	*-2,626*	*0,0121*	*-3,792*	*-2,827*	*0,0069*
KV1	1,647	1,406	0,1673	1,892	1,796	0,0791
KV2	1,218	1,074	0,2890	1,390	1,368	0,1780
KB2	0,403	0,857	0,3966	0,314	0,753	0,4554
KB3	-0,196	-0,152	0,8796	-	-	-
UM2	-0,511	-1,385	0,1735	-0,466	-1,450	0,1538
UM3	0,682	0,887	0,3801	0,576	0,944	0,3503
KP2	*1,134*	*2,514*	*0,0160*	1,138	2,658	0,0108
KP3	*1,310*	*2,316*	*0,0256*	*1,433*	*2,733*	*0,0089*
KP4	0,599	1,079	0,2868	0,601	1,209	0,2327
KP5	-0,044	-0,048	0,9621	0,050	0,058	0,9540
KP6	*2,410*	*2,052*	*0,0466*	2,360	2,127	0,0388
IA1	0,077	0,16	0,8737	-	-	-
IA2	0,633	0,745	0,4607	-	-	-
AK1	0,130	0,301	0,7651	-	-	-
AK2	***2,899***	***3,59***	***0,0009***	***2,927***	***4,088***	***0,0002***
GV1	-0,010	-0,023	0,9816	-	-	-
GV2	*-1,747*	*-2,162*	*0,0365*	-1,853	-2,648	0,0111
VT1	0,740	1,684	0,0997	0,705	1,917	0,0615
VT2	*-1,494*	*-2,615*	*0,0124*	-1,567	-3,024	0,0041
HCI1	***-1,916***	***-4,229***	***0,0001***	***-1,746***	***-4,360***	***0,0001***
HCI2	***-4,797***	***-3,969***	***0,0003***	***-4,411***	***-4,076***	***0,0002***
KK1	0,499	1,48	0,1466	0,461	1,717	0,0927
KK2	0,877	1,153	0,2554	0,889	1,247	0,2187
SV1	0,220	0,377	0,7083	0,187	0,338	0,7369
SV2	0,839	1,392	0,1714	0,863	1,584	0,1200
SV3	***6,376***	***3,75***	***0,0005***	***6,170***	***3,904***	***0,0003***
SG1	0,542	1,67	0,1026	0,515	1,715	0,0931
SG2	0,820	1,345	0,1859	0,888	1,811	0,0766
SG3	2,020	1,03	0,3089	2,278	1,330	0,1901

As a consequence from the results, the regression model was modified to get a minimal model that adequately could predict item difficulties. Features with low significance and low impact on the item difficulty can be dropped from the model. Furthermore, significant differences in the rating levels can be assessed by a one-way analysis of variance.

For this reason, the feature IA was removed. Both rating levels have low significance and the regressions coefficient of IA1 is fairly (IA1: $b = .077$, $r(42) = .16$, $p = 0.87$; IA2: $b = .633$, $r(42) = .75$, $p = .46$). Additionally, no significant differences between the three rating levels could be assessed. The rating levels of the feature KB showed a significant difference between KB1 and KB2, but not between KB2 and KB3. For this reason, the upper two rating levels were combined. Though the feature AK seems to differentiate well on the level AK2, the difference between AK0 and AK1 is not significant, which lead to the combination of those two rating levels. The distinction between “no”, “few” and “many” system components can be reduced to “few or none” and “many” components. The same is true for the feature GV, now only differentiating between “low or none degree of connectedness” and “large degree of connectedness”. Figure 3 shows the boxplots for the mentioned features.

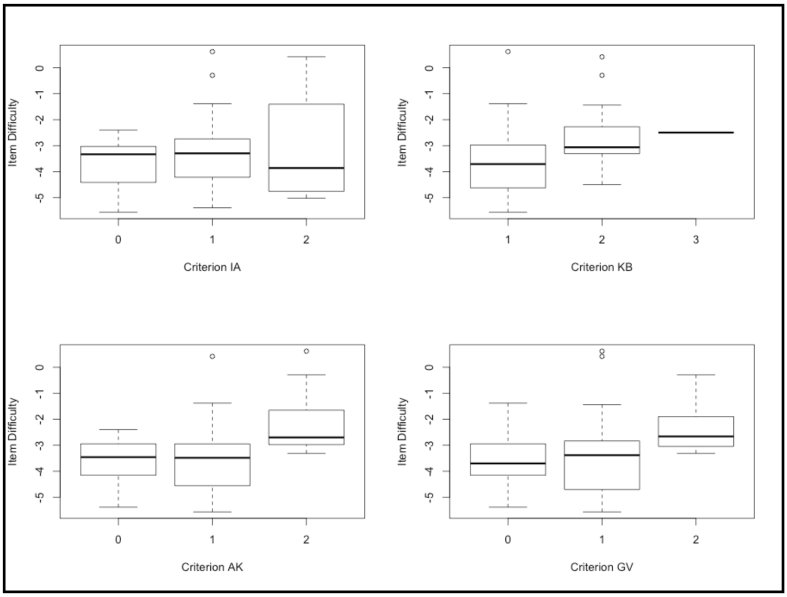


Figure 3: Boxplots for the features IA, KB, AK and GV

The newly evaluated regression model is still significant with a slightly lower coefficient of determination ($R^2 = 0.707$, $F(27,46) = 4.114$, $p < .001$) and an increased $R^2adj = .535$. To get to a minimal model with only significant rating levels, the insignificant variables can be stepwise removed and the model revaluated. By doing so, the features AK, VT, HCI, SV and SG remained in a significant model (R^2 = 0.48, $F(8, 65) = 7.504$, $p < .001$). These features can explain about 48 % of the item difficulties.

5 Conclusion

The analysis of the results of the expert ratings shows that most of the variance in item difficulties can be explained by the selected features. By reducing the features to the most significant ones, item difficulties can still be predicted an amount of 48 % variance determination. To allow for a feature-oriented interpretation of the IRT results, the next step will be to use the significant features to calculate the expected difficulty of items, rated with certain combinations of the features. These combinations will define appropriate thresholds between the proficiency levels (Beaton, Allen, 1992; Hartig, 2007; Schaper et al., 2008). The selection of suitable combinations of the features has to be based on theoretical and empirical sound decisions. For example the proficiency levels should be appropriately spaced and include items that define them, by satisfying the selected features. Moreover, the features should be useful to give meaningful explanations of the expected abilities in each proficiency level (Hartig, 2007; Watermann, Klieme, 2002).

If the a-priori rating of items yields no appropriate features to reasonably explain the difficulty of the items, it might be necessary to utilize post-hoc analysis methods used in other large-scale assessments like TIMSS III and PISA 2000 (Helmke, Hosenfeld, 2004; Schaper et al., 2008). With this method, distinctive items, characterized by certain thresholds in the item difficulties, are analysed for features that can be used to describe the proficiency levels. This approach has the disadvantage though, that the description for each level is dependent on the items used in the competence assessment.

References

Anderson, L. W., Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Addison Wesley Longman.

Beaton, A. E., Allen, N. L. (1992). Interpreting Scales Through Scale Anchoring. *Journal of Educational and Behavioral Statistics, 17*(2), pp. 191–204. doi:10.3102/10769986017002191

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., Krathwohl, D. R. (1956). *Taxonomy of educational objectives: Handbook I: Cognitive domain*. (Bloom, B. S., Ed.). New York: David McKay.

Bortz, J., Schuster, C. (2010). *Statistik für Human-und Sozialwissenschaftler. Lehrbuch mit Online-Materialien* (7 ed.). Berlin, Heidelberg, New York: Springer.

Bruner, J. S. (1960). *The Process of Education*. Cambridge, MA: Harvard University Press.

Hartig, J. (2007). Skalierung und Definition von Kompetenzniveaus. In Klieme, E., Beck, B. (Eds.), *Sprachliche Kompetenzen. Konzepte und Messung. DESI-Studie (Deutsch Englisch Schülerleistungen International)* (pp. 83–99). Weinheim: Beltz.

Hartig, J., Jude, N., Wagner, W. (2008). Methodische Grundlagen der Messung und Erklärung sprachlicher Kompetenzen. In DESI-Konsortium (Ed.), *Unterricht und Kompetenzerwerb in Deutsch und Englisch. Ergebnisse der DESI-Studie* (pp. 34–54). Weinheim, Basel: Beltz.

Helmke, A., Hosenfeld, I. (2004). Vergleichsarbeiten – Standards – Kompetenzstufen: Begriffliche Klärung und Perspektiven. In Wosnitza, M., Frey, A., Jäger, R. S. (Eds.), *Lernprozess, Lernumgebung und Lerndiagnostik Wissenschaftliche Beiträge zum Lernen im 21. Jahrhundert* (pp. 56–75). Landau: Verlag Empirische Pädagogik.

Linck, B. et al. Competence model for informatics modelling and system comprehension. In *Proceedings of the 4th global engineering education conference, IEEE EDUCON 2013*. pp. 85–93, Berlin (2013).

Moosbrugger, H. (2008). Item-Response-Theorie (IRT). In Moosbrugger, H., Kelava, A. (Eds.), *Testtheorie und Fragebogenkonstruktion* (pp. 215–259). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-71635-8_10

Piaget, J. (1983). Piaget's Theory. In P. Mussen (Ed.), *Handbook of Child Psychology* (Vol. 1). New York: Wiley.

Rost, J. (2004). *Lehrbuch Testtheorie–Testkonstruktion* (2nd ed.). Bern, Göttingen, Toronto, Seattle: Huber.

Schaper, N., Ulbricht, T., Hochholdinger, S. (2008). Zusammenhang von Anforderungsmerkmalen und Schwierigkeitsparametern der MT21-Items. In Blömeke, S., Kaiser, G., Lehmann, R. (Eds.), *Professionelle Kompetenz angehender Lehrerinnen* *und Lehrer: Wissen, Überzeugungen und Lerngelegenheiten deutscher Mathematikstudierender und -referendare – Erste Ergebnisse zur Wirksamkeit der Lehrerausbildung* (pp. 453–480). Waxmann Verlag.

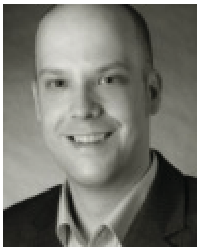
The CSTA Standards Task Force. (2011). CSTA K–12 *ComputerScience Standards*. (Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., et al., Eds.) (pp. 1–73). Retrieved from http://csta.acm.org/Curriculum/sub/K12Standards.html

Watermann, R., Klieme, E. (2002). Reporting Results of Large-Scale Assessment in Psychologically and Educationally Meaningful Terms. *European Journal of Psychological Assessment, 18*(3), pp. 190–203. doi:10.1027//1015-5759.18.3.190

Wu, M. L., Adams, R. J., Wilson, M. R., Haldane, S. A. (2007). *ACER ConQuest Version 2.0: Generalised item response modelling software*. Melbourne, Australia: ACER Press.

Biographies

Johannes Magenheim is a Professor for Computer Science Education at the Institute of Computer Science at the University of Paderborn. His primary areas of research and teaching are CSE, E-Learning, CSCL. He is also member of different working groups of the German Informatics Society (GI) and member of IFIP WGs 3.1 and 3.3.



Wolfgang Nelles is psychologist (Diplom-Psychologe) and his research interests concern Work- and Organizational Psychology. Since 2008 he is employed as a research assistant at the University of Paderborn.

Jonas Neugebauer received his first state exam for secondary education in computer science and mathematics from the University of Paderborn, Germany (2011). Since April 2011 he is employed as a research assistant at the Computer Science Education Group at the University of Paderborn.

Laura Ohrndorf received her diploma in computer science (2011) from the University Duisburg-Essen, Germany. Since July 2011 she is a research assistant at the University of Siegen. Her main research interests concern misconceptions in computer science.

Niclas Schaper is a Professor for Work and Organisational Psychology at the University of Paderborn. His primary areas of research are job analysis, occupational training, approaches of competency modeling, methods of competency measurement, E-Learning, and teaching in higher education. He is a member of the German Society of Psychology and vice president of the German Society of Teaching in Higher Education.

Since 1979 **Sigrid Schubert** has taught informatics in secondary, vocational and higher education. She has been professor of "Didactics of Informatics and E-Learning" (Universities Siegen and Dortmund, Germany) since 1998. Her research interests are Informatics Teacher Education and E-Learning. She is Fellow of the German Society for Informatics (GI), member of the Technical Committee 3 "Education" of the International Federation for Information Processing (IFIP) and chair of the IFIP Working Group 3.1 "Informatics and digital technologies in School Education".

BugHunt – A Motivating Approach to Self-Directed Problem-solving in Operating Systems

Simone Opel, Matthias Kramer, Michael Trommen, Florian Pottbäcker, Youssef Ilaghef
University of Duisburg-Essen
D-45127 Essen
{simone.opel, matthias.kramer, michael.trommen, florian.pottbaecker, youssef.ilaghef}@uni-due.de

Abstract: Competencies related to operating systems and computer security are usually taught systematically. In this paper we present a different approach, in which students have to remove virus-like behaviour on their respective computers, which has been induced by software developed for this purpose. They have to develop appropriate problem-solving strategies and thereby explore essential elements of the operating system. The approach was implemented exemplarily in two computer science courses at a regional general upper secondary school and showed great motivation and interest in the participating students.

Keywords: Educational software, operating system, student activation, problem-solving, interactive course, interactive workshop, edutainment, secondary computer science education

1 Introduction

The effective and efficient use of modern digital technologies has become a key competency in today's society (OECD, 2005), for both the private and the professional sector. In 2013, the Educational Testing Service (ETS) published an analysis (Burrus et al., 2013) on the most important 21st century workforce competencies by comparing three international 21st century skill frameworks, namely ATC21S, Finegold & Notabartolo and P21. A key component identified in view of the working requirements is "information processing" with the leading variable "computers and electronics". Students are being introduced to the underlying principles of such electronic systems in the context of computer science education. This should enable them to use this technology in a qualified way and further developing it in the future.

In Germany, recommendations for national educational standards for lower secondary computer science (Brinda et al., 2009) were developed by a task force of the German Informatics association (GI), and most current German computer science school curricula published since then are oriented on these recommendations. In these educational standards, informatics systems are established as one of five school-relevant content areas. Therefore, students of all ages are to understand the basics of the structure of informatics systems and their underlying working principles with the aim to apply these systems in an effective and efficient way and to be able to learn and to understand further systems. The process of identifying school-relevant topics focuses on long-lasting computer science concepts and principles (Schwill, 1994), particularly as specific products as key aspect have also been criticised, because products evolve and only little transferable knowledge can be developed from them. Nevertheless, an essential aspect in the educational process is to link such product knowledge with conceptual knowledge (Hartmann et al., 2006).

Although more and more people use mobile devices such as tablets, traditional computers and laptops are still of great importance. Their qualified usage requires conceptual and product knowledge of the software needed (e.g. to solve a given problem) and of the underlying operating system software. This is also supported by international curricula such as the ACM K12-Curriculum (Tucker et al., 2003), or the IFIP Curriculum (van Weert, Tinsley, 2000), which list knowledge on operating systems and file management as their desired outcomes.

There are numerous systematic approaches to teach the use of software in full width (such as courses preparing for the European Computer Driving License (ECDL), (ECDL, 2013a, 2013b)). However, such offers are rarely made for secondary school students and also do not focus on presenting content in a motivating and explorative way.

At a project course for student teachers of computer science at the University of Duisburg-Essen, Germany, during the summer term of 2013, the students (split in small groups) designed motivating computer science learning units for secondary school students. One of these teams (the last three authors of this paper) developed the idea of using special "educational viruses" ("bugs") to stimulate students to explore the underlying operating system. Students should deal with undesired system behaviour produced by these bugs and solve the arisen problems. In this paper, we (course tutors and students) present the educational concept, the design and implementation of the learning software for the "BugHunt" project as well as first results and experiences with its use in two computer science classes at a regional general secondary school.

2 Related Work

To find ideas on how to explore an operating system combined with the fostering of problem-solving skills, existing approaches in this field were analysed. It soon became clear that ideas for this approach were very rare, if not unique. Although universities teach courses in virus programming (e.g. Aycock, 2013), it does not seem probable that they are used with educative intention, least of all in secondary education.

In a first step, international curricula were reviewed. According to the ACM curriculum (Tucker et al., 2003), students should gain a conceptual understanding of principles of computer organisation and its major components, such as storage or the operating system. The "IFIP-Unesco: ICT Curriculum for Secondary Schools" (van Weert, Tinsley, 2000), appendix A1, contains the demand that "students should understand how computers and the basic operating system work and demonstrate that the computer is under their control". Appendix A8 demands "[that] students are expected to understand basic concepts such as […] computer security (theft, hacking, and viruses)". Another modularised ICT curriculum and course offer is the "European Computer Driving License" (ECDL), which contains learning modules on "computer essentials" (DLGI, 2013a) and "IT-Security" (DLGI, 2013b). Specific educational objectives are being listed, such as "understanding how to use an operating system to organize drives, folders and files in a hierarchical structure" or "knowing how malware can be hidden in the system".

Based on these overall objectives, in the second step it was necessary to theoretically establish the competencies needed to successfully complete the planned learning unit in a second step. Within the framework of the MoKoM project (Linck et al., 2013), a competence model for informatics modelling and system comprehension was theoretically derived and empirically refined. The final model consisted of the five dimensions system application (K1), system comprehension (K2), system development (K3), dealing with system complexity (K4) and non-cognitive skills (K5), with several competencies subsumed under each dimension. In detail, the model provides competence goals such as "systematically explore system functions (K1.2.1)", "independently explore systems (K2.3)" or "know & analyse architecture & organization (K2.5)" to which the educational concept of the BugHunt project intends to contribute.

In a third step, reports on existing approaches were reviewed. Tulodziecki (2000) wrote on how computer-based media can be used to teach not only computer science, but all kinds of subjects. He stated that exercises were more effective when having a personal relevance for the students (such as occurring in their everyday life), and when the problem was on the one hand not solvable with the current knowledge, but on the other hand not too sophisticated. Westram (2006) described the importance of the subject "IT security" and demanded that it should be compulsory in secondary education. In her opinion, questions like what viruses, worms or Trojan horses are and how to deal with them is something computer science education has to convey. To promote these goals, teachers have to be provided with the necessary teaching materials. Schlüter (2006) gave an example of how a unit on internet risks could be structured. She proposed that in a first lesson the students should have a look on the topic "computer viruses". As the unit progresses, the students reflect on the behaviour and risks of viruses as well as an appropriate behaviour in case of a virus attack. This is just one possibility for students to increase their awareness on dealing with a malware situation in general.

In summary, it can be stated that operating systems and security aspects as well as the understanding of the underlying computer science concepts and principles are relevant educational goals to which teaching approaches have been described, but not in the integrated way suggested in this work.

3 Requirement Analysis

Starting point of the development was the idea to create a motivating learning unit in which students can explore selected aspects of different versions of the operating system Microsoft Windows in a short period of time. The basic idea was to place special "educational viruses" ("bugs"), which cause undesired system behaviour, as a motivating element in a protected environment on the computers. To remove the undesired system behaviour, the students should explore the settings of the operating system on their own and with the help of a particular text book. Due to possible safety concerns in school or university networks, the operating system should be run in a virtual machine, although the software leaves no undesired system behaviour in the operating system after its termination and should not need any network or internet access.

The so-called "BugHunt" learning unit should be designed for group sizes of up to 30 students (working in pairs) and a duration of 90 minutes. The course should be implementable both in traditional teaching at a secondary school as well as in the context of university courses, so the software should include different levels of difficulty. In order to create real experience, it is necessary to provide the students not only with an operating system simulation, but to bring the undesired, virus-like system behaviour (such as slowing down the system or executing undesired functions) to the real system. Each student team should be provided with appropriate tasks ("bugs") and have to try to deal with them as independently as possible. This requires a variety of bugs in varying degrees of difficulty to support an internal differentiation of the learning group. Moreover, the undesired system behaviour should be reflected in clearly recognisable symptoms that can be identified easily and directly. Furthermore, an additional teaching text should be provided in which the symptoms of the induced undesired system behaviour are linked with hints to an appropriate problem-solving strategy. To ensure that the students have to remove the bugs following the given instructions, the software must be secured against unauthorized termination. For this reason, there must not be any loopholes or workarounds the students could use to terminate the bugs without solving the actual problem. Therefore, the teacher must be able to start the controlling software, configure the level of difficulty and the quantity of bugs and to terminate it separately from the actual bug programme. By processing the tasks, the students should build up or enhance computer science competencies as they have been described for example in the MoKoM project (Linck et al., 2013). For example, the students should systematically explore system functions (K1.2.1) and know and analyse architecture and organization (K2.5) of an operating system. They should learn by exploring the operating system independently (K2.3) and not by being taught the solutions and techniques in traditional teacher-oriented lessons. Being confronted with a series of problems on a running operating system, they should know about and evaluate consequences of informatics systems (K5.1.1.5) and finding solutions to the arisen problems in a self-directed way, we hope to increase their affinity and enthusiasm (K5.1.2.1) and make them willing to improve their informatics abilities and knowledge (K5.3.2.1). In order to prevent the students from copying the solutions from other teams, the sequence of released bugs must be randomized.

To minimize the preparation time of such a course, the teacher or university tutor should only have to prepare the computers by activating the software on each computer without the necessity of an installation process. During the course, the teacher should be able to configure the difficulty level and the selection of bugs provided to a student team. After showing the students how to use the software and the teaching material, the teacher should not need to give any further assistance except for answering questions or solving general technical problems.

Finally, the software should be executable on any computer or laptop with one of the operating systems Windows 8, 7, Vista, XP, 2000 installed and not require any special resources. It should be developed modularly for an easy extension by more or improved bugs.

4 Design and Implementation

According to the requirements, the software should be set up with minimal effort and without an installation process or any necessary runtime environment. At first, we had to decide which language should be used to implement the software. We decided to use AutoIt, a Basic like programming language, which provides access to the Microsoft Windows API by using C++ syntax in dll-calls (Aristides de Fez Laso, 2013; Petzold, 2013). Using this technology enables easy access to every input device and the desktop environment. Thus it was possible to develop bugs, which would be able to take control of the mouse cursor, the keyboard, the file system and the most common system functions. To secure the software against manipulation by students, it is developed to run in only one hidden process with a single thread. Other advantages of this solution are that the software needs just a small amount of system resources and that its tasks can hardly be recognised.

The Software is split into three parts: The BugHuntMaster, the BugHunt main process and the BugHunt recovery process (cf. Fig.1).

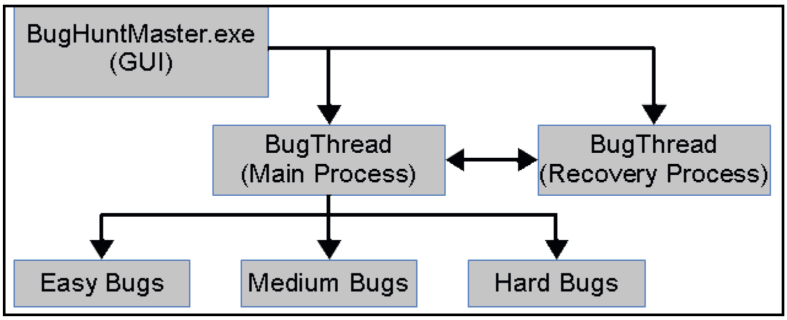


Figure 1: Sketch of the BugHunt architecture

The BugHuntMaster.exe runs portable from the BugHunt USB device. It ensures the teacher's control over the bugs. Each bug is programmed in a separate file, which contains the polled function of the bugs and its global variables.

The simple GUI (cf. Fig. 2) is built by an algorithm, which scans for implemented bugs during compilation, and is configured dynamically.

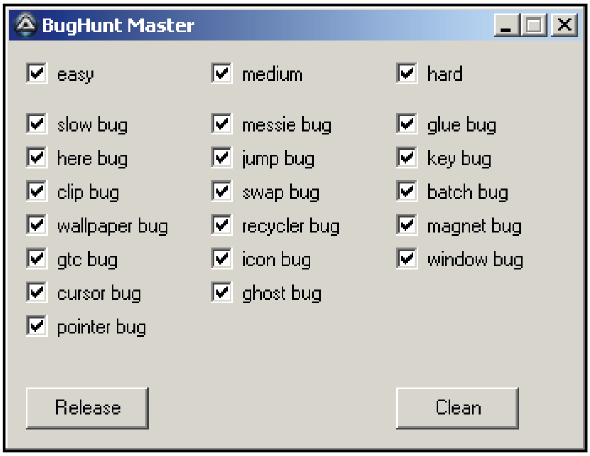


Figure 2: BugHuntMaster.exe GUI

The desired bugs can be activated individually or as a whole difficulty group by checking or unchecking the box on top of the column.

The main process runs the bug functions, which are arranged in three levels of difficulty. The bugs are not separate applications, but different functions in the main process. Despite this architecture, more than one bug can run simultaneously. The BugHunt main process executable is copied to the system by the BugHuntMaster.exe. That main process executable contains the bug functions along with a random bug activation algorithm according to the written configuration. Only one bug of each difficulty level can be active at the same time to prevent an unsolvable scenario. The main process simulates the symptoms and decides whether a bug has been solved or is still active. Furthermore, it alerts the student in case of success. The main process also ensures that the recovery process is running.

The recovery process monitors the main process in a 10ms interval and keeps it running. Moreover, it restores the BugHunt system files which may be deleted due to manipulation or system failure. If the main process is being terminated, the recovery process will start it again. In case of missing BugHunt files, it automatically restores them from backup files. This virus-like behaviour is to challenge the students to systematically explore possible problem-solving approaches without giving them the opportunity to simply stop the task and delete the BugHunt programme.

Despite those security features, the teacher should be able to stop all bugs immediately at any given time. Both BugHunt processes scan for the BugHunt USB device at the beginning of every iteration. If the device is being detected, both loops will terminate immediately. So the teacher can easily start the BugHuntMaster again and change the configuration or stop the programme.

5 The Bughunt Software

The teacher has to connect the BugHunt USB device to a computer. The BugHunt-Master.exe, which has to be executed, is located in the root directory of the BugHunt device. When the "RELEASE" button (cf. Fig. 2) is clicked, the BugHuntMaster.exe copies all BugHunt files to a hidden directory and creates two hidden backup copies of all BugHunt files at different locations. The BugHuntMaster.exe resides on the USB device and is never copied to the system. Thus, it is unreachable after the device has been removed. After the BugHunt configuration is written to the system, a message box will appear asking to disconnect the BugHunt USB device and click "OK". When the device has been disconnected the BugHunt main and the BugHunt recovery process (cf. Fig. 1) will be invoked after a short time of 10 seconds.

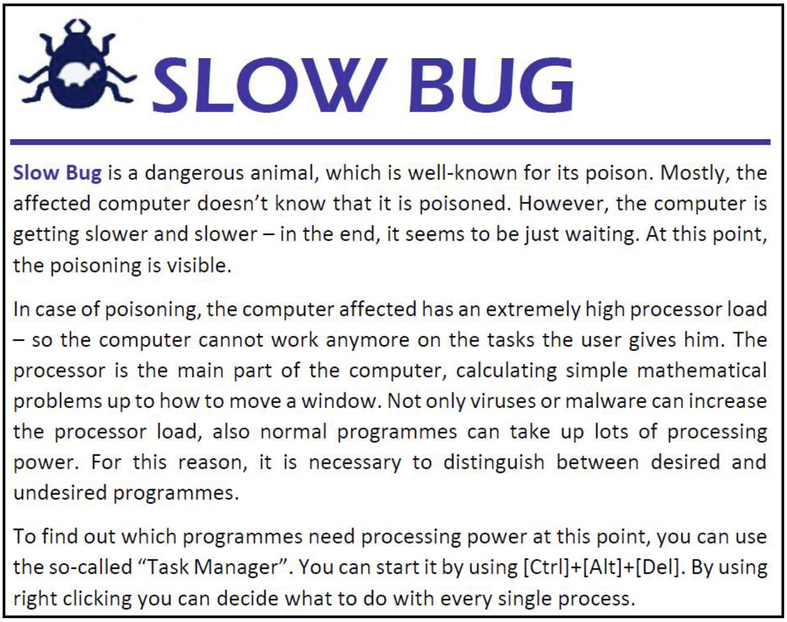
The "CLEAN" button stops each running BugHunt action. Additionally, it deletes all files from the system and resets all changes, which have been done. The software is fully removed in less than one second by only one click.

Exemplary "Bugs"

Subsequently, we describe the induced behaviour of selected bugs and the learning objectives to be achieved by their removal.

The *GTC Bug* (General Terms and Conditions Bug, category "easy") shows a dialog box, which asks the student to click "YES“ to remove unwanted software (apparently the bug itself). There is an obligatory checkbox at the bottom of this dialog box, which is already checked to accept the terms and conditions. This common situation postulates the student to know and evaluate consequences of informatics systems (K5.1.1.5). If the student just clicks "YES“ without at least having a quick view at the information, a random number of folders named "WASHING MACHINE MODEL_XYZ“ will be created on the desktop. The dialog box will be shown again after a few seconds. If the student decides to read the information, he will notice that he just has to uncheck the terms and conditions box to remove the bug. Accepting all given terms without reading the text itself is a common but careless behaviour for most computer users – not only for school students. This bug should encourage the students to reconsider their own behaviour and to learn how to act in a reasonable way.

The *Slow Bug* (category "easy") slows down the whole system. It is designed to simulate the well-known effect that hidden processes consume lots of resources without being identified as the root of the problem. To ensure that the bug is being noticed even on very powerful systems, the mouse cursor movement judders due to cursor manipulation by this bug. The student has to systematically explore system functions (K1.2.1) to find and open the task manager. All CPUs will show a usage of 100 %. The bug is removed by closing all "Slow Bug" processes, which are in the list.



SLOW BUG

Slow Bug is a dangerous animal, which is well-known for its poison. Mostly, the affected computer doesn't know that it is poisoned. However, the computer is getting slower and slower – in the end, it seems to be just waiting. At this point, the poisoning is visible.

In case of poisoning, the computer affected has an extremely high processor load – so the computer cannot work anymore on the tasks the user gives him. The processor is the main part of the computer, calculating simple mathematical problems up to how to move a window. Not only viruses or malware can increase the processor load, also normal programmes can take up lots of processing power. For this reason, it is necessary to distinguish between desired and undesired programmes.

To find out which programmes need processing power at this point, you can use the so-called "Task Manager". You can start it by using [Ctrl]+[Alt]+[Del]. By using right clicking you can decide what to do with every single process.

Figure 3: Bug description in the textbook, category "easy"

The *Swap Bug* (category "medium") makes the mouse cursor move in a random interval between 15 and 30 seconds. The movement of the cursor is not random; it always follows the same path. The student has to independently explore the system (K2.3) to find suitable graphic editing software. The path can be made visible by clicking and holding the mouse in a drawing area. The cursor will draw the letters "Ctrl B U G". The student has to identify the output as hotkey combination. The written capital letters must be identified as a combination of upper case characters. Hence, the actual hotkeys are "ctrl +

shift + character". The bug will be solved when the three hotkey combinations are pressed in order of appearance. Since the students can hardly control the mouse cursor, they have to use the keyboard to find and open the graphic software. Although all modern computers have mice or other graphical input devices, students become acquainted with an efficient way to use their computer by removing this bug.

The *Key Bug* (category "hard") encrypts the keyboard with a random Caesar cipher. Encryption in general is a fundamental concept in computer science and computer security, which is modelled by this bug. A text document called "Message from Key Bug.txt" will appear on the desktop. It contains an encrypted text, which in its decrypted version says "Who should I be afraid of?". The correct answer ("julius caesar" or "alan turing") has to be written to the file with the keyboard still being encrypted. The "bug identification textbook" contains some basic information about the bug and the idea of the Caesar cipher. The student has to identify the decryption and write the correct answer into the file to remove the bug. That means that the student has to understand and use decryption as well as encryption to solve the problem.

6 Implementation and Evaluation of the Learning Unit

6.1 Implementation of the learning unit

The course was carried out twice with 10^{th} grade students at ages between 14 and 17 years of two elective computer science classes of the same general upper secondary school ("Gymnasium") in a computer room supplied with personal computers and laptops. The students were divided into pairs with one pc or laptop per group. The pcs and laptops were prepared by creating a new user account on the operating systems for the course. On these accounts, the "BugHunt" software was started and easy bugs were chosen for every group to begin with. At the beginning of the course, an additional teaching text (the "bug identification textbook", cf. Fig. 3) was given to each group. After that, the students got a short introduction on how to use the book. We explained the different levels of difficulty and the general handling of the software. Then the students were asked to solve the problems caused by the bugs. They should first try to identify the type of the active bug and after that they should remove it, using the information given in the textbook. Afterwards, the students started to work. Since some questions were asked repeatedly, additional hints were written on the board. After about half of the time, several groups finished all bugs from the easy category and started with the medium bugs. Other groups however had difficulties with different bugs from the easy category, so they removed only these easy bugs.

6.2 Evaluation

Since a main component of the concept was to motivate the students and to interest them in operating system issues, the participating students were interviewed using a questionnaire after the implementation of the learning unit. For this purpose, we developed a questionnaire consisting of three parts to explore the motivation and interest of the students in partaking in this course.

In the first part we asked for personal information such as age and sex as well as experience in computer science lessons and career aspirations of the students.

The second part consisted of questions about their attitudes towards the course (six items) and their personal interests (four items). Exemplary items are “the course was exciting and interesting” or “I am interested in solving difficult problems”. We used a 4-point scale answering format with the options “yes” (4), “generally yes” (3), “generally no” (2) and “no” (1). The items are not standardised. For this reason, the internal consistency of the questionnaire about attitudes is only $\alpha = .641$ and about interest $\alpha = .800$.

In the third part, we used the questionnaire on subjectively perceived boredom in mathematics at elementary schools by Sparfeldt et al. (2009). Todt (1990) describes boredom as the opposite of interest (Todt, 1990), beyond that a highly negative correlation between interest and boredom has been identified (Lohrmann, 2008; Pekrun et al., 1998). Whereas interest and school grades are highly correlated, no correlation has been found between boredom and school grades (e.g. Dickhäuser, Stiensmeier-Pelster, 2003; Todt, 2000). Thus, perceived boredom seems to be a reasonable predictor to evaluate the interest of the students in partaking in the developed course. This part of the questionnaire consisted of 14 items and used a 5-point scale format from “never” (1) to “always” (5). The original questionnaire had a reliability of $\alpha = .957$. We modified this questionnaire for the purposes of the BugHunt course ($\alpha = .959$) by replacing “mathematics” with “this course”. Exemplary items are “In my opinion, the course was boring” or “During the course, I looked out of the window because I was bored”.

6.3 Results

As reported before, we performed this learning unit in two elective computer science courses (10^{th} grade) of the same general upper secondary school ("Gymnasium") with 20 students each. 38 of these students participated in the course. Two-sided t-tests showed that both courses originated from the same population. For that reason, both courses have been evaluated as one. The courses consisted of 30 boys and eight girls at ages between 14 and 17 (M = 15.47, *median* = 15). Only 16 students related that they had had elective computer science lessons in former school years.

The students reported a positive attitude towards the course (M = 3.265; SD = 0.413). In particular, the students related that they would like to have more similar courses, also on other topics from computer sciences. The students were mostly interested in computer science (M = 2.906; SD = 0.723).

Although the second part of the evaluation had a low internal consistency, it nevertheless confirmed the results of the questionnaire about the perceived boredom: The students reported only little boredom during the course (M = 1.705; SD = 0.845). These findings were also confirmed by several annotations on the evaluation sheets: "It was cool, completely different!", "The course was very interesting and informative. Surely, it would be possible to cover different problems, too" or "Very cool, it would be nice if you visited us again!". The most astonishing feedback was that one of the students encrypted his statement by using a Caesar cipher.

Furthermore, our study confirmed the findings of Pekrun & Hoffmann (1999) and Lohrmann (2008). We also found a negative correlation between perceived boredom and interest in computer science (r = -.567; p < .001) and between perceived boredom and attitudes towards the course (r = -.599; p < .001). This result was also confirmed by informal feedback and observation of the behaviour of the students during the course.

The previous knowledge of the students about operating systems was quite diverse. For this reason, the students began with very different approaches.

A few groups with advanced experience with the operating system tried to terminate the whole software by terminating the tasks in the task manager. Being unsuccessful, they attempted to figure out how the BugHunt software works by using all skills they had. After that, these groups started to solve the bugs the designated way. Generally, these groups approached all bugs by trying different strategies on their own and rarely by reading the "bug identification textbook".

Students with fewer previous skills mostly started with trial-and-error strategies to solve the problems caused by the bugs, but during the course they managed to develop more useful strategies, e.g. "first describe the problem, then read the identification textbook, after that start working by interpreting the given instructions". Following these rules, they were able to remove at least all easy bugs.

While observing the students, we noticed increasing problem-solving strategies. Furthermore, some students found creative ways to solve the given problems. For instance, to fix one of the bugs, it is necessary to freeze the picture by making a screenshot. Several students did not know how to do this, so two of them used their smartphones to take a photo of the screen. Even though this was not the intended way to solve the problem, it was a very creative solution.

All things considered, most students seemed to work in a motivated and interested way on the different given problems. The groups were very focused during the course. There was much intra-group and cross-group communication. At the end of the time, most of the groups reached the medium category. There were only a few groups who started the bugs from the hard category and also those who did not finish the easy one.

7 Summary and Outlook

In this paper, we have described a differing approach to operating systems and other computer science concepts that is based on the hypothesis that removing virus-like behaviour on the computer is motivating for students. For this purpose, a special learning aid, the BugHunt software, was designed which induces the system behaviour required on the respective computer. The process of removing the bugs required individual problem-solving strategies and made the students explore various computer science concepts. A written survey after an exemplary implementation showed that the students worked with great interest and motivation on the tasks and would wish for more such concepts. Although these results are very encouraging, is it necessary to evaluate and improve this concept with more students of different age.

Of course, we are aware that the induction of virus-like behaviour on school computers can be discussed controversially with regard to safety. For this reason, we recommend to run the software in a virtual machine, although it is safe to run it on a real computer.

The approach presented here does not claim to be a systematic approach to operating systems concepts, but it can be used to support corresponding teaching sequences and to promote the motivation of the students. The software has been designed in a way that the integration of other bugs is easily possible. More information on the BugHunt project can be found on the website http://udue.de/bughunt.

References

Aristides de Fez Laso, E. (2013). *Instant AutoIt Scripting*. Birmingham: Packt Publishing.

Aycock, John (2013). CPSC 527 – *Computer Viruses and Malware. University of Calgary, Department of Computer Science*. Retrieved February 10, 2014 from http://pages.cpsc.ucalgary.ca/~aycock/virus-info.html

Brinda, T., Puhlmann, H., Schulte, C. (2009). Bridging ICT and CS – Educational Standards for Computer Science in Lower Secondary Education. *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education (ITiCSE'09)*, 288–292.

Burrus, J., Jackson, T., Xi, N., Steinberg, J. (2013). *Identifying the Most Important 21st Century Workforce Competencies: An Analysis of the Occupational Information Network (O*NET)*. Educational Testing Service: Princeton.

Claus, V., Schwill, A. (2006). *Duden Informatik A–Z*. Mannheim: Dudenverlag.

Dickhäuser, O., Stiensmeier-Pelster, J. (2003). Wahrgenommene Lehrereinschätzungen und das Fähigkeitsselbstkonzept von Jungen und Mädchen in der Grundschule. *Psychologie in Erziehung und Unterricht, 50*, 182–190.

Dienstleistungsgesellschaft für Informatik – DLGI (2013a). *ECDL / ICDL Computer-Grundlagen*. Retrieved February 10, 2014 from http://www.ecdl.de/fileadmin/redaktion/Syllabi/ECDL_new/ECDL_Computer-Grundlagen_2013.pdf

Dienstleistungsgesellschaft für Informatik – DLGI (2013b). *ECDL / ICDL IT-Sicherheit*. Retrieved February 10, 2014 from http://www.ecdl.de/fileadmin/redaktion/Syllabi/ECDL_new/ECDL_IT-Sicherheit_2013.pdf

Hartmann, W., Näf, M., Reichert, R. (2006). *Informatikunterricht planen und durchführen*. Berlin: Springer.

Linck, B., Ohrndorf, L., Nelles, W., Neugebauer, J., Magenheim, J., Schaper, N., Schubert, S., Stechert, P. (2013). Competence model for informatics modelling and system comprehension. *Proceedings of the 4th global engineering education conference*, IEEE EDUCON 2013, 85–93.

Lohrmann, K. (2008). *Langeweile im Unterricht*. Münster: Waxmann.

Nussbaum, M., Infante, C. (2013). Guidelines for Educational Software Design That Consider the Interests and Needs of Teachers and Students. *Proceedings of the 13th International Conference on Advanced Learning Techniques, IEEE ICALT* 2013, 243–247.

OECD (2005). *The Definition and Selection of Key Competencies*. Retrieved February 10, 2014 from http://www.oecd.org/pisa/35070367.pdf

Pekrun, R. (1998). Schüleremotion und ihre Förderung: Ein blinder Fleck der Unterrichtsforschung. *Psychologie in Erziehung und Unterricht, 44*, 230–248.

Pekrun, R., Hoffmann, H. (1999). Lern- und Leistungsemotionen: Erste Befunde eines Forschungsprogramms. In Jerusalem, M., Pekrun, R. (Eds.), *Emotion, Motivation und Leistung* (pp. 247-268). Göttingen: Hogrefe.

Petzold, C. (2013). *Programming Windows: 6th Edition*. Redmond: Microsoft Press.

Schlüter, K. (2006). Gefahren im Internet. *LOG IN – Informatische Bildung in der Schule, 140*, 35–44.

Schwill, A. (1994). Fundamental ideas of computer science. *Bulletin of the European Association for Theoretical Computer Science, 53*, 274–295.

Sparfeldt, J., Buch, S., Schwarz, F., Jachmann, J., Rost, D. (2009). Rechnen ist langweilig – Langeweile in Mathematik bei Gundschülern (German). *Psychologie in Erziehung und Unterricht, 1*, 16–26.

Todt, E. (1990). Entwicklung des Interesses. In Hetzer, H., Todt, E., Seiffge-Krenke, I., Arbinger, R. (Eds.), *Angewandte Entwicklungspsychologie des Kindes- und Jugendalters (2. Aufl.)* (pp. 213–264), Heidelberg: Quelle & Mayer.

Todt, E. (2000). Geschlechtsspezifische Interessen – Entwicklungen und Möglichkeiten der Modifikation. Empirische Pädagogik, 14, 215–254.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A (2003). *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee*. Retrieved February 10, 2014 from https://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf

Tulodziecki, G. (2000). Computerbasierte Medien. LOG IN – *Informatische Bildung in der Schule, 5/2000*, 8–12.

van Weert, T., Tinsley, D. (eds.) (2000). *Information and Communication Technology in Secondary Education – A Curriculum for Schools*. Paris: UNESCO.

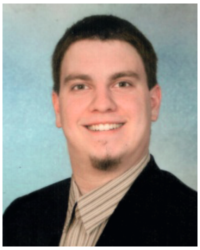
Westram, H. (2006): IT-Sicherheit im Unterricht. *LOG IN – Informatische Bildung in der Schule, 140*, 20–24.

Biographies

Simone Opel studied Information Technology at the University of Applied Sciences of Nuremberg and Vocational Education for Electrical Engineering and Computer Science at the University of Erlangen-Nuremberg. She worked as a trainer for computer science and teacher at several vocational schools. Since 2010, she is working as a scientist in the "Didactics of Informatics" groups at the Universities of Erlangen-Nuremberg (until Oct. 2012) and Duisburg-Essen (since Nov. 2012).

Matthias Kramer studied Computer Science and Mathematics for grammar schools at the Friedrich-Schiller-University in Jena. He completed his internship of teaching practice in January 2013. Since April 2013, he has been working as a research assistant at the chair of "Didactics of Informatics" at the University of Duisburg-Essen.

Michael Trommen took his O-Level at the Gymnasium Rheinkamp in 2001 and got his apprenticeship diploma as IT system technician in 2005. He took his high school diploma (Abitur) at the Abendgymnasium Duisburg in 2010. Since 2010 he is studying Computer Science and Mathematics for grammar schools at the University of Duisburg-Essen.



Florian Pottbäcker took his high school diploma (Abitur) at the Krupp-Gymnasium grammar school in Duisburg (in June 2009). Since October 2010 he is studying Computer Science and Mathematics for grammar schools at the University of Duisburg-Essen.

Youssef Jlaghef got his high school diploma (Abitur) in June 2006 at the Mercator Berufskolleg in Moers. He is studying Computer Science and Mathematics for grammar schools at the University of Duisburg-Essen since October 2010.

Improving children's Cognitive Modifiability through Mediated Learning and Dynamic Assessment within 3D Immersive Virtual Reality Environment

David Passig, David Tzuriel, Ganit Eshel Kedmi
Bar-Ilan University
Ramat-Gan, Israel
{david.passig, david.tzuriel}@biu.ac.il, geshelk@gmail.com

Abstract: The objectives of this study were to examine (a) the effect of dynamic assessment (DA) in a 3D Immersive Virtual Reality (IVR) environment as compared with computerized 2D and non-computerized (NC) situations on cognitive modifiability, and (b) the transfer effects of these conditions on more difficult problem solving administered two weeks later in a non-computerized environment. A sample of 117 children aged 6:6-9:0 years were randomly assigned into three experimental groups of DA conditions: 3D, 2D, and NC, and one control group (C). All groups received the pre- and post-teaching Analogies subtest of the Cognitive Modifiability Battery (CMB-AN). The experimental groups received a teaching phase in conditions similar to the pre-and post-teaching phases. The findings showed that cognitive modifiability, in a 3D IVR, was distinctively higher than in the two other experimental groups (2D computer group and NC group). It was also found that the 3D group showed significantly higher performance in transfer problems than the 2D and NC groups.

Keywords: Dynamic assessment, mediated learning experience, cognitive modifiability, analogical thinking, virtual reality

1 Introduction

1.1 Dynamic Assessment

A growing number of research evidences in the literature of cognitive evaluation shows significant contribution of the dynamic assessment (DA) approach in obtaining a richer and more reliable feedback with respect to (a) children's cognitive capacity (b) construction of intervention programs and (c) effective programs for the development of the abstract thinking (Tzuriel, Klein, 1985; Tzuriel, 2001; Tzuriel, 2000; Tzuriel, Caspi, 1992; Tzuriel, Kaufman, 1999; Tzuriel, Shamir, 2002; Tzuriel, Shamir, 2007; Tzuriel, Shamir, 2010).

The DA approach is based on the Structural Cognitive Modifiability (SCM) and the Mediated Learning Experience (MLE) theory and forms a key working assumption in the current research (Tzuriel, 2001, 2002; Feuerstein, et al., 1980, 1991; Feuerstein, Klein, Tennenbaum, 1991). The DA approach to the measurement of the learning process represents a relatively new trend in evaluating learning potential (cognitive modifiability) and is offered as an alternative with an advantage over the static assessment (SA) in evaluating the child's cognitive ability. While the conventional static procedure measures only the level of the subject's achievements, without any attempts at intervention, the focus of DA is on the observing and measuring of the learner's cognitive modifiability with the assistance of adequate MLE (Tzuriel, 2001).

The concept of cognitive modifiability refers to structural change brought about with the help of intervention, which guides the individual's absorption of external stimuli (Lidz, 1991; Tzuriel, 2000). The measurement process of the cognitive modifiability in a dynamic assessment consists of a pre-test which provides a preliminary evaluation of an initial performance, a learning phase which includes mediation by an adult and a post-test to examine post-learning performance.

1.2 Dynamic assessment in computerized environments

Numerous studies indicate, alongside the developments in the DA, that the use of computerized environments, including virtual reality environment, contributes to the development and empowerment of children's thinking ability (Klein, Nirgal, Darom, 2000; Tzuriel, Shamir, 2007; Clements, Samara, 2002; Passig, Neuman, Eden, 2002, Passig, Miler, 2014; Passig, 2013; Passig, Eden, Rosenbaum, 2008; Passig, Eden, 2002).

In light of the findings attesting to the significant contribution of a computerized environment to the learner's thinking development, well as the findings indicating that the DA process provides a clearer picture of the child's learning potential, we have decided to integrate the two domains. In the current research we focus on examining the learner's cognitive modifiability, through experiencing various environments, including computer environments, using a DA approach. The diverse learning environments in which we conducted the DA process were: (1) a three-dimensional immersive environment (3D Immersive) via virtual reality technology with three-dimensional immersive Head Mounted Display-HMD (enabling the subject to feel as if he or she were immersed within the virtual world) (2) two-dimensional computerized environment with a mouse-screen interface (where the virtual world is in front of the subjects) (3) board and blocks (with no technological aids). DA in a virtual reality environment (three-dimensional immersive environment – 3D IVR) is the first known study.

In the process of the DA we have examined the following questions: (a) in which learning environment will children show higher cognitive modifiability? (b) Does DA provide a more accurate measurement of the learning potential than static assessment? Likewise, we examined the degree of the learning potential over time under various learning conditions.

1.3 Transfer test

An important aspect of the present study was the transfer test of the principles learned in the DA procedure regarding problem solving of a higher order. Transfer is the effective and reasoned use of principles, relationships, and strategies at the time of carrying out a task perceived by the examiner as clearly more difficult than the tasks whose frameworks were taught (Salomon, Perkins, 1989).

An additional aspect, in which the concept of transfer was examined, touched on the correlation between the use of computerized technology and the improvement of cognitive skills over time, as opposed to the improvement in cognitive skills over time without technology. A few studies (Pea, 1987; Salomon, Perkins, Globerson, 1991) posited a distinction between two processes in which technologies impact cognition over time. One included a process called the "Effect with Technology," while the other is called the "Effect off Technology."

The first process addresses the changes in achievements, which happen in the course of interaction with technology, and therefore, is called the "Effect with Technology". The second process addresses the effect on the cognitive ability of the user over time. The intellectual partnership with the computerized tool leaves a cognitive imprint transfer on different cognitive abilities, such as the ability to generalize and self-regulate.

In the present study we assumed that the subject's experience in DA in a variety of environments, via the computer and through the use of wooden cubes, would not only affect the learner's achievements at the time of the assessment, but would also be preserved and consequently manifested even at a later stage.

1.4 Analogical thinking and dynamic assessment

The cognitive domain was selected from a major field in children's cognitive development – analogical reasoning. It constitutes one of the important fields in evaluating cognitive capacities and considered central to the measurement of learning processes and mathematical thinking (Holyoak, 2004; Halford, 1993; Sternberg, 1977; Goswami, 1992). Analogical thinking is strategic thinking, which enables children to reach conclusions about phenomena, which are presented to them for the first time (Holyoak, 2004). In a number of studies it was shown that infants demonstrated an ability to solve analogical problems at the age of 18 months, but failed to reach a high level of ability by the time they have reached puberty (Richland, Morrison, Holyoak, 2006). Although the overall consensus is that analogical capability is important to a child's cognitive development, there is a lack of agreement regarding the mechanism involved in developing analogical conclusions.

One of the interesting findings, which surfaced from the research based on the Dynamic Assessment approach over the last decade, is that children succeed in solving analogical problems on a much higher level after a short, intensive phase of learning (Tzuriel, 2000, 2001, 2007; Tuntler, Resing, 2007). In those studies, which examined children's analogical ability, researchers found that mediation in analogical thinking relevant to children based on familiar relationships with visual and concrete imaging or gaming, helped young children in analogies solving (Richland, Morrison, Holyoak, 2006).

To sum, the main hypothesis was that children's cognitive modifiability in analogical thinking, in a DA process, within a three-dimensional immersive computer environment would be higher than in a two-dimensional computerized environment, in a non-computerized environment and in a control group.

Furthermore, it was hypothesized that children's cognitive performance scores in transfer analogies (i.e., more complex than those tested at the DA stage) given two weeks later in a non-computerized environment, would be higher in the three-dimensional immersive computerized environment group as compared with the other experiential groups and the control group.

2 Method

2.1 Subjects

The sample was composed of 117 children at the age of 6:6 to 9:0 years, 61 boys and 56 girls. All children attended schools in the central region of Israel and were randomly selected from 4 schools. The children were randomly assigned into four groups: three experimental and one control group. The experimental groups participated in three different DA environments (i.e., three-dimensional immersive computer environment (3D IVR), two-dimensional computerized environment (2D), and non-computerized environment (NC)); whereas the control group participated in a NC environment, in which cognitive performance measurement was held with no learning phase. Following in table 1 is the breakdown of the groups by gender.

Table 1: Breakdown of the research groups by gender

Groups	Gender			
	Boys		Girls	
	N	%	N	%
1-3D-IVR	19	52,6	17	47,4
2- 2D	21	58,3	15	41,7
3-Blocks	12	50,0	12	50,0
4- Control	9	42,9	12	57,1
Total	61	52,1	56	47,9

In general, the number of boys was somewhat larger than the number of girls, 52.1 % vs. 47.9 %. $\chi 2$ analysis did not indicate a statistically significant difference between the groups $\chi 2 = .75$, df =1, ns.

It is worth noting that since the focus of this study was about DA in a computerized environment, we decided, already at the planning stage, that the groups assessed through the computer would be bigger than the other two groups.

In the course of the research we also examined the age of the children and the education level of the parents. According to the findings, there were no statistically significant differences among the four groups studied in our research, except significant difference in the NC group in which the fathers' education level was relatively lower than in that of 3D IVR Group.

2.2 Research instruments

Analogies sub-test from the CMB test (Tzuriel, 1995) was used for the purpose of examining cognitive modifiability in analogical reasoning. The test was designed for children in kindergarten and those attending first through fourth grades The test is built of a board, 18 cm x 18 cm which includes 9 windows set in a format of 3 x 3, with 64 wooden cubes in four colours (yellow, blue, red and green). Each coloured brick has four lengths (2 cm, 3 cm, 4 cm, and 5 cm). Four windows are open at the top of the board. The examiner places the bricks in three of the four open windows, and asks the child to complete the placing of the bricks in the fourth window. The problems are based on dimensions of colour, height, number, and location (for example, figure 1 – problem #14 from post-learning stage). The test has problems of three levels of difficulty, derived from the number of dimensions included in the problem. The problems are organized from easy to difficult.

The test also produced a measure of transfer scores. The goal of the transfer problems was to assess the degree of internalization of the principles of problem-solving through the use of analogical thinking, which were taught in the first stage.

The test on transfer was administered according to the static assessment approach, which included problems with no a learning stage. An example of the problem of transfer (TR8-A) can be seen in figure 1.

This test was also converted to a computerized version. The computer program made it possible to observe the problem from three angles: from above, from the side, and from within. We placed three buttons in the upper centre of the screen, and while pressing any one of them made it possible to move from one to any other angle of observation on the problem. In figure 2 there is a sample of an analogical problem from side angle.

In addition, the computerized program was written to enable the problem to rotate on a 360º horizontal axis (and thus made it possible to observe it from several perspectives) and at a 45º angle on a vertical (up and down) axis.

The grading system, in the DA and transfer stages, in both versions (original CMB test and computerized CMB test) was carried out according to the measure-research approach, which included two methods of grading: 1. all or nothing, and 2. partial scores. In the all or nothing approach, a score was registered when all of the problem`s dimensions have been identified correctly.

The total number of points for each stage in all or nothing method (pre or post-learning) was 14 points. Based on this scoring method graded the level of analogical performance of the pre-learning and the post-learning stages. In the partial scores method, the scores were given for each correct identification of one of the dimensions (colour-(c), number-(n), height-(h), location-(l)). For each correct answer the child was scored with one point. Based on this scoring method we produced the grade representing the level of analogical performance of each of the analogy's dimensions. The total number of points was 56 points. The advantage of assessing each child with two methods lies with the recorded gap between the two results. This indicated a difficulty in the integration of the dimensions in solving the problem (Tzuriel, 2001). In total, from the CMB-AN test we obtained three measures: pre-learning score, post-learning score and transfer score. The Cronbach's alpha reliability coefficient of the wooden bricks' format of the CMB-AN test was found to be α = .83 for pre-learning and α = .78 for post-learning (Tzuriel, 2000).

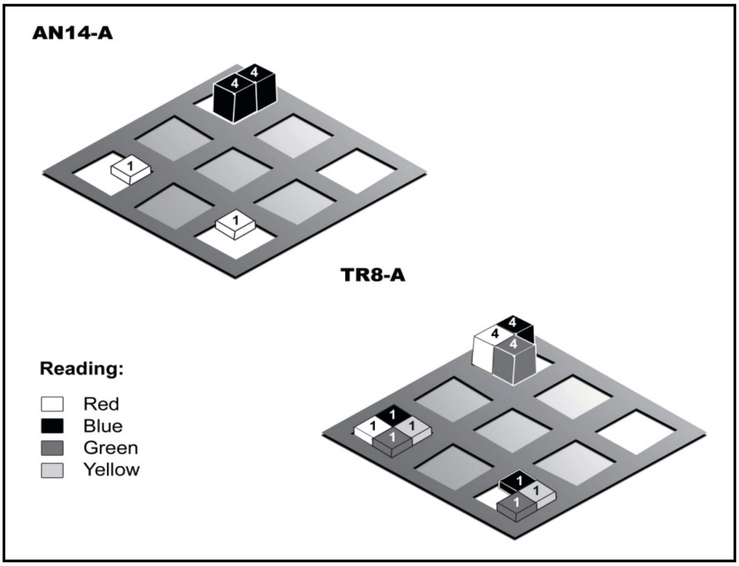


Figure 1: Analogical problem #14 from post-learning stage (AN14-A) and Transfer problem (TR8-A)

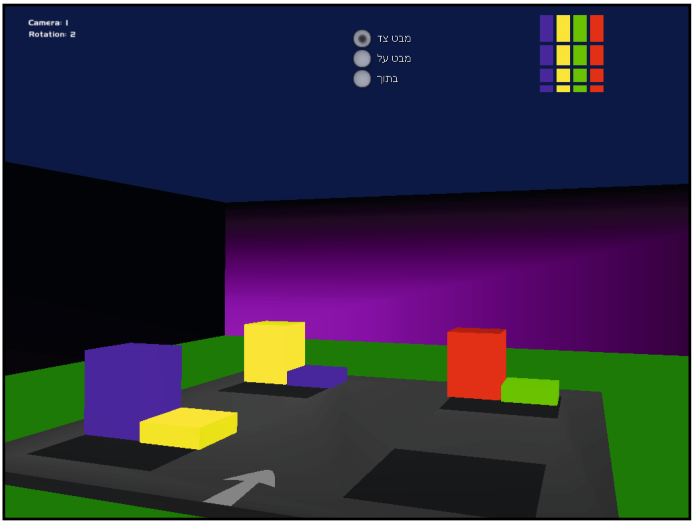


Figure 2: Presentation of problem TR2-B on the multimedia board as seen from the side

2.3 Procedure

The research consisted of two measuring stages, conducted two weeks apart from one another. The first consisted of DA in analogical thinking in the various diagnostic environments; the second consisted of transfer problem solving (more complex problem) in a non-computerized environment.

The assessments were performed in a small room assigned for that purpose by the school. Individual DA procedures were carried out in the first phase in all experimental groups.

All groups received the pre- and post-teaching Analogies subtest of the Cognitive Modifiability Battery (CMB). The experimental groups received a teaching phase in conditions similar to the pre- and post-teaching phases. In this way we assessed the level of the subjects' cognitive modifiability. The amount of time allotted for the DA procedure in the three groups was identical: 90 minutes divided equally for each stage (30 minutes each).

The assessment included all the items of the CMB-AN test. Each part of the assessment included 14 items. In the control group we measured cognitive achievements with problems of pre- and post-learning phase without the learning stage.

3 Results

3.1 Differences among the groups in cognitive modifiability

Table 2: Averages, standard deviations, and F analyses of analogy scores pre- and post-learning among all four groups

Research Groups											
		3D-IVR		2D		Blocks		Control		Group X Time	
		Pre	Post	Pre	Post	Pre	Post	Pre	Post	F(3,113)	Eta^2
Scores	M	2.58	10.72	4.02	9.75	4.70	10.45	4.00	4.19	25.18***	.40
	SD	3.27	3.89	3.36	2.87	4.49	3.20	4.42	3.57		

Table 3: Covariance analysis of the comparison between couples of four experimental groups in the pre- and post-learning (DA)

Group comparison	df	F	Eta^2
3–4	1,43	28.53***	.40
2–4	1,55	49.11***	.47
1–4	1,55	117.70***	.68
2–3	1,58	.00	.00
1–3	1,58	5.88*	.09
1–2	1,70	9.89**	.12

3.2 Differences between the groups in the dimensions of analogical thinking

Table 4: Averages, standard deviations, and F analyses of the four dimensions in the various groups and the MANOVA results for each dimension separately.

Dimensions		3D-IVR		2D		Blocks		Control		Time X Gr.	
		Pre	Post	Pre	Post	Pre	Post	Pre	Post	F(3,113)	Eta²
H	M	9.47	12.02	9.36	11.58	10.0	13.16	9.95	10.14	*2.82	07.
	SD	3.56	43.	3.81	43.	3.16	52.	3.90	5.64		
L	M	9.55	12.25	9.13	12.05	9.33	11.58	9.71	10.76	1.85	04.
	SD	2.71	38.	3.04	38.	2.69	46.	4.31	50.		
C	M	10.52	13.50	11.13	13.00	9.16	13.08	10.52	10.42	3.83*	09.
	SD	4.84	34.	3.79	34.	4.44	42.	3.41	45.		
N	M	11.69	13.33	11.11	12.72	9.16	12.70	9.57	9.66	3.56*	05.
	SD	2.85	36.	3.38	36.	4.74	44.	4.08	48.		

3.3 Differences between the groups in the transfer test of analogical thinking

Table 5: Averages, standard deviations, F analysis of cognitive performance in the transfer test in all four research groups.

Groups							
		3D-IVR	2D	Blocks	Control	F(3,113)	² Eta
Transfer Analogies Scores	M	5.32	3.59	3.50	1.47	17.34***	.32
	SD	2.47	1.76	2.02	1.20		

***$P < .001$

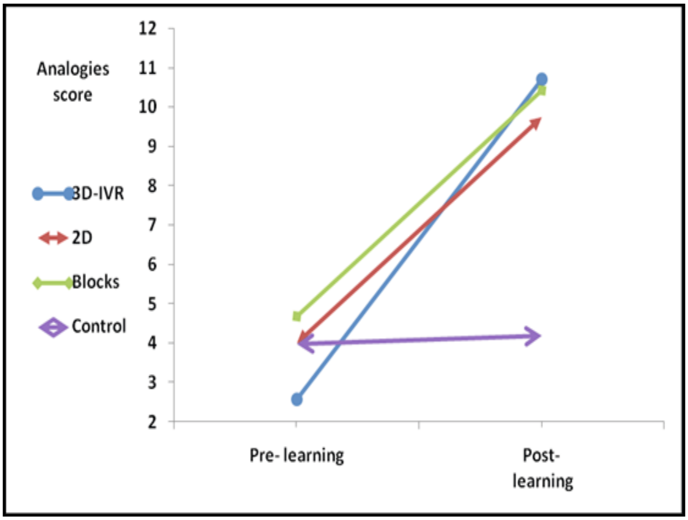


Figure 3: Average analogies scores of the pre- and post-learning among the four experimental groups

3.4 Differences between groups in the dimensions of the transfer test

Table 6: Averages, standard deviations, and the F analysis of different dimensions in the transfer test by the four research groups

Dimen-sions		Groups					
		3D-IVR	2D	Blocks	Control	F(3,113)	Eta²
H	M	10.75	10.36	9.95	8.66	3.91*	.09
	SD	1.72	1.95	2.47	3.24		
L	M	9.75	9.75	9.04	8.61	1.28	.03
	SD	2.40	1.82	2.09	3.91		
C	M	12.41	12.13	11.87	9.90	8.18**	.18
	SD	1.18	1.29	2.55	2.93		
N	M	10.61	9.75	10.08	9.47	1.47	.04
	SD	1.98	2.37	1.93	2.56		

*P< .05, **P < .01

4 Discussion

4.1 Differences among the groups in cognitive modifiability

The research findings indicate distinct improvement in the analogies scores from pre- to post-teaching phases in the experimental groups. As hypothesized, cognitive modifiability in a 3D IVR was distinctively higher than in the two other experimental groups (2D computer group and NC group).

These results support findings of other VR studies, and add another important layer. From our findings we learn that the experience in solving problems with the assistance of virtual reality can improve cognitive abilities. Our study, in this regard, supports Passig's (2014) claims and broadens his scope to include the areas of mediated learning and Dynamic Assessment. But different from Steinwandel and Ludwig's (2011) results where recognition and processing of spatial structures within the working environment "model" was superior to the other two forms of representation – like illustration or interactive animation.

As opposed to earlier research in the fields of mediated learning and DA, this study adds an additional layer by integrating a DA procedure of analogical thinking with 3D IVR. Indeed, it seems that a DA procedure in a 3D IVR setting can better reflect the subject's potential for learning than other settings. One possible explanation for this lies in the manner in which we use virtual reality. The improvement of cognitive skills derives from the possibilities embedded within this technology to present abstract concepts in concrete, visual, three dimensional, and game oriented ways. It is well established from earlier research, in the field of the development of analogical thinking in early childhood, that when analogies are presented to children by means which are both familiar to them and which in their view have concrete significance; they do well at solving them (Goswami, 1992; Halford, 1993). These characteristics, embedded in the nature of the VR technology, seem to have expanded the ways in which information is presented, as well as having assisted the young children's ability in the course of the DA procedure to reach an analogical conclusion.

It seems that in the course of learning and assessment, the children's opportunities for gaining concrete experiences are empowered by means of exposure to additional information – both visual and new which are solely virtual. It seems that this visual information stimulates a unique perceptual experience which contributes to the understanding of the transformations in the dimensions of the problem and creation of new and more broadened representations as well as schemes which empower the children, and present them with the ability to solve problems.

On the basis of these findings, one may conclude that the use of the virtual reality environment contributes to the empowerment of the children's cognitive capacity. We may further conclude that integrating the DA into a three-dimensional immersive environment reflects to a greater extent the learning potential of the subject's in comparison to the DA in the other research diagnostic environments.

4.2 Dynamic assessment in the context of the four dimensions of the test

The performances requiring analogical thinking was also tested with four additional dimensions: location, height, number and colour. In testing the impact of mediation on the performance in the different dimensions, we found a statistically significant difference between the first and the second measurements, though not find a significant difference in cognitive performances on the various dimensions between the 3D IVR group and the rest of the experimental groups.

We may explain the difference between the results of the first to the second by means of the differing approaches with which we scored the achievements. In measuring the score for the analogical from pre- to post-learning, we adopted the approach of 'all or nothing'. In this approach the emphasis was on the complete solution of the problem. In this scoring method, the subject must weigh a number of transformations together and provide one answer. Only a correct answer in all four dimensions would give him or her one credit. However, in measuring the score with each of the test dimensions, we applied the partial scoring approach, according to which the score given to the correct solution of each of the dimensions was calculated separately, with no interdependence between them. According to this approach, it was possible the subject to solve three out of four dimensions correctly and to receive for this a partial grade of three points (one point for each correct answer). It may be that for each scoring method, different thinking abilities are required, and thus the 'all or nothing' method required integration of all the dimensions. That way, the advantage we found in the 3D IVR environment overall score was not preserved in each of the dimensions of the analogy.

We may summarize by saying that the DA experience with 3D IVR had an impact on the cognitive performance of the child in a way that it improved his other ability to generally observe the problem, simultaneously address the transformations which occurred in the dimensions of the analogy, and generate a valid integration between them towards its full solution. Accordingly, when dealing with a solution involving each dimension separately, the DA experience with 3D IVR had a similar impact as a DA experience with a computerized 2D or a wooden board and bricks settings.

4.3 Differences between the groups in the transfer Test of analogical thinking

Moreover, it was found that in the transfer test, held two weeks later in an NC environment and consisted of more complex problems, the cognitive performance, among the subjects who experienced an assessment in a 3D IVR, was maintained and distinctively higher as opposed to the subjects' achievements in the other groups. The DA in the 3D IVR was more effective in internalizing the mediated cognitive principles, namely in the ability to apply them in solving more complex problems. These findings point both to the credibility of the results, obtained at the DA stage, and to the possibility of maintaining and 'transferring' the level of achievements, measured in assessment in a three-dimensional environment to an environment with no technological aids.

Integrating the use of a 3D IVR in a DA procedure generates "an intellectual collaboration" (Pea, 1987; Salomon, Perkins, Globerson, 1991) among the computer, the subject and the examiner. This collaboration apparently creates a unique perceptual experience, which broadens the subject's mental imagery world, heightens the internalization of the mediated cognitive principles and contributes to its performance. The virtual reality technology, therefore, is an appropriate and important diagnostic environment.

As in the first phase no differences were found between the 3D IVR to the other groups in terms of the scores of the test's dimensions.

5 Conclusion

This research has increased our understanding regarding the contribution of integrating the use of computerized environments in DA processes. The current research is added to a limited number of earlier studies, which had examined thinking development in a virtual reality environment as well as to a line of research in the DA domain. It may be inferred from the current research findings as a whole that integrating the virtual reality technology in a DA procedure is one of the effective means of a computer use in this procedure. Thus, a DA of an analogical reasoning capacity in such environment reflected the child's learning potential to a greater extent in comparison with a 2D computerized environment and a NC environment.

Consequently, we suggest that the evaluation of the child's cognitive modifiability capacity is affected by the environment where the assessment is conducted through the collaboration between the child, the computer and the examiner.

In practice, the research has clinical and educational applications. Based on the current research findings, we may conclude that diagnosticians and educators can relate to the DA results in a 3D-IVR as predictive of cognitive modifiability capacity in reality. A possibility is opened for diagnosticians to consider and select out of a number of diagnostic environments the one, which would best reflect the child's learning potential. Conducting the DA in the virtual reality is an additional layer in the development and integration of dynamic assessment processes in computerized worlds and advanced technologies.

References

Clements, D., Samara, J. (2002). The role of technology in early childhood learning. *National Council of Teachers Mathematics, 8*, 340–343.

Eden, S., Passig, D. (2007). Three-dimensionality as an effective mode of representation for expressing sequential time perception. *Journal of Educational Computing Research. 36*(1), 51–63.

Feuerstein, R., Rand, Y., Hoffman, M. B., Miler, R. (1980). *Instrumental Enrichment: An Intervention Program for Cognitive Modifiability*. Baltimore: University Park Press.

Feuerstein, R., Klein, P. S., Tennenbaum, A. (Eds.) (1991). *Mediated Learning Experience*. London: Freund.

Goswami, U. (1992). *Analogical Reasoning in Children*. Hillsdale, NJ: Lawrence Erlbaum.

Halford, G. S. (1993). *Children's Understanding: The Development of Mental Models*. Hillsdale, NJ: Lawrence Erlbaum.

Holyoak, K. J. (2004). Analogy. In Holayoak, K. J., Forrison, R. J. (Eds.). *The Cambridge Handbook of Thinking and Reasoning*. New York: Cambridge University Press, 117–142.

Klein, P., Nir-Gal, O., Darom, E. (2000). The use of computers with or without adult mediation: Effects on children's cognitive performance and behaviour. *Computers in Human Behavior, 16*, 591–608.

Lidz, C. S. (1991). *Practitioner's Guide to Dynamic Assessment*. New York: Guilford.

Passig, D., Neuman, T., Eden, S. (2002). Improving the awareness to toddlers' initial emotional experiences in kindergarten with virtual reality. *Educational Media International, 39*(2), 185–193.

Passig, D., Eden, S. (2002). Virtual reality as a tool for improving spatial rotation among deaf and hard-of-hearing children. *CyberPsychology & Behavior, 4*, 681–686.

Passig, D., Eden, S., Rosembaum, V. (2008). The impact of virtual reality on parents' awareness of cognitive perceptions of a dyslectic child. *Education and Information Technologies, 13*(4), 329–344.

Passig, D. (2013). *Forcognito – The Future Mind*. Tel-Aviv: Yediot Press.

Passig, D., Miler, T. (2014). Solving conceptual and perceptual analogies with virtual reality among kindergarten children of emigrant families. *Teacher College Record, in print on 116*(2).

Pea, R. D. (1987). Integrating human and computer intelligence. In Pea, R. D., Sheingold, K. (Eds.), *Mirrors of Mind: Patterns of Experience in Educational Computing*. Norwood, NJ: Albex, 128–146.

Richland, E. A., Morrison, G. R., Holyoak, J. K. (2006). Children's development of analogical reasoning: Insights from scene analogy problem. *Journal of Experimental Child Psychology, 94*, 249–273.

Salomon, G., Perkins, D. N. (1989). Rocky roads to transfer: Rethinking mechanism for a neglected phenomenon. *Educational Psychologist, 24*, 113–142.

Salomon, G., Perkins, D., Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researchers, 20*(3), 2–9.

Steinwandel, J., Ludwig, M. (2011). Identifying the structure of regular and semi regular solids – a comparative study between different forms of representation. In *Proceedings of the Seventh Congress of the European Society for Research in Mathematics Education*, 670–680. Rzeszów, Poland: European Society for Research in Mathematics Education.

Sternberg, R. J. (1977). Component processes in analogical reasoning. *Psychological Review, 84*, 353–378.

Tunteler, E., Resing, C. M. (2007). Effects of prior assistance in using analogies on young children's unprompted analogical problem solving over time: A micro genetic study. *British Journal of Educational Psychology, 77*, 43–68.

Tzuriel, D. (1995). *The Cognitive Modifiability Battery (CMB): Assessment & Intervention. Instruction Manual*. School of Education, Bar-Ilan University: Ramat-Gan, Israel.

Tzuriel, D. (2000). Dynamic Assessment of young children: Educational and intervention perspectives. *Educational Psychology Review, 12*, 385–435.

Tzuriel, D. (2001). *Dynamic Assessment of Young Children*. New York: Kluwer Academic/Plenum Publishers.

Tzuriel, D. (2007). Transfer effects of teaching conceptual versus a perceptual analogies. *Journal of Cognitive and Education Psychology, 6*(2), 194–217.

Tzuriel, D., Klein, P. S. (1985). Analogical thinking modifiability in disadvantaged, regular, special needs, and mentally retarded children. *Journal of Abnormal Child Psychology, 13*(4), 539–552.

Tzuriel, D., Caspi, N. (1992). Dynamic assessment of cognitive modifiability in deaf and hearing preschool children. *Journal of Special Education, 26*, 235–252.

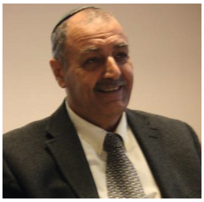
Tzuriel, D., Kaufman, D. (1999). Mediated learning and cognitive modifiability: The dynamic assessment of young Ethiopian immigrants in Israel. *Journal of Cross-Cultural Psychology, 13*, 539–552.

Tzuriel, D., Shamir, A. (2007). The effects of peer mediation with young children (PMYC) on children's cognitive modifiability. *British Journal of Educational Psychology, 77*, 143–165.

Tzuriel, D., Shamir, A. (2010). Mediation strategies and cognitive modifiability in young children as a function of peer mediation with young children program and training in analogies versus math tasks. *Journal of Cognitive Education and Psychology, 9*, 48–72.

Biographies

David Passig is an Associate Professor at the school of Education at Bar-Ilan University in Israel. He is the Director of the Graduate Program in Information and Communication Technology and Education. He teaches graduate courses and conducts research on Educational Futures, Future Technologies, Social Systems Theories, Futures' Methodologies, and Virtual Reality.



David Tzuriel is a Professor Emeritus at the school of Education at Bar-Ilan University, Ramat-Gan, Israel 52900. He is a clinical and educational psychologist and an expert on dynamic assessment of learning potential. He served as President-Elect of the International Association for Cognitive Education and Psychology (IACEP) during the years of 1997–1999 and President during the years of 1999–2001.

Ganit Eshel Kedmi (Ph.D.) is a researcher at the VR Lab, School of Education, Bar-Ilan University. She is an organizational consultant, expert in organizational change management.

The Key Competencies in Informatics and ICT viewed from Nussbaum's Ten Central Capabilities

Toshinori Saito
Professional Program of School Education
Japan Professional School of Education
t-saito@kyoiku-u.jp

Abstract: This article shows a discussion about the key competencies in informatics and ICT viewed from a philosophical foundation presented by Martha Nussbaum, which is known as 'ten central capabilities'. Firstly, the outline of 'The Capability Approach', which has been presented by Amartya Sen and Nussbaum as a theoretical framework of assessing the state of social welfare, will be explained. Secondly, the body of Nussbaum's ten central capabilities and the reason for being applied as the basis of discussion will be shown. Thirdly, the relationship between the concept of 'capability' and 'competency' is to be discussed. After that, the author's assumption of the key competencies in informatics and ICT led from the examination of Nussbaum's ten capabilities will be presented.

Keywords: Capability approach, competency, teaching informatics in general education, philosophical foundation of informatics pedagogy, education and public policy

1 Introduction

'Key competencies in informatics and ICT' seems to be a topic which tends to be discussed in the context of technology and social change. It is actually natural because the topic has emerged with the recognition of the impact of social change brought by the proliferation of digital networking technologies. However, the author considers that the focus of the discussion should not be restricted within the context of technology and social change because the concept of 'competency' has also the strong link toward other contexts, such as pedagogy, philosophy, and public policy. The attempt with which the author has engaged in this article is to build a ground of the discussion of the topic in terms of the universality of human well-being, come from the viewpoint of moral philosophy, which gives a starting point to consider current problems of informatics education and the educational use of ICT as subjects of public policies.

More specifically, this article attempts to show a discussion about the key competencies in informatics and ICT viewed from Martha Nussbaum's central capabilities. In this article we consider the following question: How should we delineate key competencies in informatics and ICT, which are acceptable from the viewpoint of social justice? The basis of the discussion is put upon the recognition that the execution of informatics and ICT education is a kind of the matter of public policy, which is necessarily grounded on a normative standard. Hence, we attempt to conduct the reasoning referring to The Capability Approach (CA), which is proposed by Amartya Sen and Martha Nussbaum, as a normative basis of public policies.

2 What is the Capability Approach?

The CA is a theoretical framework presented by Amartya Sen and Martha Nussbaum, which is intended to evaluate and assess the state of social welfare, namely, "individual well-being and social arrangements, the design of policies, and proposal about social change in society" (Robeyns, 2005, p. 94). This framework comprises two core terms, which are 'functionings' and 'capability'. The term 'functionings' is explained as that which "represent parts of the state of a person – in particular the various things that he or she manages to do or be in leading a life" (Sen, 1993, p. 31), such as "being happy, having self-respect, taking part in the life of the community, and so on" (Sen, 1992, p. 39). On the other hand, the term 'capability' is that which "reflects the alternative combinations of functionings the person can achieve" (Sen, 1993, p. 31). The CA is characterized by evaluating the equality of the enjoyment of "substantive freedom" (Sen, 1992, p. 49) to achieve functionings which a person thinks are valuable, whilst the utilitarian approach features merely the achievement of individual 'utility' which is defined in terms of some mental characteristics, such as pleasure, happiness, or desire, and ignores freedom and achievements which are other than those reflected in one of these mental metrics (ibid, p. 6).

3 What and why Nussbaum's ten Central Capabilities?

This article attempts to examine how the key competencies in informatics and ICT are illustrated from the viewpoint of 'ten central capabilities' (Figure 1) proposed by Nussbaum (Nussbaum, 2000, 2005, 2011b). 'Ten central capabilities' is a list of capabilities about which she argues are "fundamental entitlements inherent in the very idea of minimum social justice, or a life worthy of human dignity" (Nussbaum, 2011a, pp. 24–25). Though Sen denies to indicate or emphasize the importance of any specific capabilities positively in the CA, Nussbaum has presented a comprehensive view of indispensable capabilities as 'ten central capabilities', that she insists the government is responsible to ensure, which is based on "common humanity" (Nussbaum, 1993, p. 263) led from Aristotle's list of the sphere of the 'grounding experience' and virtues. This is because she believes the CA should have the potential of evaluating the achievement of creating just society, and for the purpose, "those human capabilities that can be convincingly argued to be of central importance in any human life" (Nussbaum, 2000, p. 75) have to be isolated.

The reason for applying 'ten central capabilities' as a viewpoint of illustrating the key competencies is that it is thought to be a reliable framework that leads justifiable objectives of public policies about human well-being which any just society should pursue. More precisely, we would point out three reasons that lead us to apply 'ten central capabilities'. Firstly, it is the framework which dares to bring the universal foundation upon the discussion of human well-being. Her notion of universalism, which comes from the interpretation of Aristotelian philosophy, is oriented toward cross-cultural consensus and to be distinguished from absolute view of culture and humanity in allowing pluralism and being open to necessary revisions (Nussbaum, 2000, p. 77). Secondly, it is presented essentially as a normative framework which intends to assess the moral basis of political issues. This shall give an obvious advantage for our attempt of seeking a moral basis and justifiable understanding of the key competencies which should be provided in terms of public policies. Thirdly, it is described clearly and explicitly, which enables it to be shared and discussed in public. We recognize that dealing with moral issues as a research topic is necessarily followed by the difficulty of proving its validity and has inevitable limitation in its result because the ultimate foundation of the research is always placed on intuitive judgement. Therefore it is crucial for our research to have an open and explicit theoretical ground which gives a starting point of our argument.

4 The Relationship between Competency and Capability

According to the preceding discussions, capability is understood as border in its scope of the meaning than competency. Stephenson (1998, p. 3) describes competency approach as "essentially a top-down control model which aims to secure the effective delivery of current services based on standards determined by past performance", and in addition, he explains as below:

> Capability is a broader concept than that of competence. Competence is primarily about the ability to perform effectively, concerned largely with the here and now. Capability embraces competence but is also forward-looking, concerned with the realization of potential (Stephenson, 1998, p. 3).

Otto and Ziegler (2006) have also referred to the relationship of both concepts in the context of education.

> Thus the strength of the capability approach lies in its capacity to provide sensible tools and frameworks within which literacy, competences and other educational aspects might be appropriately conceptualised and evaluated (Otto, Ziegler, 2006, p. 270).

Based on the discussion above, in this article we shall adopt the presumption that the concept of competency is to be contained by the concept of capability. More precisely, we regard competency as a factor which belongs to humans, that supports realization of capabilities through ensuring some functionings and enabling humans to have alternatives to choose concerning one's well-being. Another factor that would support realization of capabilities is assumed to be socio-cultural context, which affects human consciousness externally and prepares them to acquire some competencies through providing specific social and/or cultural preconditions. If the socio-cultural context allows a person to find and recognize the existence of the demand for a competency, and he/she actually succeeds in acquiring it, then we are able to tell that a functioning which supports a certain capability is prepared for being practiced.

1. **Life.** Being able to live to the end of a human life of normal length; not dying prematurely, or before one's life is so reduced as to be not worth living.
2. **Bodily Health.** Being able to have good health, including reproductive health; to be adequately nourished; to have adequate shelter.
3. **Bodily Integrity.** Being able to move freely from place to place; to be secure against violent assault, including sexual assault and domestic violence; having opportunities for sexual satisfaction and for choice in matters of reproduction.
4. **Senses, Imagination, and Thought.** Being able to use the senses, to imagine, think, and reason – and to do these things in a "truly human" way, a way informed and cultivated by an adequate education, including, but by no means limited to, literacy and basic mathematical and scientific training. Being able to use imagination and thought in connection with own choice, religious, literary, musically, and so fourth. Being able to use one's mind in ways protected by guarantees of freedom of expression with respect to both political and artistic speech, and freedom of religious exercise. Being able to have pleasurable experiences and to avoid nonbeneficial pain.
5. **Emotions.** Being able to have attachments to things and people outside ourselves; to love those who love and care for us, to grieve at their absence; in general, to love, to grieve, to experience longing, gratitude, and justified anger. Not having one's emotional development blighted by fear and anxiety. (Supporting this capability means supporting forms of human association that can be shown to be crucial in their development.)
6. **Practical Reason.** Being able to form a conception of the good and to engage in critical reflection about the planning of one's life. (This entails protection for the liberty of conscience and religious observance.)
7. **Affiliation. (A)** Being able to live with and toward others, to recognize and show concern for other human beings, to engage in various forms of social interaction; to be able to imagine the situation of another. (Protecting this capability means protecting institutions that constitute and nourish such forms of affiliation, and also protecting the freedom of assembly and political speech.) **(B)** Having the social bases of self-respect and nonhumiliation; being able to be treated as dignified being whose worth is equal to that of others. This entails provisions of nondiscrimination on the basis of race, sex, sexual orientation, ethnicity, caste, religion, national origin.
8. **Other Species.** Being able to live with concern for and in relation to animals, plants, and the world of nature.
9. **Play.** Being able to laugh, to play, to enjoy recreational activities.
10. **Control over One's Environment. (A) Political.** Being able to participate effectively in political choices that govern one's life; having the right of political participation, protections of free speech and association. **(B) Mental.** Being able to hold property (both land and movable goods), and having property rights on an equal basis with others; having the right to seek employment on an equal basis with others; having the freedom from unwarranted search and seizure. In work, being able to work as a human being, exercising practical reason and entering into meaningful relationships of mutual recognition with other workers.

Figure 1: Nussbaum's ten central capabilities (Nussbaum, 2011b, pp. 33–34)

5 Informatics and ICT Competencies viewed from the Central Capabilities

As a result of the discussions in this article, we shall present a summery of the assumption of the key competencies of informatics and ICT, which we found through the examination of Nussbaum's ten central capabilities. To acquire the summary, we examined each of the meanings of the ten capabilities in terms of the conditions indispensable for the achievement of each of the ten capabilities. Then we analysed those conditions and found out the possible informational factors, which were supposed to have a link with using information or the knowledge of informatics or ICT, which were thought to affect the formation of the conditions. Following that, we analyzed the detail of the informational factors and distinguished what was thought to belong to human competency from what was thought to belong to socio-cultural context. Then, we summarized the results about the human competencies which provides the basis of the informational conditions indispensable for the achievement of the ten capabilities and made eight descriptions listed below which represent the substance of the examination. The numbers within the parentheses after each of the sentences indicate the relating item(s) of the Nussbaum's ten capabilities.

1. **Accessing, collecting and understanding information.** Being able to access and collect the information about what helps improve quality of life and realize well-being: to understand and use available support known from the collected information. (1, 2, 3, 7, 8, 9, 10)
2. **Vision organizing.** Being able to organize a vision about human life and well-being based on the available information. (1, 2, 3, 6, 7, 8, 9, 10)
3. **Participation in communication.** Being able to participate in the communication which is concerned with the realization of well-being and, if necessary, to create a platform of communication with the available informational tools. (1, 2, 3, 5, 7, 8, 9, 10)
4. **Expansion of sense, imagination, and thought.** Being able to learn and expand the use of senses, imagination, thought, and reasoning using the help of informational environment: to express as and convert into information the fruit of the use of such abilities. (4)
5. **Relationship control with informational environment.** Being able to control the relation with informational environment in response to its influence upon one's emotion. (5)
6. **Application of informatics concepts toward practical reasoning.** Being able to use and apply the concepts of informatics as the basis

of practical reasoning, which is required by the nature of the informational environments consist of information and communication technologies. (6)

7. **Use of practical reasoning ability.** Being able to make the best use of the ability of practical reasoning helped by informational environment for the pursuit of conceptions of the good. (6)
8. **Management of property's information.** Being able to manage by self the information of one's own property assisted by informational environment. (10)

5.1 Accessing, collecting and understanding information

This competency is about the connection between the source of information and those who demand the information. In accordance with the listed capabilities (1, 2, 3, 7, 8, 9, 10) which give the ground for the competency, the instances of the information are assumed to be the following: the social conditions and the person's individual circumstances, the rights guaranteed by law, the social support services provided by the government or the other various formal/non-formal organizations, the procedure of political participation and the choices of the political groups to support, the circumstances of the labour market and the available support for job seeking, the basic knowledge about acquiring the goods and the properties indispensable for healthy and cultured human life.

This competency is intended to make the listed capabilities feasible in terms of improving the certainty of the receipt of information. The realization of the listed capabilities is thought to require two important conditions which are related to the receipt of information: firstly, people are able to have the idea of and the desire for well-being, secondly, the existence of the services which are relevant to the support of the realization of well-being is recognized by the targeted people. With regards to the former condition, the idea and the desire for well-being cannot be formed without any information about what human life is. For the latter condition, there would be no chance for the services to be used by the targeted people if the information about the existence of the services doesn't reach to them. In terms of the human competency, these are assumed to be the problems of the ability of accessing, collecting, and understanding information because these three factors are seem to be essential to ensure the certainty of the receipt of the information.

5.2 Vision organizing

This competency is about the conversion of information into visions of the future and we think this is what consists of the realization of the listed capabilities (1, 2, 3, 6, 7, 8, 9, 10) by providing the fundamentals of the desire for well-being. When we think of doing something for the sake of improving quality of life, there must be a vision, regardless of whether it is clear or not, about what a desirable life would be. If the vision is clear and long-term enough, it would well enlighten what and how we should engage in the improvement of the current quality of life. On the contrary, if we are merely allowed to have ambiguous and short-term visions, it would be quite difficult to figure out what the problem is in the current quality of life. If we are in the same condition about accessing and collecting needed information, one of the most influential factors which decides whether we could have a clear and long-term vision of well-being or not shall be the ability of interpreting the meaning of the information and forming an idea based on the acquired meaning. In terms of the informational competency, this ability is to be expanded as the capability of organizing a vision about well-being based on the available information.

5.3 Participation in communication

This competency expresses the fundamental condition which enables the social participation to maintain human quality of life. We think this competency directly links to the capability 7 (affiliation), and also has deep relationship with the other listed capabilities (1, 2, 3, 5, 8, 9, 10). The communication which is concerned with the realization of well-being may be understood quite broadly, such as communication concerning with daily life, culture and hobbies, jobs, political and economical issues, and so forth, all of which provide information that indicates explicitly or implicitly the state of individuals' and communities' well-being. This competency is crucial for the realization of the listed capabilities because of the following two reasons: Firstly, communication assisted by ICT (Information and Communication Technologies) is assumed to extend grounds for practical urging on society by individuals or groups for the sake of human quality of life: secondly, it is also to extend opportunity of individuals to acquire self-respect and human dignity through accepting consideration by others.

5.4 Expansion of sense, imagination, and thought

This competency indicates the importance of the potential to develop the basis of human intelligence and sensibility, making full use of the merits provided by given informational environment, which is assumed to be formed with available informational resources and the tools to use them. This competency gives an essential precondition for the capability 4 to be realized under the circumstance of digital informational environment, for we need to have the opportunity of learning and developing how to use our intelligence and sensibility in cooperation with the informational environment before we enjoy fully these abilities. In this context, informational environment shall take the role of providing directly the access to intellectual or cultural assets or systems of education, and also the role of providing the indirect information about how to access such assets or educations.

5.5 Relationship control with informational environment

This competency focuses on the power of mentality and intelligence to keep emotional autonomy from the influence of informational environment and is thought to have a strong link to the capability 5 through giving the ability of choose appropriately the distance with the informational environment. In the society where the use of digital network has become ordinary, the influence of informational environment upon human emotion is supposed to be crucial and unavoidable, so that control over the distance with informational environment would be essential to maintain good health of our emotion in the life of on-line and off-line.

5.6 Application of informatics concepts toward practical reasoning

This competency is about the concepts of informatics that would become indispensable knowledge to provide the basis of practical reasoning within the society where the use of digital informational environment has been diffused. ‘Practical reasoning’ is assumed to be the application of the reasoning ability toward practical problems which we encounter in daily life. On the other hand, the concepts of informatics, such as abstraction of data, algorithm construction, structuralization of knowledge, system analysis and integration, formalization and standardization of expression, and so on, are kinds of knowledge base which propose proper solutions for the problems of human and computer software. If we hope truly to have the capability of forming a conception of the good and engaging in critical reflection about the planning of one's life, according to the description of the capability 6, we must have good command of changing and reorganizing given informational environment following the requests that come from the conception of the good we pursue and the critical reflection we are engaging in. Therefore the concepts of informatics would become the common basis of practical reasoning under the circumstance of digital information environment.

5.7 Use of practical reasoning ability

This competency features the actuality of realization of conceptions of the good by using the ability of practical reasoning. We assume that the capability 6 would be supported not only by the knowledge of the informatics concepts, which is argued as the competency 6, but also by the actual ability to execute reasoning helped by informational environment. The ability of reasoning is supposed to be various and not limited within the informatics concepts, however, if we hope to enjoy fully the merits of digital informational environment for our reasoning practice, it is obvious that learning the way of reasoning using the artefacts of computing, which is known and practiced as computational thinking, must be helpful.

5.8 Management of property's information

This competency is about the economic independency and autonomy which are enhanced by informational environment. This competency is related to the former part of the capability 10 – (B), which is concerned with holding property. We consider the competency is essential because the information of the status of the property which is owned by individual is getting to be computerized recently and much easier to be recognized, and such recognition is believed to allow the individual to make use of the property with a careful and long-term plan, which means the economic liberty possessed by the individual has been increased relatively.

6 Conclusion

In this article we have discussed the key competencies of informatics and ICT from the viewpoint of Nussbaum's ten central capabilities. As a result, we have found eight descriptions of the assumption of the key competencies, which are concerned with (1) accessing, collecting and understanding information, (2) vision organizing, (3) participation in communication, (4) expansion of sense, imagination, and thought, (5) relationship control with informational environment, (6) application of informatics concepts toward practical reasoning, (7) use of practical reasoning ability, and (8) management of property's information. To draw the shown result, we have also examined the nature of Nussbaum's 'ten central capabilities', which is characterized by the connection toward Aristotelian philosophy and the standpoint of universalism, and the conceptual relationship between competency and capability.

The core motivation for us to engage in this research is to present a version of philosophical understanding of teaching informatics, and through this attempt, we hope to make clear the public and universal nature of teaching informatics as a genre of pedagogy. As we argued before (Saito, 2013), establishing the logic that justifies the position of teaching informatics as a public matter would be one of the keys to diffuse fair understanding of and share awareness with those outside the ICT-education-practitioner-communities about the indispensability of teaching informatics in general education, and this research has been conducted along such interest. We expect that the eight key competencies could be referred as a normative foundation of the purpose of teaching informatics in general education and the criteria which are used to assess and evaluate the achievement of it.

We recognize that the research still has some problems left to be solved. Firstly, the detail and the basis of the process of leading the eight key competencies from Nussbaum's ten capabilities are not shown sufficiently. To lead the eight competencies from the ten capabilities, we examined the general conditions which we thought were essential for the achievement of each of the capabilities, and then, we analyzed the relevance of the informational factors with those general conditions. Moreover, we distinguished the informational factors which we thought belong to human competency from those which belong to socio-cultural context. This process is based on the belief that the informational factors, which can be understood as combinations of the aspect of human competency and that of socio-cultural context, are necessarily to be crucial for the achievement of each of the ten capabilities. However, in this article we could not present clearly the discussion about the reason why we considered the belief to be proper and reliable as a basis of the reasoning process. Secondly, the examples which explain the application of each of the eight competencies to pedagogical occasions are also insufficient. Not until they are applied to indicating the purpose and the objectives of teaching informatics in general education would the presented eight key competencies be worthy of being referred as an instance of the knowledge which provides a version of philosophical understanding of teaching informatics. We recognize that concrete examples would help making clear the pedagogical meaning of the eight key competencies and without examples, on the other hand, the contribution of the eight key competencies toward informatics pedagogy would remain quite limited. Both of the problems shown here are surely to be addressed continuously in our following researches.

References

Nussbaum, M. C. (1993). Non-relative virtues: an Aristotelian approach. In Nussbaum, M. C., Sen, A. (Eds.), *The quality of life* (pp. 242–269). Oxford University Press.

Nussbaum, M. C. (2000). *Women and human development: The capabilities approach*. Cambridge University Press.

Nussbaum, M. C. (2005). *Frontiers of justice. Disability, nationality, species membership*. Harvard University Press.

Nussbaum, M. C. (2011a). Capabilities, Entitlements, Rights: Supplementation and Critique. *Journal of Human Development and Capabilities, 12*(1), pp. 23–37.

Nussbaum, M. C. (2011b). *Creating capabilities*. Harvard University Press.

Otto, H., Ziegler, H. (2006). Capabilities and Education. *Social Work & Society, 4*(2), pp. 269–287.

Robeyns, I. (2005). The Capability Approach: a theoretical survey. *Journal of Human Development*, 6(1), pp. 93–117.

Saito, T. (2013). An Examination of the Democratic Purposes of Informatics Education, In Reynolds, N., Webb, M., Syslo, M., Dagiene, V. (Eds.), *Learning While We Are Connected*, Paper presented at WCCE2013 10th IFIP World Conference on Computer in Education, Nicolaus Copernicus University, Toruń, 2–5 July (pp. 301–302).

Sen, A. (1992). *Inequality reexamined*. Oxford University Press.

Sen, A. (1993). Capability and well-being. In Nussbaum, M. C., Sen, A. (Eds.), *The quality of life* (pp. 30–54). Oxford University Press.

Stephenson, J. (1998). The Concept of Capability and its Importance in Higher Education. In Stephenson, J., Weil, S. (Eds.), *Capability and Quality in Higher Education* (pp. 1–13). London: Kogan Page. Retrieved from http://www.heacademy.ac.uk/

Biography

Toshinori Saito is an associate professor in the Professional Program of School Education at Japan Professional School of Education, also is a member of Information Processing Society of Japan, Japan Society for Educational Technology, Community for Innovation of Education and Learning through Computers and Communication Networks, and Asia-Pacific Media and Information Literacy Education Centre.

Considerations for the Design of Computing Curricula

Mary Webb
King's College London
Stamford Street, London, SE1 8WA UK.
mary.webb@kcl.ac.uk

Abstract: This paper originated from discussions about the need for important changes in the curriculum for Computing including two focus group meetings at IFIP conferences over the last two years. The paper examines how recent developments in curriculum, together with insights from curriculum thinking in other subject areas, especially mathematics and science, can inform curriculum design for Computing. The analysis presented in the paper provides insights into the complexity of curriculum design as well as identifying important constraints and considerations for the ongoing development of a vision and framework for a Computing curriculum.

Keywords: Curriculum, Computer Science, Informatics, curriculum theory

1 Introduction

Recent review of the ICT curriculum in the UK (The Royal Society, 2012) identified a need for major reform that recognises the value of Computer Science as an academic discipline. Similar calls have been made in the United States (Wilson, Sudol, Stephenson, Stehlik, 2010) and throughout Europe (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013). These initiatives emphasise refocusing Computing education to incorporate Computer Science as the underlying subject discipline. A major concern is that the curriculum has become unbalanced with too much focus on basic digital skills at the expense of deeper understanding of concepts. This has led to much debate about what should be included on Computing and/or ICT in the curriculum. In this paper I will examine this debate and consider the role and purpose of the Computing curriculum. In particular I will focus on the theoretical basis for the design of curricula for Computing: how will we decide the curriculum content and focus? What is its purpose? And what are the implications for the design of the structure and sequencing?

This paper builds on focus group meetings that took place at the IFIP Conference in Manchester and the World Conference on Computers in Education 2013 in Torun, Poland. These meetings highlighted that there are a range of views among professionals working in the area of Computer Science and ICT. A general agreement reached in these meetings was that in order to define a vision or framework, which may help to inform curriculum development, we need to define what is the range and scope of the subject and what are the key ideas and subject matter in the field(s) and at the same time explain why these are important for people to learn. In this way we can move towards a vision and rationale for the curriculum and perhaps a framework. In this paper I am not aiming to synthesise all of the debate from those meetings. Instead I aim to suggest some ways to take forward this debate and of moving towards a vision for future development of the curriculum relating to Computing/ICT. In order to do this I will examine recent debates in curriculum theory generally as well as other subject areas with relevance to Computing and consider possible implications for the Computing curriculum. First I will briefly explain the terminology and background including the debate about the Computing curriculum in the focus group discussions.

2 Background and Terminology

The variation in terminology has been a source of much confusion in relation to Computing/ICT. The Royal Society report (2012) provided some useful definitions based on the situation in the UK in 2012 (see Table 1) and these will form the basis for definitions in this paper with some further clarification as explained below.

Table 1: Computing in schools terminology (The Royal Society, 2012, p. 5)

Computing The broad subject area; roughly equivalent to what is called ICT in schools and IT in industry, as the term is generally used. **ICT** The school subject defined in the current National Curriculum. **Computer Science** The rigorous academic discipline, encompassing, programming languages, data structures, algorithms, etc.	**Information Technology** The use of computers, in industry, commerce, the arts and elsewhere, including aspects of IT systems architecture, human factors, project management, etc. (Note that this is narrower than the use in industry, which generally encompasses Computer Science as well.) **Digital literacy** The general ability to use computers. This will be written in lower case to emphasize that it is a set of skills rather than a subject in its own right.

While the school subject in the UK had come to be called ICT, in many other parts of Europe, a scientific discipline, known as Informatics (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013) had continued to be followed in some countries. Informatics is a broader term than Computer Science, for example the Joint Informatics Europe & ACM Europe Working Group on Informatics Education use the term Informatics to "cover the entire set of scientific concepts that make information technology possible" (2013, p. 9). In this paper the terminology shown in Table 2, which is largely based on the Royal Society Report, will be used. Thus the term ICT will be avoided in this paper as its meaning is subject to too much variation in interpretation and Computing will be used to designate the broad subject area.

Table 2: Terminology used in this paper

Information Technology (IT) – The use of computers, in industry, commerce, the arts and elsewhere, including aspects of IT systems architecture, human factors, project management etc. (Note that this is adopted from the Royal Society Report and is the title of courses in the UK at GCSE and A-level)
Computer Science – The rigorous academic discipline, encompassing programming languages, data structures, algorithms, etc.
Computing – The broad subject area. This is now the title for the new curriculum in the UK
Digital literacy – The general ability to use computers

The focus group meetings at the IFIP conferences in Manchester and Torun aimed to debate the issues with a view to moving towards a consensus about a vision for the curriculum and how to develop a framework for the design of a curriculum for Computing/digital literacy. The debate was enthusiastic and quite wide-ranging and suggested that reaching a clear consensus and way forward was likely to be difficult. Table 3 summarises key ideas that arose together with an estimate of the level of consensus amongst the participants. The range of views and lack of consensus indicated in Table 3 as well as in other debates about the curriculum for Computing/digital literacy indicate the extent of the challenge of designing a curriculum framework and suggest a need to identify a possible theoretical basis for curriculum design in this area.

3 A theoretical Basis for Curriculum Design

An obvious starting point for a theoretical basis for designing the Computing/digital literacy curriculum is curriculum theory. However curriculum theorists have identified a crisis in their field (Priestley, 2011; Young, 2013). Priestley argued that the crisis is due to new uncertainties which require new approaches to practice and new ways of thinking (Priestley, 2011). This, Priestley argues, is evident in the emergence of new models of national curricula around the world characterised by outcomes sequenced into linear levels and an emerging focus on generic skills (ibid.). Curricula have been developed that are a-theoretical and instead aim to meet the needs of learners using a pragmatic approach with inherent contradictions and lack of conceptual clarity (Priestley, Humes, 2010). Young attributes the crisis to the neglect of the role of access to knowledge in current curriculum theory (Young, 2013). Biesta (2014), on the other hand, questions whether knowledge can be viewed in isolation from other considerations such as critical judgement. Biesta argues, based on Dewey's work, that knowledge is a construction in "transaction" (interactions taking place in nature) which means that knowledge is both constructed and real. In Dewey's view as examined by Biesta, in order to get knowledge we need action. Within this view of knowledge, deciding curriculum content is a matter of coordination between individual learners and social factors. Therefore, Biesta argues, when designing curricula, we should proceed pragmatically in a careful and precise way in relation to matters of human concern. Thus significant philosophical differences are evident in curriculum theory.

This crisis is not just a recent phenomenon but rather a series of cycles of reconceptualisations that started in the 1970s (Pacheco, 2012). Curriculum studies and curriculum theory are complex and many factors have been important in recent changes and recent thinking including philosophical and epistemological considerations, internationalisation, and politicisation. The precise nature of the current crisis or cycle is beyond the scope of this paper – see Pacheco (2012) for an overview of the state of the curriculum studies field. What is clear is that there is no one theory of curriculum or curriculum design that is commonly accepted and will provide us with the means of establishing a curriculum vision and framework. What we can take from the various debates about curriculum theory are a series of questions and issues applicable to curriculum design in general which can be examined in relation to current thinking about Computing/digital literacy curricula and specific examples of curricula or curriculum frameworks. Furthermore various constraints on curriculum design have been identified particularly from epistemological considerations (Winch, 2013; Young, 2013) and we might consider how they should constrain curriculum design for Computing/digital literacy.

Table 3: Views that emerged from the WCCE panel discussion (Torun 2013)

Key idea/question about Computing curricula	**Level of consensus**
Computer Science and digital literacy are complementary – both are needed in the school curriculum	High
Need room for flexibility in interpretation	High
What is the importance of Computer Science for general education? – This is important	High consensus that this question is important
Problems of defining terms	Consensus that terminology is important and difficult
We need to develop aware citizens – not necessarily creators but more than consumers	Controversial
Teaching children to be aware, not necessarily how to create from scratch	Controversial
Current trend is a grass roots movement that appears to have joined forces and coordinated. At the heart of it is an understanding that Computing is essential for all children but also a need for opportunities for career paths and citizenship	Fairly high
A set of concepts based on Computer Science should be defined as a basis for the curriculum – some concepts have a long shelflife whereas others are short-lived	Fairly high
Computer Science is for everyone	Controversial
What are the good practices that are working?	Controversy over whether this is an important question or not?
Towards a curriculum framework: When – from the beginning What – clear examples How – basic principles Who – concerns with teacher training	The key principles of what needs to be decided or agreed.

The questions include: What is learners' entitlement? What is the nature of knowledge in relation to the curriculum? What is the relationship between theory and practice? How detailed should curriculum specifications be?

4 Learners‘ Entitlement to Knowledge

It is self-evidently obvious that curriculum design should pay attention to learners’ entitlement. For Young (2013) resolution to the crisis in curriculum theory should be through a “knowledge based” approach which starts from the learner’s entitlement to knowledge. He argues that there are two important models for curriculum design which both need to be harnessed. The first model, which inherits a view of the curriculum as a source of the sacred, puts trust in knowledge and in teachers as pedagogic authorities whereas the second put its trust in the emancipatory capacities of learners (ibid). According to Young, curriculum theory must take, from the sacred tradition, both: 1) the idea of a store of knowledge and 2) human values of inwardness and dedication that shape and are associated with disciplined study and enquiry. At the same time, according to Young, in order to harness the emancipatory capacities of learners, the curriculum should take them beyond their own experience so curriculum design should start from the learner’s entitlement to knowledge. Thus the goal of the curriculum becomes to define its content in a world in which the entitlement to knowledge is the goal. In this endeavour “powerful knowledge” is key, defined as specialised discipline-based knowledge which is different from the experience-based knowledge that pupils bring to school (Young, 2013). The next step in this paper is to examine epistemological considerations in more depth in relation to knowledge in curricula.

5 The Nature of Knowledge

A philosophical issue influencing both curriculum change and pedagogy is the changing nature of knowledge in the knowledge society in which a view of knowledge as a fixed body is giving way to something with verb-like characteristics which is rapidly changing and developing through networked interactions (see for example Hipkins, Reid, Bull, 2010 for a review of recent thinking). In the knowledge society the need for people to be involved in continual knowledge creation through interaction in social cultural settings is in conflict with educational processes which build young people’s knowledge through a predetermined sequence of learning (Bauman, 2005). These knowledge creating processes are undeniably important in most areas of endeavour but the extent to which knowledge creation should determine the curriculum in primary and secondary schools is a matter of debate. The concept of powerful knowledge (Young, 2013) presents a way of thinking about knowledge where, even in this changing scene, some types of knowledge are more important for curriculum design than others. Young's explanation of "powerful knowledge" resides in the knowledge associated with academic subjects which he argues is specialised and differentiated by the boundaries between school and everyday knowledge. Clearly, as Young argues, this knowledge is not fixed nor is it equally easily identifiable across all subjects but in each discipline there are people committed to creating and evaluating some kind of knowledge base. Whether or not we believe that powerful knowledge should be the key ingredient of a Computing/digital literacy curriculum, reviewing what such powerful knowledge might be will at least give us an insight into how such thinking plays out in our discipline.

Epistemologists generally recognise three kinds of knowledge: propositional knowledge, know-how and knowledge by acquaintance (Winch, 2013). In examining how these forms of knowledge are relevant for curriculum design, Winch utilises Paul Hirst's characterisation of forms of knowledge into propositional knowledge, conceptual structures and methods of investigation (Winch, 2013). In this characterisation, forms of knowledge can be distinguished through variation across all three of these dimensions. According to Winch a key insight of the Hirstian classification is the close relationship between propositional and practical knowledge as well as the close inter-dependence of propositions. These propositions, through the mediation of concepts, particular to the subject, as well as more general concepts, form the basis of under-standing through the mastery of inferential relationships. Thus attention focuses on 1) learning as concept formation and 2) on practical procedures for managing knowledge. From this view curriculum design is about the management of growth of expertise within a subject which recognises different kinds of knowledge and their interrelationships. Winch argues that gaining a coherent view of this "epistemic ascent" within a subject is a key element in curriculum design. Therefore a major issue in curriculum design is to obtain sound grounds for the construction of schemata of epistemic ascent that are at least conceptually and normatively sustainable even if they are not yet empirically ratified.

The discipline of Computer Science encompasses foundational principles, widely applicable ideas and concepts as well as techniques and methods for solving problems and advancing knowledge as well as a distinct way of thinking and working (The Royal Society, 2012). Thus according to Young's (2013) definition Computer Science provides all or part of a powerful knowledge base which would be learners' entitlement. In this description provided by the Royal Society (2012) the three types of knowledge described by Winch (2013) are evident as is the importance of concepts and of practical procedures for managing knowledge. Key concepts identified by the Royal Society Report were programs, algorithms, data structures, architecture and communication (The Royal Society, 2012). The Joint Informatics Europe & ACM Europe Working Group identified similar concepts but theirs, they explained, were just examples from a much longer list. Therefore there remains a task, perhaps for IFIP TC3, to consider a complete high-level list of concepts for the curriculum. The techniques and methods that the Royal Society Report identified were modelling, decomposition, generalising with algorithms or data, designing, writing, testing, explaining and debugging programs (The Royal Society, 2012). Again the Joint Informatics Europe & ACM Europe Working Group identified similar techniques and methods but they also identified the importance of various intellectual practices such as tolerance for ambiguity (Joint Informatics Europe & ACM Europe Working Group on Informatics Education, 2013). Thus we are seeing consensus emerging from these working groups about the key concepts and techniques of the discipline although perhaps not yet agreement about the importance of more general intellectual practices. However, Computing as a practical subject, as well as theoretical raises other more complex issues concerning the relationship between theory and practice (Schwab, 1971; Winch, 2013) that will be discussed in the next section.

6 The Relationship between Theory and Practice

As Schwab (Schwab, 1971) argued, the difficulties in reconciling theory and practice within the curriculum are associated with the fundamental differences between them in that practical is concrete and particular whereas theory is general and economic in its specification. Thus in dealing with a practical problem in any discipline it is necessary to take account of a range of conditions which may not be addressed by the theory. Arguably, in learning Computing, the practical is more critical than in some other subjects with practical elements such as science where the practicals, at least at the level of primary and secondary science, are predominantly for the purposes of motivation and pedagogy rather than for developing the practical techniques per se (Abrahams, Reiss, 2012). In Computing, it is not only Computing professionals who need to develop practical skills in Computing, but skills and processes such as programming and computational thinking are needed both for personal productivity and across a range of other professions. Thus practical work is essential and may serve a variety of purposes.

Schwab's solution (Schwab, 1971) to the dilemma of the complex relationship between theory and practice is to employ a cyclical process whose purpose is to link theory and practice by mastering two or more theoretical viewpoints and their practical application, one at a time avoiding comparisons, and then once they have been mastered on their own terms to compare and contrast them. Each cycle contains two stages: the first of which is to master the theoretical viewpoint and the second to apply it to a series of cases. In brief, Schwab's rationale for this approach is the need for understanding plurality and the tendency for students, if faced with a new viewpoint before they have assimilated the first, to assimilate the new doctrine only in terms of the first. It is beyond the scope of this paper to develop this idea in depth in relation to Computer Science but I suggest that it has implications for the learning of programming, in particular the range of programming paradigms to be included in the curriculum as well as the mastery of specific programming languages. This approach has implications both for the range of content of the curriculum and for its structure and organisation.

7 Learners' Entitlement – A Curriculum for all

Young's argument, outlined above, is that the curriculum question: what knowledge? is primarily an epistemological one about what should constitute students' entitlement, together with identification of the epistemological constraints on structuring knowledge from the discipline into sequences suitable for different developmental stages (Young, 2013).

So far I have discussed the nature of knowledge and the importance of practical as well as theoretical knowledge. Learners' entitlement implies entitlement for all and therefore we need to consider a more controversial question: do *all* students need to understand the powerful knowledge in Computing that we have begun to identify? Such a question is rarely asked of other major discipline areas such as maths, English, science, history etc. because in most countries their place in the curriculum is assumed. Therefore the discussion of the curriculum in these traditional subjects focuses on the extent and range of the subject and the rationale. However Computing being a young discipline, its very existence in a curriculum needs to be justified carefully. There are many who have argued that the best basis for studying Computing at higher levels is a grounding in mathematics and natural science and that therefore strong foundations in Computer Science are not necessary in compulsory schooling. Arguments against such a view and in favour of Computing being part of the compulsory curriculum are varied and include the need for careful and systematic development of the principles and processes of Computer Science in order to avoid the ad hoc development of bad habits as well as the need to support creativity and problem solving more generally. There are three particularly compelling arguments for the Computing curriculum in compulsory education. First if learners are never introduced to Computing as a disciplinary area and to the knowledgebase and approaches that Computing academics and professionals use then they will not be able to determine whether this is for them. This therefore is an entitlement issue. Second, as many in the profession have argued, programming is difficult and it takes many years to learn to program. While programming is only one element of Computer Science, it is an essential element and it is inconceivable that an introductory course in Computer Science would not contain programming. Furthermore, while Computing professionals do not necessarily do the programming themselves, they need to understand essentials of programming in order to undertake a career in Computing. There is a view among Computer Science educators that coming to programming late in students' development is disadvantageous and that if they were to learn some of the techniques, approaches and thinking involved in programming at an earlier stage more of them would be successful. This therefore is both an entitlement issue for individuals looking towards a fulfilling, creative and potentially lucrative career as well as of concern to countries in terms of their economic performance and prosperity. The third argument is based on the ubiquitousness of Computing: since so much of our lives is dependent upon Computing we need to develop the understanding and skills of Computing necessary to participate in society. Both the Royal Society Report (2012) and the Joint Informatics Europe & ACM Europe Working Group (2013) emphasise individual entitlement, effects on economic prosperity and social aspects in their arguments for redeveloping Computer Science education.

The first two arguments outlined above are primarily based on the vocational rationale of enabling individuals to fulfil their ambitions of careers in Computing if they wish and also providing the workforce that will support the country's economy and its place in the world. While such justifications are often used and are sufficient to support the existence of a Computing curriculum there are many other roles which this curriculum could encompass including: the development of computational thinking as a basic literacy that is important to everyone in everyday life (Wing, 2006); developing the ability to solve problems with computers which is needed by a wide range of professionals; understanding the advantages and limitations of computer technology in order to make informed decisions about technology futures. In the light of this range of different roles that the Computing curriculum could support it is important to consider whether or not a broader rationale is achievable or would result in too many conflicting priorities. Linked to this question is the place of digital literacy and whether or not a Computing curriculum should incorporate all or some of the elements of digital literacy or whether digital literacy is an entirely cross curricular element. If we take Mioduser et al.'s explanation of the literacies associated with technologies as both affording and demanding their evolvement then there are seven such literacies: multimodal information processing, navigating the infospace, interpersonal communication, visualliteracy, hyper-literacy (hyperacy), personal information management (PIM), and coping with complexity (Mioduser, Nachmias, Forkosh-Baruch, 2008). Even a brief consideration of some of these literacies suggests that they require theoretical understanding in addition to practical skills. Furthermore while this theoretical understanding is broader than Computing there are some elements of understanding of Computing that would support most of these.

Other disciplinary areas have addressed these issues of how to deal with multiple, potentially conflicting, rationales, notably science education which, in Europe at least, has recently emphasised active, participatory approaches and a focus on contemporary societal issues in the earlier stages of compulsory education moving towards consideration of the nature of science and scientific method in upper secondary education (Eurydice, 2011). In maths education, where applied mathematics shares some characteristics with Computing, a debate has started about the needs of "constructors", "operators" and "consumers" (Skovsmose, 2004). We can see parallels here with the debate about the Computing curriculum in which the call for enabling students to be technology designers and creators not just consumers (The Royal Society, 2012) has echoed across recent debates in Computer Science education. Skovsmose's classification of those who practice mathematics is based primarily on an economic and vocational perspective. Thus constructors are those who maintain and further develop knowledge and techniques incorporating mathematics across a range of disciplinary areas; operators use mathematics only as part of the tools and instruments which they operate and consumers are those who only use mathematics in their daily life in order to interpret information such as tax returns, discounts etc. For Skovsmose the implication of the broadening of mathematics education, so that it can no longer rely for its position in the curriculum entirely on its intrinsic value but rather depends on its role as preparation for a range of social practices, is the need to live with uncertainty. In such uncertainty there is no foundation from which to build a strategy and instead it is necessary to live with uncertainty. From this perspective the only option is Biesta's (2014) pragmatic approach of proceeding thoughtfully with careful attention to matters of human concern. A next step is to examine constraints which might guide such a process.

8 Constraints

Important epistemological constraints on curriculum design were identified by Winch's (2013) analysis that focused on expertise and the idea that both subject knowledge and the growth of expertise in practical subjects require "epistemic ascent" through mastery of different kinds of practical ability. Moreover Winch argues the need to explore the constraints that the conceptual structure of the subject might impose on pedagogically and cognitively coherent schemata of epistemic ascent and then explore the implications of such constraints within conceptualisations of the subject. The constraints identified by Winch include three interrelated issues. First, it is necessary early in a curriculum (e.g. at primary level) to introduce all three major types of knowledge. This is because knowledge of individual propositions implies some understanding of the concepts that such propositions express and this in turn implies a significant ability to understand and make inferences within the subject. This is Knowing How to do something. Second there is a need for a structured approach to progression in learning the basic facts and central concepts of the subject because knowledge is systematic in terms of 1) classification of its various conceptual elements; 2) the relationships between the elements and 3) the procedures required to gain and validate knowledge. Third the kind of knowledge required to expand and manage subject matter requires a profound understanding of the subject including all of these interacting knowledge types. This therefore is not accessible to school students but comes in more advanced studies beyond school. The fourth constraint follows from the third and requires that the relationship between the ways in which pupils learn by simulating procedures for the acquisition of knowledge in their learning and the actual processes of expansion of disciplinary knowledge should be clarified. For example, project work in Computing often involves the systems development life cycle. Winch argues that simulating such procedures may be pedagogically important in developing acquaintance with the knowledge set of the subject as well as building understanding of techniques used in knowledge management. However these simulations should not be seen as simplified versions of expert practice as that might propagate an illusion that high-level design and planning activities are generic and can be used free of the reality of the skills and materials that are needed to execute the plan. Instead it should be recognised that such expertise requires extensive knowledge and is therefore only possible in higher level courses that build upon previous structured development.

9 How detailed should Curriculum Specification be?

The level of detail for the curriculum specification depends on its purpose and the view of knowledge which it embeds. Current curricular specifications are very variable, ranging from the 2-page spread of the UK National Curriculum for Computing to many pages of detailed specification. Some recent specifications have featured structured sequences of outcomes and a focus on generic skills or capacities rather than detailed specification of knowledge (Priestley, 2011). The use of these structured sequences, often described as competences, raises two important issues for curriculum design. First the understanding of competences varies in different countries (Gordon et al., 2009): for some competences are overall capacities in relation to a broad occupational field. Typically these competence frameworks incorporate a range of knowledge and skills as well as personal qualities. In other countries, such as the UK, competencies are generally defined as the ability to perform prescribed tasks to a certain standard. While these different definitions are sometimes distinguished by the use of the terms competencies and competences with and without an 'i', these distinctions are not universally adhered to. The second issue concerns how competences are derived. The two possibilities are: 1) by a rational analysis of the subject domain or area of expertise and 2) from the vocational demands of an area of work. Both are potentially problematic when changes are rapid as with Computing, where detailed specification may lead to stagnation unless the curriculum is kept under review.

10 Discussion and Conclusion

The brief journey into curriculum theory discussed in this paper has highlighted issues contributing to complexity as well as tensions and constraints in relation to designing a Computing curriculum. This explains, at least in part, the reasons for the challenges identified in previous focus group discussions. The lessons from curriculum theory and from experiences of curriculum design in other subjects suggest that we need to live with uncertainty and to accept the need for a dynamic and continually renegotiated curriculum. However there are epistemological considerations and constraints which can guide curriculum design (Winch, 2013; Young, 2013) even if it is necessary to take a primarily pragmatic approach as advised by Biesta (2014). A key consideration obviously should be learners' entitlement (Young, 2013) and identifying the knowledge that might constitute at least part of this entitlement must be an important part of the endeavours of those in the discipline that understand the current state of disciplinary knowledge. While the identification of such knowledge is no doubt made more complex by the changing nature of knowledge brought about by the knowledge society, it is clear that in computer science, some knowledge is sufficiently stable to be classified as powerful knowledge. At the same time the changing nature of knowledge and ways in which expertise can be developed are important issues in the 21st century which learners need to gain access to. In a previous paper, primarily focused on pedagogy (Webb, 2012), I argued that pedagogy needs a balance between learner-led and learning-led approaches which incorporate respectively development of specific expertise in limited areas focusing on learners interests together with curriculum-based learning led by specialist teachers that provides entitlement to knowledge.

Recent curriculum design for computing suggests that a consensus is emerging with respect to powerful knowledge which includes the key concepts of the discipline of computer science and of the techniques and methods. There is not yet clear agreement about the importance of various intellectual practices such as tolerance for ambiguity, which are broader than computing. Based on the analysis presented in this paper I suggest that the ongoing task of defining a vision and framework for Computing curricula requires the following interrelated and iterative activities:

- Continuing to identify and developing a consensus about key conceptual structures, propositional knowledge and methods.
- Deciding which knowledge elements are important for the varying roles of curricular identified as:
    - learners' entitlement in relation to future careers/employment;
    - personal productivity and the use of technologies for learning, social participation and leisure;
    - economic prosperity and future development at both country and global levels;
    - participation as informed citizens in decisions about technology futures e.g. in the role of robots and development of robotics.
- Deciding sequences/schemata for mastering knowledge that are conceptually and normatively sustainable bearing in mind the epistemological constraints identified in this paper. This process includes deciding on the level of detail that might be appropriate given the advantages and disadvantages of over specification discussed earlier.
- Identifying developmental constraints and pedagogical considerations including motivational aspects such as creativity and context.

References

Abrahams, I., Reiss, M. J. (2012). Practical work: Its effectiveness in primary and secondary schools in England. *Journal of Research in Science Teaching, 49*(8), 1035–1055. doi: 10.1002/tea.21036

Bauman, Z. (2005). Education in Liquid Modernity. *Review of Education, Pedagogy, and Cultural Studies, 27*(4), 303–317. doi: 10.1080/10714410500338873

Biesta, G. (2014). Pragmatising the curriculum: bringing knowledge back into the curriculum conversation, but via pragmatism. *The Curriculum Journal*, 1–21. doi: 10.1080/09585176.2013.874954

Eurydice, A. (2011). *Science Education in Europe: National Policies, Practices and Research*.

Gordon, J., Halász, G., Krawczyk, M., Leney, T., Michel, A., Pepper, D., … Wiśniewski, J. (2009). Key Competencies in Europe: Opening Doors For Lifelong Learners Across the School Curriculum and Teacher Education *CASE Network Reports* (Vol. 37). Warsaw, Poland: CASE-Center for Social and Economic Research on behalf of CASE Network.

Hipkins, R., Reid, A., Bull, A. (2010). Some reflections on the philosophical and pedagogical challenges of transforming education. *Curriculum Journal, 21*(1), 109–118. doi: 10.1080/09585170903560881

Joint Informatics Europe & ACM Europe Working Group on Informatics Education. (2013). *Informatics education: Europe cannot afford to miss the boat: Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education*.

Mioduser, D., Nachmias, R., Forkosh-Baruch, A. (2008). New literacies for the knowledge society. In J. Voogt, G. Knezek (Eds.), *International Handbook of Information Technology in Primary and Secondary Education*. London: Springer.

Pacheco, J. A. (2012). Curriculum Studies: What is The Field Today? *Journal of the American Association for the Advancement of Curriculum Studies*, 8.

Priestley, M. (2011). Whatever happened to curriculum theory? Critical realism and curriculum change. *Pedagogy, Culture & Society, 19*(2), 221–237. doi: 10.1080/14681366.2011.582258

Priestley, M., Humes, W. (2010). The development of Scotland's Curriculum for Excellence: amnesia and déjà vu. *Oxford Review of Education, 36*(3), 345–361. doi: 10.1080/03054980903518951

Schwab, J. J. (1971). The Practical: Arts of Eclectic. *The School Review, 79*(4), 493–542. doi: 10.2307/1084342

Skovsmose, O. (2004). *Research, Practice and Responsibility*. Aalborg: Institut for Uddannelse, Læring og Filosofi, Aalborg Universitet.

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*. London: The Royal Society.

Webb, M. (2012). Pedagogy with information and communications technologies in transition. *Education and Information Technologies*, 1–20. doi: 10.1007/s10639-012-9216-x

Wilson, C., Sudol, L. A., Stephenson, C., Stehlik, M. (2010). *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. Association for Computing Machinery (ACM), Computer Science Teachers Association (CSTA).

Winch, C. (2013). Curriculum Design and Epistemic Ascent. *Journal of Philosophy of Education, 47*(1), 128–146. doi: 10.1111/1467-9752.12006

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–36.

Young, M. (2013). Overcoming the crisis in curriculum theory: a knowledgebased approach. *Journal of Curriculum Studies, 45*(2), 101–118.

Acknowledgements

I would like to thank all those who contributed to the focus group discussions at the IFIP Conference in Manchester and the World Conference on Computers in Education 2013 (WCCE 2013) in Torun, Poland. While the analysis and views presented in this paper are entirely my own, they were informed by the debate during these meetings. My hope is that this paper will make some contribution to ongoing discussion which might lead to an emerging consensus for vision and framework for Computing curriculum. The key presenters at WCCE 2013 are listed below.
Monique Grandbastien, France; Alain Senteni, UAE and Quebec; Mary Webb, England; Nicholas Reynolds, Australia; Andrej Brodnik, Slovenia; Ivan Kalas, Slovak Republic; Bruria Haberman Israel; Harriet Taylor, USA

Biography

Mary Webb is Senior Lecturer in Information Technology in Education at King's College London and Chair of IFIP Working Group 3.3 on Research. Mary has developed and researched computing and the use of digital technologies in learning and teaching since computers first appeared in schools. She has taught Computer Science in secondary schools and runs a PGCE Programme for new Computer Science teachers.

How Things Work – Recognizing and Describing Functionality

Michael Weigend
Institut für Didaktik der Mathematik und der Informatik
University of Münster
Fliednerstr. 21
48149 Münster, Germany
michael.weigend@uni-muenster.de

Abstract: Recognizing and defining functionality is a key competence adopted in all kinds of programming projects. This study investigates how far students without specific informatics training are able to identify and verbalize functions and parameters. It presents observations from classroom activities on functional modeling in high school chemistry lessons with altogether 154 students. Finally it discusses the potential of functional modelling to improve the comprehension of scientific content.

Keywords: Function, programming, parameter, competence, abstraction

1 Recognizing Functionality

In mathematics a function is a mapping, a relation between a set A of inputs and a set B of outputs. Each input from set A is related to exactly one output from B.

Computer scientists consider functions as a programming construct that is used to cope with complexity. A complex operation serving a certain purpose is divided in less complex operations, which are easier to implement by program text. This is a method of structural decomposition and a fundamental idea of computer science (Schwill, 1994).

All higher programming languages like Java or Python support defining functions. Technically a function definition consists of a function name, parameters and a block of instructions, defining in what way the parameters are processed in order to produce a result which (in many cases) is returned to the calling process in a special return statement.

There are functions that do not return any object explicitly (procedures) but change the state of a mutable object. For example a list may be extended by appending a new element. On a higher level of abstraction one might say that the old state of the mutable object is the input and the new state is the output. Functions may be stand-alone objects. In object-oriented programming they are connected to classes or instances of classes (class methods and methods). An object represents some holistic entity from real life or fantasy. The methods are related to the general meaning of the object. For example in Python the class list represents mutable sequences of items. The methods of list objects represent meaningful facets of the holistic concept of a list like inserting, removing, changing or appending items.

Let me now shortly discuss four properties of functions that are relevant for learners.

1.1 A function is different of structure

According to Kroes (1997) all technical artifacts have a structure and a function, which "has a meaning only in the context of intentional human action" (p. 291). The function of a clock is to tell the time. We need this to manage our lives. Its structure is its physical implementation by electronic components, power supply etc. The dichotomy of structure and function is also adopted in biology to describe natural systems. In physiology the human body is seen as an aggregate of organs, which have certain functions in relation to other organs. For example the function of the heart is to pump blood. A function in a computer program is not a physical but a digital artefact. Its dual nature is given a) by its purpose (the desired effect) and b) by its implementation consisting of a block of program statements. When a programmer decomposes a complex task into less complex subtasks by defining functions or a class structure, she or he focuses on functionality and ignores the implementation of the functions or classes. They are considered as black boxes.

1.2 Functions are abstractions

One and the same function can be used to describe different activities. Lakoff and Nunez (1997) discuss conceptual metaphors for arithmetic operations that are used in math education. For example the addition 4 + 3 can be represented by putting together a collection of four beads and a collection of three beads ("arithmetics is collecting objects"). Another metaphor for the same operation is walking four steps and then walking another three steps in the same direction ("arithmetics is walking along a line"). Creating a function is just the other way round. It is finding an abstraction of activities like putting together collections of beads and walking certain distances. (In this example it is inventing addition.) When a programmer creates a new function within a software project, she or he tries to create an abstraction that can be called several times. This strategy leads to an efficient development process.

Abstraction takes place when (existing) functions or operators are overloaded. For example, one can apply the concept of "adding" to different domains

- Numbers: 2 + 2 == 4
- Sequences : [1, 2] + [2, 3] == [1, 2, 3, 4]
- Colors: red + green = yellow.

Addition is an arithmetic operation, but in the context of sequences (strings, lists etc.) adding means concatenation. In physics, "adding colors" refers to mixing light of different colors (additive colors).

Technically, overloading means to reuse an already existing operator (like +) or function name (like len) for a new activity. In Python the name __add__ corresponds to the operator +. When you want to define an addition for objects of class C you define a method named __add__ within the class definition of C.

1.3 Functions represent holistic concepts

To be of help in a modelling process a function must represent a single holistic idea of activity. Most functions are labelled by one verb: to add, to append, to destroy. It is good style in computer programming to use meaningful names for all kind of objects. It is recommended that a function name should be a verb. Regarding the mental representation of a function a meaningful name is more than good style but essential. According to Baddeley (2003) humans can only handle a few chunks of information in working memory at the same time. A function call can be regarded as such a chunk. If the idea of a function is not fully understood and clear it must be rehearsed first before it can be used for problem solving. It is for instance impossible to create or to understand an algorithm based on adding numbers and extending lists, when the meaning of these operations is not perfectly clear. A reason for overloading an operator like + is that it represents a gestalt-like concept that is already familiar. It is easier to extend this to a new domain than to create something new.

1.4 Functions may have parameters

There exist functions without any parameters. Each constant object can be considered as a zero-ary function. But these are special cases. There are two prominent intuitions visualizing the idea of a function: factory and tool (see Weigend, 2007). The factory-model is a black box with an entrance for input data and an exit through which produced output data leave. The tool model visualizes the function as a tool (e.g. a knife) that is able to modify a mutable object (e.g. cut off something). However, the objects which are processed by the function are specified by parameters. A function call (like a metaphor) implies a transfer of knowledge from one domain to another. And parameters with meaningful names can support this cognitive operation. Parameters are used in a function call (as arguments) and represent objects from domain A, where the function is used. Corresponding parameters appear also in the definition of the function (formal parameters) and represent objects within the domain B of the function definition. Consider this simple function, which calculates the area of a rectangle (Python):

```
def area (length, width):
return length*width
```

The parameters represent objects from the domain *geometry*. Imagine to use this function for calculating the area of a rectangular door, which is appropriate for humans. The function call (with position arguments) may look like this:

```
area(height + 10, armspan)
```

The parameters are related to the physical properties of a human. Thus they are from a different domain: biology. The transition from one domain to another can be made more explicit by using keyword arguments (Python):

```
area(length=height+10, width=armspan)
```

Each keyword argument *key=value* includes a mapping from an item of domain A to an item domain B.

2 Modeling with Functions as a Competence

The four properties of functions discussed in the previous section, correspond to cognitive operations that a programmer has to perform in some way, when she or he creates functions or classes of objects in order to model a scenario.

- *Abstraction.* The programmer must find similar activities within the scenario, which can be modeled by the same function. This implies the ability to use only functional aspects (not structural) as criteria for classification.
- *Conceptualizing.* The programmer must find a concept that describes all activities of a category on a more abstract level. She or he has to find a meaningful name that labels the concept.
- *Parameterization.* The programmer needs to identify parameters, i.e. objects that are taken as input and processed by the function.

What kind of cognitive operations are performed, when a programmer uses functions that already exist in the repertoire of the programming language?

- *Deductive reasoning*. When a programmer browses through class libraries looking for an appropriate function or class she or he has to understand functionality described in the documentation and apply it to a new context.
- *Transfer of knowledge.* In programming literature functions belong to a context like a class or a library. For example, in Python 3.3 instances of the built-in class list have 33 methods (22 of them are overloaded operators and functions). All these functions are related to the concept of a linear sequence of objects, which could be pictured by – say a row of ten boxes. Imagine Jenny using a list to model a collection of airports. When she uses the function len() to calculate the number of airports, she transfers the term *length* from the image of a sequence of objects in a row (which has a certain length) to a new domain. Airports are not boxes laying in a row. The term *length* is now metaphorical.

According to Schwill (1994), fundamental ideas (like decomposition) can be explained and understood on a low level without specific computer science (CS) knowledge. This is an implication of the "vertical criterion". The major question this contribution is focused on is: How far are students without specific informatics training able to identify and verbalize functions and parameters related to objects in real life?

In the years 2013 and 2014 I have conducted a couple of classroom activities in a high school that were related to functionality. The students had to associate things from everyday life to laboratory equipment with the same functionality (abstraction), verbalize this common function (conceptualization) and name parameters (parametrization). For example, a glass tube has the same function as a trail, this function can be verbalized by the verb "to guide" and typical parameters are fluids in case of a tube and people in case of a trail.

Before I present more details of these classroom exercises let me briefly characterize the three facets of functional analysis from the perspective of Raymond Cattell's theory of fluid and crystalline intelligence (Cattell, 1963). Abstraction by classifying activities or tools, is related to fluid intelligence, since it does not require language skills. It is rather a general ability to solve problems in a novel situation independent from specific knowledge or experience. For example when Jenny associates a glass tube to a trail, she compares typical processes related to tubes and trails and finds a common principle. On the other hand, *verbalizing* functions and parameters implies a lot of crystallized intelligence, which is the ability to use skills, knowledge and experience. It is language-related and culture-dependent.

3 Activity 1: Functionality of Laboratory Equipment

The participants got a worksheet depicting items from a chemistry lab (glass tube, spoon, Erlenmeyer flask etc.) on the left hand. On the right hand side there were things from everyday life. Although the items were from different domains and had different structures, some of them had similar functions. For example, a rubber plug and a crown cap look different and are made of different materials but they are both used to close containers to keep the content safe. The students' task was

1. to connect corresponding items by a line and
2. to name the function they have in common and write the words on the line.

The search for similar functionality corresponds to browsing through class libraries looking for appropriate functions for a software project.

75 high school students from grade 6 and 7 (age 11 to 14, average age 11.8, including 42 girls and 31 boys) were asked to perform this task. Beside the given example (rubber plug and crown cap with the function: to close) there were seven intended relations. Three images were meant as distractors and were not expected to be associated to anything from the complementary domain: Erlenmeyer flask, pasta, glass slide. The students found additional unexpected associations. For example, one person connected protective goggles with a knife and as a common function he called protection.

The students found an average of 6.5 pairs of corresponding objects and verbalized an average of 4.0 functions. Table 1 shows some results from the analysis of students' work.

Figure 1: Worksheet from activity 1 “What has the same function?”

Table 1: Some results from “What has the same function?” (n=75)

Laboratory device	expected association (percent)	most selected unexpected (percent)	wording for expected functions (examples)	verbalizing a function, including unexpected	referring to structure (material, shape)
glass tube	trail (29%)	slide (25%), macaroni (25%)	guide, transport	44%	23%
spoon patel	excavator (65%)	knife (19%)	dig, pick, excavate	65%	0%
mortar	knife (69%)	excavator (9%)	destroy, crush	67%	0%
protective goggles	umbrella (65%)	glass slide (16%)	protect	85%	0%
sieve	barriers for cars (28%)	barriers at queue (28%)	sort out, prevent big things from entering	44%	5%
funnel	barriers at queue (16%)	barriers for cars (28%)	let through only a little	33%	10%
One-hole rubber stopper	door (22%)	macaroni (25%)	close, shut, bar	37%	16%

The second column (expected association) tells objects from everyday life that share a common function with a laboratory device from the first column and the percentages of students who have chosen this association. The third column shows the most popular unexpected associations. The students used a variety of phrases to describe the functionality. Column 4 tells a few examples.

In some cases students did associate objects because common structural features (like shape and material) instead of common functionality. For example some students connected a glass tube with macaroni and wrote "both have a hole in the middle". The last column shows the percentage of such misunderstandings.

The findings from this activity demonstrate that students in grade 6 and 7 are able to distinguish between structure and function, classify objects according to functionality and verbalize a function. Some functions (like the common function of a spoon and an excavator) are easier to identify than others (like the common function of a glass tube and a trail). Why are some connections easier to find than others? A possible reason could be the degree of abstraction involved. A simple approach to determine the level of abstraction would be to compare the parameters. The common function of a spoon and an excavator is to move portions of amorphous material (like sand or powder). The parameters are quite similar. The common function of a glass tube and a trail is to guide objects from one location to another. In this case the parameters are fluids resp. humans, which are very different. Torreano et al. (2005) define levels of abstraction for metaphors in a quite similar way by checking common elements in the metaphorical and the literal meaning of a phrase.

4 Activity 2: Functional Analysis of Electrolysis

57 students (31 boys, 23 girls, 3 did not tell the gender) from chemistry classes in grade 10 performed a functional analysis of an electrolysis apparatus consisting of a battery (power supply), ammeter, two electrodes in a U-tube with diaphragm, filled with a solution of copper chloride (see Fig. 2 in the middle).

The students were asked to

- connect as many parts as possible from the electrolysis apparatus in the middle of the worksheet to objects from everyday life around it, which have the same function,
- verbalize the common functions,
- name the parameters at both ends of the connecting lines,

- draw an image of another object that shares a function with one of the part of the apparatus and connect it.

Before they started, the terms “function” and “parameter” were explained discussing the example given on the worksheet: Ammeter and ruler have the same function *to measure*. The parameter (the entity that is measured) is *electric current* in case of the ammeter and *length* in case of the ruler.

The worksheet suggested a functional decomposition of the electrolysis apparatus. A component may have several different functions. The negative electrode for example (1) attracts positive ions (like a magnet attracting iron) and (2) donates electrons to positive ions (like a person donating presents).

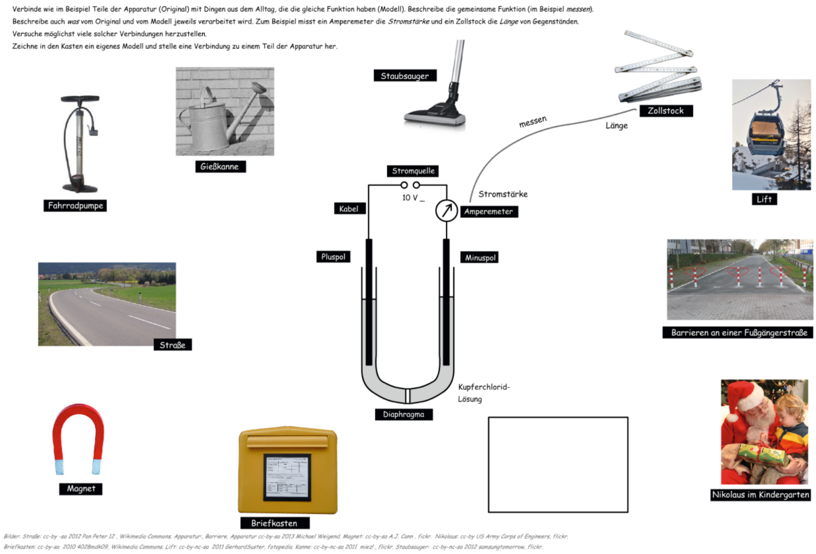


Figure 2: Worksheet “Functional Analysis of Electrolysis”

Beside the example (ruler connected to ammeter) there were nine more images from everyday life to work on. On average the students drew 6.0 connecting lines and verbalized 4.1 functions. Only verbal expressions indicating purposeful activity (e.g. attraction, storage, to guide, to donate) were accepted as proper function names. Expressions referring to structural properties (e.g. to have a positive pole, to need electricity) were not. Whereas naming functions seems to be pretty much part of common knowledge at the age of 16, identifying parameters is not. The average number of parameter pairs was 0.8. Only 33 % of the students were able to name parameters at all. Parameters seem to mark a barrier, a transition from “common sense” to computational thinking that requires a special education.

13 students (23 %) created an additional image, 9 of them connected it to a part of the apparatus and 5 verbalized a function.

5 Activity 3: Functional Analysis of a Spectrophotometer

22 students from a high school chemistry class in grade 13 (5 boys, 17 girls, age 18 to 20) who had studied the principle of operation of a spectrophotometer were asked to perform a functional analysis of this device by connecting parts of the apparatus to objects from different domains with the same function (Fig. 3). Additionally they tried to verbalize the function and name parameters. At the beginning of this exercise the given example was explained: The common function of a tap and a light bulb is *to emit*. The parameters are *water* in case of the tap and *light* in case of the light bulb. The students discussed the matter in small groups and solved the task in a collaborative way.

There were seven parts and seven objects from everyday life to connect. Again, in some cases the students found not intended relations, which still might be considered to be reasonable.

However, the plausibility of the Table 3 shows some findings. Columns 2–4 tell how far the students were able to find a connection and name a common function and parameters, disregarding the plausibility (or correctness) of their choice. The last column tells the percentage of students who have chosen the intended pair of objects.

Later the students (n=21) evaluated the activity by rating the degree of agreement with some statements. They reported, that they talked about the spectrometer (average degree of agreement, ADA: 95 %), discussed at least one issue controversially (ADA 86 %), got a better understanding of a spectrophotometer (ADA 77 %) and had fun (ADA 76 %). Only a minority felt that it was not interesting (ADA 21 %) and took too much time (38 %).

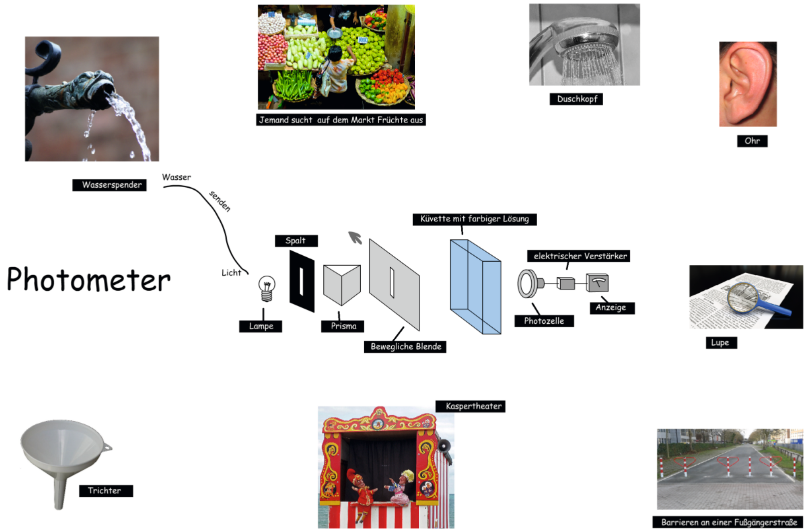


Figure 3: Worksheet „Functional Analysis of a Spectrophotometer".

Table 2: Some results from "Functional Analysis of a Spectrophotometer".

Part (corres-ponding object)	connection only	connection and function name	connection, name and parameters	having chosen the intended pair
Slit (funnel)	5%	14%	82%	59%
Prism (shower head)	0%	9%	91%	95%
Movable slit (person selecting fruits)	0%	23%	77%	50%
Cuvette with colored liquid (barriers)	14%	5%	73%	50%
photocell (ear)	0%	18%	82%	50%
Amplifier (magnifying glass)	0%	23%	77%	100%
Display (puppet theatre)	0%	23%	77%	50%

6 Benefits from Computational Thinking for Understanding Science

In the previous sections I have presented some empirical findings on students' competence of functional modelling. This competence is usually one of the major goals of computer science education at high schools. The idea of programming projects in the classroom is not to produce software specialist but to foster "computational thinking" (Wing) that is of use in wider areas of knowledge processing.

The question is: Does the ability of functional modelling help learning and understanding sciences like chemistry and physics? Andrea di Sessa (2002) illustrates the advantages of algebra for understanding physics. He presents Galileo's original proofs of simple propositions in kinetics, which were written without any equations, since Galileo did not know algebra. These proofs are very difficult to understand. But every ninth-grader can proof the same propositions just by transforming equations, and gets some understanding this way. Students do not learn the competence of handling equations in physics but (basically) in math lessons.

What about functional modelling? Let me mention four issues:

1. Functional modelling by defining functions with parameters is a pattern that might help understanding structures. It is an approach to cope with complexity by decomposing that not limited to the design of software systems.
2. Science students sometimes mix up structure and function. In chemistry classes students sometimes say that a negative electrode attracts positive ions in a solution (like Cu2+) through "magnetic force". In fact the physical cause for the movement of ions is "electric force". But people have much more experience with magnets than with electrically charged bodies. Thus "magnetic attraction" is often just meant (metaphorically) as a *functional* concept (to attract = being magnetic). This can be clarified by a defining functions and parameters in a quasi-programming style. The pattern "a function processes parameters" forces to explicate the difference between a magnet and a negative electrode.
3. Misconceptions often remain in the dark, just because nobody talks about them. Collaborative functional modelling encourages discussions and explication of ideas. Here is an opportunity to make useful mistakes. This way misconceptions can be "diagnosed" and "cured". This is comparable to using mathematical techniques to check the plausibility of a scientific calculations and chains of evidence.
4. The technique of functional modelling is a facet of computational thinking. It cannot just be noted like a piece of information, but it must be practised (rehearsed) and reflected, to be understood and to be of use. It is a competence. To develop this competence is an object of computer science rather than natural science education.

References

Baddeley, A. (2003): Working Memory Looking Back and Looking Forward. *Nature Reviews Neuroscience*, Vol. 4, 829–839.

Cattell, R. B. (1963): Theory of fluid and crystallized intelligence: A critical experiment. *Journal of educational psychology*, 54(1), 1.

Dehn, M. J. (2008): *Working Memory and Academic Learning*. John Wiley & Sons, Hoboken, New Jersey.

diSessa, A. A. (2001): *Changing minds, Computers, learning, and literacy*. MIT Press, Cambridge, MA.

Jaeggi, S. M., Buschkueh, M., Jonides, J., Perrig, W. J. (2008): Improving fluid intelligence with training on working memory. In *PNAS*, Vol. 105, No. 19, 6829–6833.

Kroes, P. (1998): Technological explanations: The relation between structure and function of technological object. In *Techné: Journal of the Society for Philosophy and Technology*, Vol. 3, No. 3.

Lakoff, G., Núnez, R. E. (1997): The Metaphorical Structure of Mathematics: Sketching Out Cognitive Foundations for a Mind-Based Mathematics. In Mathematical Reasoning. Analogies, Metaphors, and Images (ed. Lyn D. English). Lawrence Erlbaum Associates, Publishers, Mahwah, London, 21–92.

Schwill, A. (1994): Fundamental ideas of computer science. *Bulletin – European Association for Theoretical Computer Science*, 53, 274–274.

Torreano, L. A., Cacciari, C., Glucksberg, S. (2005): When Dogs Can Fly: Level of Abstraction as a Cue to Metaphorical Use of Verbs. In *Metaphor and symbol*, 20(4), 259–274.

Weigend, M. (2007): *Intuitive Modelle der Informatik [Intuitive Models of Computer Science]*, Universitätsverlag Potsdam.

Biography

Michael Weigend studied Computer Science, Chemistry and Education at the University of Bochum and at the University of Hagen and received a PhD in Computer Science from the University of Potsdam. He is a teacher at a secondary school in Witten, Germany and he has taught Didactics of CS at the University of Hagen for almost 20 years. He has published several books on computer programming, web development and visual modelling.

Short Papers

Computational Thinking: Videogames, Educational Robotics, and other Powerful Ideas to Think with

Rosa Bottino, Augusto Chioccariello
Istituto per le Tecnologie Didattiche
Consiglio Nazionale delle Ricerche
via De Marini 6
16149 Genova, Italy
{bottino, augusto}@itd.cnr.it

Abstract: Digital technology has radically changed the way people work in industry, finance, services, media and commerce. Informatics has contributed to the scientific and technological development of our society in general and to the digital revolution in particular. Computational thinking is the term indicating the key ideas of this discipline that might be included in the key competencies underlying the curriculum of compulsory education. The educational potential of informatics has a history dating back to the sixties. In this article, we briefly revisit this history looking for lessons learned. In particular, we focus on experiences of teaching and learning programming. However, computational thinking is more than coding. It is a way of thinking and practicing interactive dynamic modeling with computers. We advocate that learners can practice computational thinking in playful contexts where they can develop personal projects, for example building videogames and/or robots, share and discuss their construction with others. In our view, this approach allows an integration of computational thinking in the K-12 curriculum across disciplines.

Keywords: Computational thinking, programming in context, informatics education

1 Background

The educational potential of informatics was underlined from the beginning in educational technology research studies. Even if the relationship between computers and teaching to support pupils' thinking in schools has been variously conceptualised (Wegerif, 2002), one of the first identified objectives of the drive towards "computer innovation" in education was to develop new skills in students to allow their integration in a society that is deeply changed by information technologies.

In particular, in early years, an important line of study in educational technology was related to the teaching of programming. The advent of the microcomputer in the early 1980s and the development of general purpose programming languages opened up the possibility of using computers to a wide range of users and stressed the necessity of learning how to interact with them. In different contexts, the introduction to programming main ideas was carried out, through different modalities and approaches, not only in professional courses but also as part of school basic education (Olimpo, Persico, Sarti, Tavella, 1985). The declared aim was to provide some elementary notions on topics such as programming languages and methods, algorithm development, modelling of situations, use of correct and not ambiguous language. These notions were considered important both for a basic knowledge of the discipline and for the possibilities they offer in the teaching of other more traditional disciplines like mathematics (Ralston, 1981).

The evolution of hardware and software, that made computer interaction ever easier, and the parallel evolution of cognitive and pedagogical frameworks led to a gradual shift of interest, apart from specialized education, from the integration of informatics elements in school curricula to the implementation of new ICT-based educational applications and to the development and use of computer-linked methods, contents and tools for transforming and improving teaching and learning processes (Bottino, 2004). Technology design and use was progressively considered in relation to the whole teaching and learning process where a crucial role is assigned not only to the tool but also to the definition of meaningful practices through which technology can be used effectively to reach specific learning goals (Bottino, Ott, Tavella, 2011).

This line of evolution does not mean that research studies in the educational value of programming completely disappeared. They remain in the framework of constructivist approaches and mainly imply the development and use of educational specifically targeted languages, microworlds and programmable construction kits in the Logo tradition. Programming is often associated with writing code; for Papert it is a way of thinking. Papert (1980) stressed the importance of supporting active thinking on the part of learners by means of programming concrete objects that can provide immediate feedback and concepts reification. Knowledge emerges as a result of an active engagement with the world through the creation and manipulation of artefacts that are seen as objects to think with. The Logo turtle, both the virtual screen version and the robotic one, is the most famous example of a computational tool to "think with" applied to differential geometry, a powerful mathematical idea that Logo makes concrete and accessible to children.

In the course of time the interest in the educational value of informatics never disappears, currently such interest has expanded in the presence of a wide debate on the foundations of informatics and on the definition of the informatics skills to be included in basic education. "Computational thinking" emerges as the main keyword which is now broadly considered to underline informatics core skills.

2 Computational Thinking

"Computational Thinking" is the title of a "viewpoint" published in the Communications of the ACM in March 2006 by Jeanette Wing (2006). The article argues that "computational thinking is a fundamental skill for everyone, not just for computer scientists" and "we should add computational thinking to every child's analytical ability" (p. 33). The article has stimulated a lively international debate and reflections of prestigious institutions. For example, the USA National Research Council has organized two workshops and published the related reports: the first on computational thinking (National Research Council, 2010) and the second on its pedagogical implications (National Research Council, 2011). The reports, however, also document a failure: the participants did not reach a consensus on the definition of computational thinking. Subsequently, Aho (2012) and Wing (2011) did propose slightly different, but equivalent, definitions. Wing's version is: "the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms".

The debate around these concepts is active in European countries as well: the Royal Society (2012), for example, published the report "Shut down or restart? The way forward for computing in UK"; the Académie des Sciences intervened on this subject with the report (2013) "L'enseignement de l'informatique en France – Il est urgent de ne plus attendre". Moreover, Informatics Europe and the ACM Europe Working Group on Informatics Education

(2013) urged Europe “not to miss the boat” on this subject. All reports call for a change in the curricula to make room for informatics.

The history of the relationship between informatics and education points out that attempt to add a new discipline to an already crowded K-12 curriculum presents problems and furthermore it is not necessarily a warrant for computational thinking to be really mastered in way suitable to be universally applied. Introducing computational thinking into schools requires special attention to contents and levels. In terms of content, subjects like science, technology, engineering and math (STEM) seem to offer a direct match with informatics key ideas. However, other subjects can be considered as well. For example, constructing video games and interactive storytelling are activities related to media art that can offer interesting connections with computational thinking. In the following, we will consider educational robotics and digital games as two exemplary contexts that are actively contributing to introduce computational thinking in education.

Robotic construction kits, such as the LEGO Mindstorms or Lilypad Arduino (Buechley, Qiu, Goldfein, de Boer, 2013) have revitalized the idea of end-user programming. Computational textile kits, like Lilypad Arduino, address the gender gap by contextualizing computational thinking into digital versions of arts and crafts activities such as sewing. The construction of an autonomous robot or an interactive garment encompasses both the physical assembly of the artefact and programming the rules that control the artefact behaviour. Nowadays, defining robot behaviour is facilitated by the design of the kit: sensor and actuators are “smart”, i.e. their hardware contains intelligent components that reduce the complexity of the programming tasks. Furthermore, modern end-user programming environments featuring visual and tangible interfaces support the learner in writing a program by providing a structured context and powerful primitives while taking care of the syntactical aspects of coding. Robotic construction kits endowed with rule based visual and tangible programming environments can enable four-six years old children to program the behaviour of their robots (Bers, Flannery, Kazakoff, Sullivan, 2014 and Chioccariello, Manca, Sarti, 2004). These kits provide opportunities for children to explore issues of control and enable them to build and play with things that act as if they had a will of their own. Robotic construction kits are microworlds that well exemplify some important concepts that are usually mentioned when reflecting on the educational value of informatics, for example, to get in touch with powerful ideas such as feedback and emergent behaviours.

Digital game-based learning is a novel approach in the area of education and lifelong learning, displaying great potential as an active form of knowledge creation. Since games are now available in different platforms and devices, such as mobile devices, they offer the possibility to take learning outside the classroom and can provide a fun and interesting way of learning anytime, anywhere. Learning by playing is probably the ideal condition of education. One can play by the rules or with the rules, in the sense of building a new game. Video games are largely an example of playing by the rules, but games creation too can became a very valuable educational activity, able to trigger students' transversal skills, such as reasoning abilities, creative attitudes and digital competences. Specific environments to support games building activities have began to appear on the market and there is an increasing interest in their educational use. Kodu is an innovative environment for the creation of video games inspired by robot behaviour programming (Coy, 2013). Scratch, a visual programming environment where the instructions are assembled like LEGO building blocks (Resnick et al., 2009), is another popular environment for building games and interactive stories.

The construction of artefacts, robots or video games alone does not guarantee for learning. Computational activities should be embedded into an environment fostering collaboration, discussion, and reflection. Traditionally collaboration and discussion activities take place in classrooms where the teacher plays the role of mediator. How-ever, learning activities are increasingly developing also outside the school. Internet provides the condition for the birth of online social communities that involve learners of all ages. On-line communities of practice involving video games and robot con-structions are part of this movement (Kafai and Burke, 2013). They take advantage of national and international competitions where the practitioners meet (e.g. FIRST Lego League, National STEM Video Game Challenge, Robocup junior).

3 Conclusions

We advocate that learners should practice computational thinking in playful contexts where they can develop personal projects, for example building videogames and/or robots, share and discuss their construction with others. In our view, this approach allows, in principle, the integration of computational thinking in compulsory school curriculum. Let us outline a possible progression of playing, creating and exploring with programmable play kits leading to learning the mathematic and physic of motion. The math and physics of motion are usually included in secondary school programs. Computer simulations and computer-based laboratory are often used in science classes. Developing models of phenomena is advocated, but difficult to pursue due to the complexity of dealing with the required math and physics concepts. Computational model, unlike the corresponding math representations, are executable model that can be more easily tested, debugged and refined. Research in the use of computational modeling in science education provides evidence that this approach is more learnable. However, familiarity with computational thinking skills and concepts is required to pursue this approach.

Objects falling, colliding, in equilibrium are phenomena that interest children from an early age; they develop sophisticated way of thinking trough experimenting with the physical world. Learning to interpret the same phenomena according to physical laws, however, requires a conceptual change mediated by education. Computers provide children with the opportunity to play with moving objects, but also to create their own moving objects, and eventually computational models of motion. At an early age, for example, they might construct cartoon like animations of a flying bird in a visual programming environment like Scratch. Later on they might add a jumping behavior to the character of a video game they are building. Computational models of projectile motion, like the ones in the popular Angry Birds video game, are within the reach of lower secondary school learners (DiSessa, 2000).

Learners should engage in minds-on and hands-on activities to understand science. Robotic kits allow the design and construction of tangible autonomous artifacts moving and interacting with the environment. Robotic projects might start, at primary school, with a simple reactive construction and eventually, at secondary school, reach the complexity of constructing a robot that plays soccer and it is capable of kicking the ball in the goal of the opposing team, as in the Robocup junior competition. A longitudinal progression of the use of programmable play kit in formal and informal education might provide the necessary familiarity and competencies in computational thinking skills to innovate education across the curriculum.

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55 (7), 832–835. doi: 10.1093/comjnl/bxs074

Académie des sciences (2013). *L'enseignement de l'informatique en France - Il est urgent de ne plus attendre*. Retrieved from http://www.academie-sciences.fr/activite/rapport/rads_0513.pdf

Bers, M. U., Flannery, L., Kazakoff, E. R., Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. doi: 10.1016/j.compedu.2013.10.020

Bottino, R. M. (2004). The evolution of ICT-based learning environments: which perspectives for the school of the future?, *British Journal of Educational Technology*, 35 (5), 553–567. doi: 10.1111/j.0007-1013.2004.00413.x

Bottino, R. M., Ott, M., Tavella, M. (2011). Scaffolding Pedagogical Planning and the Design of Learning Activities: An On-Line System. *International Journal of Knowledge Society Research* (IJKSR), 2(1), 84–97. doi:10.4018/jksr.2011010107

Buechley, L., Qiu, K., Goldfein, J., de Boer, S. (2013). *Sew Electric: A Collection of DIY Projects That Combine Fabric, Electronics, and Sewing*. HLT Press.

Chioccariello, A., Manca, S., Sarti, L. (2004). Children's playful learning with a robotic construction kit. In Siraj-Blatchford, J. (Ed), *Developing New Technologies for Young Children* (93–112). Trentham Books.

Coy S. (2013). Kodu game lab, a few lessons learned. XRDS: *Crossroads. The ACM Magazine for Students*, 19 (4), 44–47. doi: 10.1145/2460436.2460450

DiSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. Mit Press.

Kafai Y. B., Burke Q. (2013, March). The social turn in K-12 programming: moving from com-putational thinking to computational participation. In *Proceeding of the 44th ACM technical symposium on computer science education* (603–608). ACM. doi: 10.1145/2445196.2445373

Informatics Europe & ACM Europe (2013). *Informatics education: Europe cannot afford to miss the boat*. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. Retrieved from http://europe.acm.org/iereport/ACMandIEreport.pdf

National Research Council (2010). Committee for the Workshops on Computational Thinking: *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: National Academies Press.

National Research Council (2011). Committee for the Workshops on Computational Thinking: *Report of a workshop of pedagogical aspects of computational thinking*. Washington, DC: National Academies Press.

Olimpo, G., Persico, D., Sarti, L., Tavella, M. (1985). An experiment in introducing the basic concepts of informatics. In *Proceedings of the Fourth World Conference on Computers in Education*, WCCE'85, 31–38.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Ralston, A. (1981). Computer science, mathematics and the undergraduate curricula in both. *American Mathematical Monthly*, 88(7), 472–485.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52 (11), 60–67. doi: 10.1145/1592761.1592779

The Royal Society (2012). *Shut down or restart? The way forward for computing in UK schools*. Retrieved from http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf

Wegerif, R. (2002). *Literature Review in Thinking Skills, Technology and Learning*. Bristol Futurelab. Retrieved from http://archive.futurelab.org.uk/resources/documents/lit_reviews/Thinking_Skills_Review.pdf

Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. doi: 10.1145/1118178.1118215

Wing, J. (2011). Research notebook: Computational thinking – What and why? *The Link Magazine*, Spring. Carnegie Mellon University. Pittsburgh. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why

Biographies

Rosa Bottino is director of the Istituto per le Tecnologie Didattiche at the Consiglio Nazionale delle Ricerche. She promoted and chaired both national and European projects and Networks of Exellence in Technology Enhanced Learning. She is member of international research associations and journals editorial boards. Dr. Bottino received international awards like IFIP Silver Core Award and IFIP Outstanding Services Award.

Augusto Chioccariello is a researcher of the Istituto per le Tecnologie Didattiche at the Consiglio Nazionale delle Ricerche since 1986. Dr. Chioccariello completed his Physics Degree (in 1980 at the University of Naples. From 1982 to 1986, he worked in physics education at the Educational Technology Centre, UC Irvine. In 1996, he was awarded a NATO senior researcher fellowship at MIT Media Lab with Prof. Sey-mour Papert.

The Technology Proficiency Self-Assessment Questionnaire (TPSA): Evolution of a Self-Efficacy Measure for Technology Integration

Rhonda Christensen, Gerald Knezek
University of North Texas
3940 N. Elm St., #G-150,
Denton TX 76207
rhonda.christensen@unt.edu, gknezek@gmail.com

Abstract: The Technology Proficiency Self-Assessment (TPSA) questionnaire has been used for 15 years in the USA and other nations as a self-efficacy measure for proficiencies fundamental to effective technology integration in the classroom learning environment. Internal consistency reliabilities for each of the five-item scales have typically ranged from .73 to .88 for preservice or inservice technology-using teachers. Due to changing technologies used in education, researchers sought to renovate partially obsolete items and extend self-efficacy assessment to new areas, such as social media and mobile learning. Analysis of 2014 data gathered on a new, 34 item version of the TPSA indicates that the four established areas of email, World Wide Web (WWW), integrated applications, and teaching with technology continue to form consistent scales with reliabilities ranging from .81 to .93, while the 14 new items gathered to represent emerging technologies and media separate into two scales, each with internal consistency reliabilities greater than .9. The renovated TPSA is deemed to be worthy of continued use in the teaching with technology context.

Keywords: Technology proficiency, self-efficacy, teacher competencies

1 Introduction

The Technology Proficiency Self-Assessment (TPSA) questionnaire has been used for 15 years in studies regarding technology integration in the classroom in the USA and other nations (Christensen, Knezek, 2001; Morales, Knezek, Christensen, 2008; Gencturk, Gokcek, Gunes, 2010). The instrument was originally developed by Ropp (1999) to measure teacher confidence (self-efficacy) when using technology for educational purposes. The TPSA, as refined by Christensen, Knezek and Ropp (Knezek, Christensen, Miyashita, Ropp, 2000), measured four types of technology proficiencies: using electronic mail, using the World Wide Web, using technology applications, and teaching with technology. Each area was represented by five items, rated on a scale of 1 = strongly disagree to 5 = strongly agree. These areas were based on the International Society for Technology in Education (ISTE) standards that existed at the time of the instrument's development (ISTE, 1993; ISTE, 2007).

2 Discussion

2.1 Grounding in Self-efficacy

Self-efficacy is the concept that provides the underlying rationale for the TPSA. Self-efficacy is based on Bandura's (1977, 1986) social development theory, and is sometimes defined as the expression of beliefs of individuals related to their own capacity to perform a certain behavior (Gencturk, Gokcek, Gunes, 2010). As reported by Gencturk, Gokcek, and Gunes (2010), teachers with higher self-efficacy are more ambitious and passionate in their teaching (Tuckman, Sexton, 1990), while Collis (1996) observed that the teacher shapes "… the success or eventual lack of success in any computers-in-education initiative" (p. 22). Henson (2003) found that teacher efficacy is an important component of a classroom teacher's success or failure. The authors of this paper have proposed an operational definition of self-efficacy as *confidence in one's competence*.

2.2 Reliability and Validity for 20-Item TPSA

A study using 1999 and 2000 classroom teacher data in the USA state of Texas yielded reliability estimates (Cronbach's Alpha) ranging from α =.73 (Email), to α =.87 (Integrated Applications) (Christensen, Knezek, 2001, p. 37). These reliability estimates fell in the range of "respectable" to "very good", according to guidelines provided by DeVellis (1991, p. 85). For the 2004 Texas data set gathered by Morales (2005), subscale reliability estimates ranged from .73 to .88, very close to those reported for previous studies. Cronbach's Alpha for the total 20-item scale was .93 (N=877). Gencturk, Gokcek, and Gunes (2010) reported a total scale (20 item) reliability of α = .94 for primary school teachers (n = 205) in Turkey, very close to the internal consistency reliability (α = .93) reported by Morales (2005) for teachers in the USA and Mexico. The 20 items included in Version 1.0 of the TPSA are listed in Table 1.

2.3 Refinement Process

Beginning in 2012, Christensen and Knezek drafted item revisions aimed at renovating the TPSA while Christensen and Williams (2014) explored extensions of self-efficacy measures into emerging domains such as social media and mobile learning. The new 34-item version of the TPSA was administered to 72 preservice and inservice teachers during 2014 for the primary purposes of verifying the reliability of the revised, original scales, and to explore constructs emerging from the additional 14 items added in the area of emerging technologies and media. Cronbach's alpha for the revised versions of the original four scales were found to be: 1) Email = .85 (Items 1–5); 2) WWW = .87 (Items 6–10); 3) Integrated Applications = .81 (Items 11–15); and Teaching with Technology = .84 (Items 16–20). These fall in the range of "very good" to "excellent" according to guidelines by DeVellis (1991). Items with wordings different from those indicated in Table 1 were: 3. Create a distribution list to send e-mail to several people at once ("Nickname or alias" has been replaced with "distribution list"); 6. Use an Internet search engine (e.g., Google) to find Web pages related to my subject matter interests ("Infoseek or Alta Vista" has been replaced by "Google"); 11. Use a spreadsheet to create a bar graph of the proportions of the different colors of M&Ms in a bag ("Pie chart" had been replaced by "bar graph"); 12. Create a newsletter with graphics ("and text in three columns" has been omitted from the question); and 13. Save documents in formats so that others can read them if they have different word processing programs (e.g., saving Word, RTF, or text) ("Clarisworks" has been omitted). All other items from the original 20 remained unchanged.

Table 1: Technology proficiency self-assessment questionnaire version 1.0

Scales	TPSA Version 1.0 Items
Email	**I feel confident I could…**
	1. send e-mail to a friend.
	2. subscribe to a discussion list.
	3. create a “nickname” or an “alias” to send e-mail to several people at once.
	4. send a document as an attachment to an e-mail message.
	5. keep copies of outgoing messages that I send to others.
World Wide Web	6. use an Internet search engine (e.g., Infoseek or Alta Vista) to find Web pages related to my subject matter interests.
	7. search for and find the Smithsonian Institution Web site.
	8. create my own World Wide Web home page.
	9. keep track of Web sites I have visited so that I can return to them later. (An example is using bookmarks.)
	10. find primary sources of information on the Internet that I can use in my teaching.
Integrated Applications	11. use a spreadsheet to create a pie chart of the proportions of the different colors of M&Ms in a bag.
	12. create a newsletter with graphics and text in 3 columns.
	13. save documents in formats so that others can read them if they have different word processing programs (eg., saving Word, ClarisWorks, RTF, or text).
	14. use the computer to create a slideshow presentation.
	15. create a database of information about important authors in a subject matter field.
Teaching with Technology	16. write an essay describing how I would use technology in my classroom.
	17. create a lesson or unit that incorporates subject matter software as an integral part.
	18. use technology to collaborate with other interns, teachers, or students who are distant from my classroom.
	19. describe 5 software programs that I would use in my teaching.
	20. write a plan with a budget to buy technology for my classroom.

Factor analysis (principal components, varimax rotation) was applied to the data for the 14 items gathered to represent emerging technologies and media, in order to determine if identifiable constructs emerged from this set. Data from the 72 subjects completing surveys in 2014 were used in this analysis.

Two factors were extracted based on the eigenvalue > 1 criteria, accounting for 69 % of the common variance in the data. The eight items loading most strongly on the first extracted factor are listed in factor loading order in Table 2. These focus on using emerging technologies and media for teacher professional development and instruction. Internal consistency reliability for this scale was found to be α = .93 for this set of data.

Table 2: Teacher Professional Development and Instruction Items (α = .93, 8 items)

Item	Factor Loading
30. I feel confident that I could download and read e-books.	.91
31. I feel confident that I could download and view streaming movies/video clips.	.88
32. I feel confident that I could send and receive text messages.	.82
29. I feel confident that I could download and listen to podcasts/audio books.	.74
34. I feel confident that I could save and retrieve files in a cloud-based environment.	.72
23. I feel confident that I could create a wiki or blog to have my students collaborate.	.62
24. I feel confident that I could use online tools to teach my students from a distance.	.61
27. I feel confident that I could use mobile devices to connect to others for my professional development.	.61

The six items loading most strongly on the second extracted factor are listed in factor loading order in Table 3. These focus on using emerging technologies and media to promote student learning. Internal consistency reliability for this scale was found to be α = .90 for this set of data.

3 Conclusion

Analyses of TPSA data have led the authors to conclude that the original 20-item instrument remains functional and worthy of use on a broad scale after 15 years, while the 14 new emerging technology items included on Version 2.0 of the TPSA address two measurement domains. Additional research with a larger sample is planned to determine if either of the two new constructs identified for emerging technologies are strongly related to the four constructs represented by the traditional four assessment scales.

Table 3: Emerging Technologies for Student Learning items (α = .90, 6 items)

Item	**Factor Loading**
22. I feel confident that I could use social media tools for instruction in the classroom (ex. Facebook, Twitter, etc.).	.83
25. I feel confident that I could teach in a one-to-one environment in which the students have their own device.	.83
21. I feel confident that I could integrate mobile technologies into my curriculum.	.82
26. I feel confident that I could find a way to use a smartphone in my classroom for student responses.	.80
28. I feel confident that I could use mobile devices to have my students access learning activities.	.76
33. I feel confident that I could transfer photos or other data via a smartphone.	.62

References

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. Psychological Review, 84(2), 191–215.

Bandura, A. (1986). *Social Foundations of Thought and Action: A Social Cognitive Theory*. Englewood Cliffs, NJ: Prentice Hall.

Christensen, R., Knezek, G. (2001). *Equity and diversity in K-12 applications of information technology*. Denton, TX: University of North Texas, IITTL.

Christensen, R., Williams, M. (2014). Relationships Between Teacher Personality Type and Technology Integration Indicators. In Searson, M., Ochoa, M. (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014* (2745–2753). Chesapeake, VA: AACE.

Collis, B. (1996). The internet as an educational innovation: Lessons from experience with computer implementation. *Educational Technology*, 36(6), 21–30.

De Vellis, R. F. (1991). *Scale development*. Newbury Park, NJ: Sage.

Gencturk, E., Gokcek, T., Gunes, G. (2010). Reliability and validity study of the technology proficiency self-assessment scale. *Procedia Social and Behavioral Sciences 2*, 2863–2867.

Henson, R. K. (2003). Relationships between preservice teachers' self-efficacy, task analysis, and classroom management beliefs. *Research in the Schools*, 10(1), 53–62.

ISTE. (1993). *ISTE foundation standards 1993/Adopted verbatim by NCATE*. users.wfu.edu/cunninac/technologyteacheredsummit/foundationstandards1993.pdf

ISTE (2007). *The evolution of NETS for teachers*. Available from http://cnets.iste.org/Teachers/t_timeline.html

Knezek, G., Christensen, R., Miyashita, K., Ropp, M. (2000). *Instruments for assessing educator progress in technology integration*. Denton, TX: IITTL.

Morales, C. (2005). Cross-cultural validation of technology integration. Unpublished doctoral dissertation. University of North Texas.

Morales, C., Knezek, G., Christensen, R. (2008). Self-efficacy ratings of technology proficiency among teachers in Mexico and Texas. *Computers in the Schools*, 25(1/2), 126–144.

Ropp, M. (1999). Exploring individual characteristics associated to learning to use computers in preservice teacher preparation. *Journal of Research on Computing in Education*, 31(4), 402–424.

Tuckman, B. W., Sexton, T. L. (1990). The relation between self-beliefs and self-regulated performance. *Journal of Social Behavior and Personality*, 5, 465–472.

Biographies

Rhonda Christensen, Ph.D. is a research scientist in the Learning Technologies Department in the College of Information at the University of North Texas. She is Associate Director of the Institute for the Integration of Technology into Teaching and Learning at UNT. She is Co-PI for the Going Green! Middle Schoolers Out to Save the World (MSOSW) project funded by the U.S. National Science Foundation (NSF). She serves as a Co-Chair on the Society for Information Technology in Teacher Education (SITE) Research and Evaluation Committee. She has co-authored and taught a pre-service technology integration course for more than 15 years. She also teaches doctoral courses on research design and analysis. Her research interests are the impact of technology integration in education, enhancing STEM education in middle schools and mobile learning in education.

Gerald Knezek, Ph.D. is Regents Professor of Learning Technologies at the University of North Texas. He is Past President (2008–2011) of the Society for Information Technology & Teacher Education (SITE). He is currently Lead Principal Investigator for the US National Science Foundation Innovative Technologies Project Going Green! Middle Schoolers Out to Save the World. He was a Fulbright Scholar at the Tokyo Institute of Technology during 1993–94 and shared time between Texas and Ecuador on a Fulbright Senior Specialist appointment during 2006–07. He held a second Fulbright Senior Specialist appointment to The Netherlands during 2011–12. His research interests include measuring attitudes toward information technology, assessing the impact of technology integration, bridging formal and informal learning through ubiquitous learning environments, enhancing creativity and fostering innovative STEM education.

How does the Implementation of a Literacy Learning Tool Kit influence Literacy Skill Acquisition?

Anthony Maina Gioko, Enos Kiforo Angondi, Rosemary Waga
Aga Khan Academy
Mbuyuni Road P.O. Box 90066-80100 Mombasa
{gioko, enos.kiforo}@akam.ac.ke
Aga Khan Primary School
Vanga Road P.O Box 90063-80100 Mombasa
rosemarywaga@yahoo.com

Abstract: This study aimed at following how teachers transfer skills into results while using ABRA literacy software. This was done in the second part of the pilot study whose aim was to provide equity to control group teachers and students by exposing them to the ABRACADABRA treatment after the end of phase 1. This opportunity was used to follow the phase 1 teachers to see how the skills learned were being transformed into results. A standard three-day initial training and planning session on how to use ABRA to teach literacy was held at the beginning of each phase for ABRA teachers (phase 1 experimental and phase 2 delayed ABRA). Teachers were provided with teaching materials including a tentative ABRA curriculum developed to align with the Kenyan English Language requirements for year 1 and 3 students. Results showed that although there was no significant difference between the groups in vocabulary-related subscales which include word reading and meaning as well as sentence comprehension, students in ABRACADABRA classes improved their scores at a significantly higher rate than students in control classes in comprehension related scores. An average student in the ABRACADABRA group improved by 12 and 16 percentile points respectively compared to their counterparts in the control group.

Keywords: ABRACADABRA, Early Literacy, Achievement, Teachers, Learners

1 Background

Studies have shown that children in Kenya, especially girls, are not achieving educational success to the extent they are capable of (Dubeck, Jukes, Okello, 2012; Watkins et al., 2010; UNESCO, 2010). International statistics show that Kenyan rates in literacy, particularly English literacy, are well below the standards of developed countries in the OECD.

Only 34 % of boys and 27 % of girls complete secondary school in Kenya (United Nations Children's Fund, 2012). Additionally the Uwezo Report has shown that Learning levels remain low and that one third of all children in Class 3 cannot even read a Class 2 level story (Uwezo, 2012).

It is in this context that the Aga Khan Academy Mombasa, which runs Professional Development Courses for teachers of English to improve the levels of English language performance, and Centre for the Study of Learning and Performance (CSLP), a research centre from Montreal Canada, collaborated on a project; using technology to teach early literacy. This project in which twelve schools participated exposed learners to ABRACADABRA, which is an early literacy software that supports the learning of English.

2 Methodology

2.1 Research Design

As a follow-up to phase 1 pre-test/post-test control group design, phase 2 study focussed on delivering delayed treatment to the control participants. After six phase 1 control teachers and one new teacher were trained with ABRA, they used ABRA with their students. Pre- and post-test results collected in the six classes during phase 1 study were compared with the phase 2 post-test scores. Both pre- and post-test results were collected from students in a new class in phase 2.

Seven English teachers and their grade three (standard) (N=235) and grade two (N=39) students participated in phase 2 of the project. From the total sample of 276 students, test results were missing for 95 students for a variety of reasons. Specifically, 16 students were transferred to different classes during the year, 15 students were new to their classes and 64 did not attend lessons on the days of testing during either phase 1 or phase 2 or both. These reductions resulted in usable data for 181 students (N2= 33 and N3= 148).

2.2 Instruments

A set of instruments were used to gather data. These have been divided into two, those for students' achievement measures and teachers' and classroom measures.

The Group Reading Assessment and Diagnostic Evaluation, GRADE (Williams, 2001) which is a standardized measure designed to assess reading skills and to monitor reading progress was used in phase 2 of the project. The *Literacy Instruction Questionnaire* (LIQ; Abrami et al., 2011) was used to collect information about the English Language instruction. This is a CSLP-developed instrument that elicits teacher reports on aspects of the instructional methods they used in their classroom over the past semester. Specifically, the questionnaire includes two sections to explore: 1) approaches to reading and comprehension instruction; and 2) use of technology. *An ABRA classroom observation form* (Centre for the Study of Learning and Performance, 2012) was used to collect additional data about the details of classroom instruction. The form is a CSLP-developed instrument and includes four sections, general classroom environment. Lesson plans, *involving the integration of ABRA into language instruction*, were requested from teachers in order to cross-validate the trace data collected by the software as students used ABRA. *Videotaping of English language instruction* was conducted during the 11-week long intervention in order to capture teachers' pedagogical techniques and students' learning experiences with ABRA.

ABRA trace data reports were retrieved as an objective measure of ABRA use in order to complement and corroborate the implementation information collected via teachers' self-reports, lesson plans, and observations. *Teacher final interviews* were conducted shortly after the end of the intervention. The objective of holding these interviews was to learn about teachers': attitudes towards the use of the technology when teaching generally, and in the use of ABRA specifically.

3 Analyses

Before the main analyses, we applied standard procedures to clean the data. Only data from individuals who completed tests at all times of testing were used for analyses.

For all GRADE achievement measures, simple difference scores (post-test minus pre-test) were used. Although the difference score has often been maligned as an unreliable index of change, recent work (Zimmerman, Williams, 1998; Thomas, Zumbo, 2012) demonstrates a flaw in this perspective and suggests that the resulting non-use of difference score analysis is unwarranted.

To allow for the comparability of reading achievement results collected by means of GRADE Level 1 and 2 tests we used fall grade 1 norms to convert grade 3 students' raw scores. Hence three composite scores (Vocabulary, Comprehension and Total Composites) were used in lieu of four subtest raw scores (Word Reading, Word Meaning, Sentence Comprehension and Passage Comprehension). Listening Comprehension scores were not used.

4 Findings

Results indicate that there was a remarkable positive change in scores after ABRA intervention. Even though teachers reported significantly higher frequencies of using computers for instruction at the post-test than at the pretest, these uses yet fall between rarely and occasionally. The ABRA trace data retrieved twice indicated that depending on class, an average student spent between 12 to 27 minutes per week on ABRA activities. With the exception of large classes. Specifically, grade three students spent more time on reading comprehension activities in comparison to phase 1 of the project. Additionally lesson plans showed that teachers attempted to integrate ABRA activities targeting different literacy components including phonemic awareness and phonics (e.g., word changing, syllable counting), fluency (e.g., speed reading, expression), comprehension (e.g., story elements, comprehension monitoring), and writing (e.g., word spelling). Observation forms form the implementation team showed that the teachers were able to guide the students through activities effectively. The information on ABRA activities reported in observations corroborates that in lesson plans when both are available.

Finally interviews conducted via web conferencing revealed that teachers developed some comfort level using technology and a positive shift in their attitudes towards using technology to teach literacy. The teachers also expressed a positive shift in their own teaching of English Language. The majority of teachers used a dedicated iBook in their for low ability students which brought these students up to speed with the others. Additionally most of the students who initially lacked general ICT skills, quickly learned how to successfully navigate within the software. The findings also show that Cultural sensitivity regarding the ABRA stories did not pose problems for the students some whose parents started enrolling them in lessons outside of school in order for them to learn more about computers.

5 Discussion

Reading achievement data show that after eleven weeks of ABRA exposure, students in the seven delayed ABRA classes showed similar gains with phase 1 ABRA students in regard to comprehension-related and total scores on the GRADE assessment.

Delayed ABRA students' improvements were almost ignorable on vocabulary subtest pertaining to student capacity to decode, recognize sight words and to understand their meaning. However, in phase 1 experimental and control students gained equally on the vocabulary subtests of the GRADE. An explanation we favour relates to the nature of measure of vocabulary knowledge. For instance, standardized tests used to measure changes in vocabulary skills may be accountable for capturing only small average effects in the development of vocabulary delivered by ABRA. According to the NRP report (2000), standardized tests of vocabulary development are not sufficiently sensitive. The more the vocabulary test matches the instructional context and content, the more appropriate this assessment is to measure the impact of instruction on vocabulary skills.

The data from the teacher self-reports provide some detail about the literacy instruction that occurred in the experimental and control classes. ABRA teachers' responses to the survey and interview questions reveal some positive shifts in their literacy instruction including alphabetics, comprehension and writing. Certainly their comfort level with teaching with computers improved – statistically significant gains were indicated with respect to teachers' declared use of computers. Nevertheless these im-provements were not powerful enough to make them use computers more than rarely and occasionally.

The ABRA trace data reports, observational data and lesson plans showed that during the eleven-week intervention, teachers developed a certain capacity in the integration of the ABRA software throughout the English Language curriculum. In comparison with phase 1, delayed ABRA teachers were able to address key literacy components in a more balanced fashion. However, time of exposure of an individual student to ABRA activities can be improved considering the length of a standard lesson. Additionally ABRA integration requires more effort. Better links should be established between the ABRA content and the student activities in the classroom. Similar to the phase 1, while there was a shift towards serving in new roles as facilitators of their students' learning, the period of time was rather short for them to stop using a teacher-directed approach for their literacy instruction.

6 Conclusion

This study showed that there is some level of comfort when teachers plan together observe each other and reflect on their lessons. Despite having three days of training only, the follow up scaffolding and collaboration with each other showed that teachers could actually become competent users of technology and get good results for the teaching and student achievement. However, there should be a system of tracking on how the teachers are using the software so as to advise them on effective infusion of the technology. Suffice it to say that technology integration should not be done as standalone but should be integrated in the instructional design and pedagogy

References

Abrami, P. C., Wade, A., Lysenko, L., Marsh, J., Gioko, A. (2013, May). *A study of ABRACA-DABRA early literacy software in Mombasa*, Kenya: Phase one report. Montreal, QC: Centre for the Study of Learning and Performance.

Abrami, P. C., Lysenko, L., Wade, A., Pillay, V. (2011). *Literacy Instruction Questionnaire (LIQ)* (Unpublished instrument). Montreal, QC: Centre for the Study of Learning and Performance. Retrieved from http://doe.concordia.ca/cslp/cslp_cms/sites/all/themes/jframe/downloads/PDF/LIQ_2011.pdf

Al-Alwami, A. (2005). *Barriers to Integrating information Technology in Saudi Arabia Science Education*. Doctoral dissertation, the University of Kansas, Kansas.

August, D., Shanahan, T. (Eds.). (2006). *Developing literacy in second-language learners: A report of the National Literacy Panel on Language-Minority Children and Youth*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bunyi, G. (2005). Language classroom practices in Kenya. In Lin, A. M. Y., Martin, P. W. (Eds.) *Decolonization: Language in education, policy and practices* (pp. 131–152). Clevedon: Multilingual Matters Ltd.

Commeyras, M., Inyega, H. (2007). An integrative review of teaching reading in Kenyan primary schools. *Reading Research Quarterly*, 42(2), 258–281.

Dubeck, M. M., Jukes, M. C. H., Okello, G. (2012). Early primary literacy instruction in Kenya. *Comparative Education Review*, 56, (1), 48–68. Retrieved from: http://www.jstor.org/stable/10.1086/660693

Hattie, J. A. C. (2009). *Visible learning: A synthesis of 800+ meta-analyses on achievement*. London: Routledge.

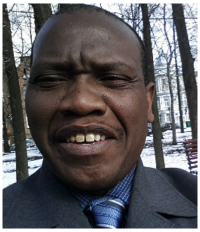
National Reading Panel (NRP). (2000). *Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction* (NIH Pub. No. 00–4754). Retrieved from National Institute of Child Health & Human Development website: http://www.nichd.nih.gov/publications/pubs/nrp/pages/report.aspx

Slavin, R. E., Lake, C., Davis, S., Madden, N. (2009, June). *Effective programs for struggling readers: A best evidence synthesis*. Baltimore, MD: Johns Hopkins University, Center for Data-Driven Reform in Education. Retrieved from http://www.bestevidence.org/reading/strug/strug_read.htm

Williams, K. T. (2001). *Group Reading Assessment and Diagnostic Evaluation (GRADE)*. Pearson Education Inc. Available at: http://www.pearsonassessments.com/HAIWEB/Cultures/en-us/Productdetail.htm?Pid=GRADE

Biographies

Enos Kiforo Ang'ondi is a Master Trainer at Aga Khan Academy Mombasa. He is also the LTK local coordinator overseeing the implementation of the ABRACADABRA Project in Mombasa Kenya. As a Professional Development Teacher, Enos has worked on many teacher training programmes sponsored by Aga Khan University and Aga Khan Academy Mombasa. Prior to his current appointment Enos was part faculty at Mount Kenya University and high school teacher.



Gíokò Maina is a Doctoral Candidate In Management who is the Head of the professional development center at the Aga Khan Academy (Microsoft Mentor School). Gíokò manages four projects in professional development, learning enrichment and teacher preparation programmes. Gíokò is a Global Education and Future of learning Facilitator. He is also a workshop leader and online facilitator. Gìokò has keen interest in professional learning, transfer of skills and sustainable development.

Rosemary Waga is a lead teacher at the Aga Khan Primary school early primary year's section. Rose is a master trainer and a digital Ambassador with the learning tool kit. Waga is a secretary for English language teachers association which does teacher peer mentorship on communicative approach of teaching English.

Assignments in Computer Science Education: Results of an Analysis of Textbooks, Curricula and other Resources

Laura Ohrndorf
University of Siegen, Germany
laura.ohrndorf@uni-siegen.de

Abstract: In this paper we describe the recent state of our research project concerning computer science teachers' knowledge on students' cognition. We did a comprehensive analysis of textbooks, curricula and other resources, which give teachers guidance to formulate assignments. In comparison to other subjects there are only a few concepts and strategies taught to prospective computer science teachers in university. We summarize them and given an overview on our empirical approach to measure this knowledge.

Keywords: Pedagogical content knowledge, computer science teachers, students' knowledge, students' conceptions

1 Introduction

The formulation and preparation of assessments is an everyday task for a teacher. As it involves a deeper knowledge of the specific topic and an understanding for students' cognition it can be classified as a part of the pedagogical content knowledge (PCK) defined by Shulman (1986).

The competency to choose, design and prepare questions and tasks for assessment and practice is part of many content-specific definitions of the PCK. In the COACTIV project the pedagogical content knowledge of mathematics teachers was tested. One of the main categories is "Tasks": "When appropriately selected and implemented, mathematical tasks lay the foundations for students' construction of knowledge and represent powerful learning opportunities" (Krauss, Baumert, Blum, 2008). The importance of this category is supported by other research projects as well (Riese et al., 2013). Of course, this can also be applied to didactical computer-science courses in university. An analysis of German university curricula for computer science education has shown that only a minority of them includes the design of tasks and preparation of assessment. Meanwhile there is a broad focus on the actual process to solve tasks (especially problem-solving).

Our experiences in teachers' training in computer science education show, that there are deficiencies in the knowledge of assessment preparation and creation. Students know about general methods and concepts like didactical models and learning software or resources. But they often fail to transfer these in order to fulfill their own needs. At times they have problems to evaluate the applicability of concepts (i.e. the use of a role play with senior class students). Nevertheless they have to do this in their teaching traineeship at the latest. This leads to the question when and how computer science teachers learn to develop assignments.

2 Related Work

Research and also general concepts for the creation and formulation of tasks in computer science education are rare. To substantiate this assertion we also analyzed textbooks used in content-specific didactical courses in mathematics and physics at German universities. These often outline and exemplify criteria to evaluate tasks, methods to create an appropriate context and other instructing information (e.g. (Kircher, Girwidz, Häußler, 2009), (Reiss, Hammer, 2010)). In contrast these are rare and less comprehensive in computer science education literature. However the relevance of this competency is undeniable, since it is mentioned in several scientific papers. For instance Hubwieser et al. (2013) developed a literature-based category system for PCK containing the topic "tasks and assignments" as part of the category "Specific teaching elements".

Ragonis et al. (2010) focus on the development and implementation of computer science teacher preparation programs. They did a survey with heads of these and present a list of main characteristics of computer science education, which should be understood by teachers. One of these are "[...] strategies to evaluate pupils' products: class assignments, homework, exams, projects" as a topic which should be understood by prospective computer science teachers.

Hazzan et al. (2011) emphasize the need of teachers' knowledge about assignments: "It is important, however, that computer science teachers be aware of the fact that additional types of questions exist." They introduce three steps for the preparation process of questions in computer science classes: Planning, Solving and Estimation of the needed time to solve the question. These involve different specific sub-topics and basic principles. Furthermore they give an overview of types of questions in terms of 12 individual concepts and illustrate the possibility to join them. They extend this by three different kinds of questions: story questions, closed questions and unsolved problems, and consider the possibility to combine these. A successful design of assessments is only possible with a reliable estimation of the difficulty level. Thompson et al. (2008) present the approach to evaluate this for programming tasks by the use of Bloom's taxonomy. Although this refers to programming only, it is presumable that an adaption to other contents is feasible.

Schlüter (2008) evaluates the difficulty of task through ten criteria: redundancy, level of formalization, closeness to the world of experience, level of abstraction, complexity, cognitive level, type of knowledge, area of content and processes (according to the German Gesellschaft für Informatik Standards (2008)) and the area of requirement.

The Bebras contest is a computer science contest for students. It is conducted in several countries. The tasks are prepared by "The Bebras International Tasks Workshop" where experts from every participating country suggest tasks and discuss them (2013). In the context of the Bebras contest (resp. the German contest "Informatik-Biber") the creation and evaluation of tasks has been documented in several papers. The basic goal of the contest is to encourage students for the field of computer science and to give them a deeper understanding of how modern information technology works (Dagiene, 2008). Thus it differs from tasks used in school environments, which are meant to support a students' learning processes and for assessment later on. Nevertheless the Bebras goals are relevant for them as well as they require similar criteria for their applicability. For instance these are the time needed to answer, an adequate difficulty level and are easily understandable (Dagiene, Fuschek, 2008). Although these are documented and described very well, they cannot be measured easily. Pohl & Hein (2013) point out that the experience gained from the improvement process of single challenge tasks can be transferred to other tasks as well. Additionally they might lead to general practices, which help to assure the quality of computer science tasks.

The selection of research and practical results presented above gives an insight into the current state of research. It is unquestionable that there are numerous research approaches. Nevertheless these are apparently rarely connected. Moreover in comparison to other subjects general and fundamental concepts are missing. We are planning an empirical survey in order to retrieve results on the actual development of competencies in the field of assessment development. These are part of a research project on the development on teachers' knowledge about student's conceptions and cognition.

3 Research Method

In our project we aim to measure teachers' competencies in students' knowledge and conceptions. To classify the fields of knowledge we adopted two categories for out test items from the COACTIV project, which are PCK student and PCK task. PCK student describes the analysis and prediction of typical students' errors or misconceptions. The focus of PCK task is the knowledge of how a student answers a task. We extend the definition of the latter with the competency to create proper task, i.e. the formulation of questions for expected answers. The test will be conducted with computer science teacher trainees and teachers. We have developed a first set of test-items, which was used for a survey in December 2013. This gave us first insights on the applicability of the items.

In this article we focus on a test item of the category PCK task, which pays attention to elementary aspects for the formulation of assignments: the knowledge of how to formulate tasks and the knowledge of how a student will answer them (Ohrndorf, 2013). The latter also involves an imagination of different approaches, which will be explained in the following example test item.

Example task: In the field of cryptology, the fundamentals of symmetric key encryption were introduced to an 8th class. As their homework your students answer the following textbook task:

> *"You want to exchange hand-written messages with your classmate. How can you assure that he or she is the only one who is able to read the message? Describe your approach in detail."*

"Please describe at least two possible answers your students could give."

It's obvious that this task allows a number of answers. We confine ourselves to three examples, which were given by university students attending computer science didactical courses:

1. We first agree on an encryption code. This can be a number from 1 to 26. When we write a message every letter is represented by a number: a = 1, b = 2, c = 3... To encrypt a text every letter is replaced by adding the code to the number to get the secret letter. When we use 9 "a" would become "j" and "b" would become "k".

2. We implement a transposition cipher by creating a unique pattern, which describes the way the message can be read when encrypted. This can be a spiral starting in the middle and rising anti-clockwise. The encrypted text must then be written in this scheme to read the message.
3. We both download a tool that allows encrypting text with a complex encryption method and agreeing on a common key. To exchange the encrypted messages we can print them or write them down.

These answers give an idea of different approaches and methods. As they are basically right or follow a right approach, they do not fulfill further demands on the understanding of students' cognition. Answer 1 explains the use of a classical Caesar cipher, which is often taught in the introduction of encryption techniques. Answer 2 uses a transposition cipher (route cipher). This allows a more secure encryption, but might be too ambitious for an 8^{th} grade student. Answer 3 actually circumvents the specified encryption of hand-written messages by typewriting it and using available software. This is undoubtedly the weakest answer since it describes an application-oriented solution without any usage of actual computer science knowledge or curricula knowledge.

4 Conclusion and Future Work

As described above one of the main challenges in the project is the adequate rating of the teachers' answers. Answers might be right from a technical view, but are not satisfying concerning their relevance and applicability in lessons. Due to this, the answers cannot be scored by a simple rating, which defines right and wrong. Therefore a specific rating procedure has to be defined for every test item, where the ambitions and goals are considered.

We have just done a preliminary assessment in December 2013 and are evaluating the results. This will help us to develop the main survey, which will be conducted at the end of 2014.

References

Dagiene, V. (2008). The bebras contest on informatics and computer literacy-students' drive to science education. In *Joint Open and Working IFIP Conference. ICT and Learning for the Next Generation* (pp. 214–223). Kuala Lumpur.

Dagienė, V., Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *Lecture Notes in Computer Science* (pp. 19–30). Berlin Heidelberg: Springer.

Gesellschaft für Informatik e.V.: *Grundsätze und Standards für die Informatik in der Schule*. LOGIN-Verlag, Berlin, Arbeitskreis „Bildungsstandards" der GI. 2008. URL: http://bebras.org/about/ (07.11.2013).

Hazzan, O., Lapidot, T., Ragonis, N. (2011). *Guide to Teaching Computer Science – An Activity-Based Approach*. London: Springer.

Hubwieser, P., Berges, M., Magenheim, J., Schaper, N., Bröker, K., Margaritis, M., Schubert, S., Ohrndorf, L. (2013). Pedagogical Content Knowledge for Computer Science in German Teacher Education Curricula. In *WiPSCE '13, Proceedings of the 8^{th} Workshop in Primary and Secondary Computing Education*. Aarhus.

Kircher, E., Girwidz, R., Häußler, P. (2009). *Physikdidaktik*. Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-662-22299-7

Krauss, S., Baumert, J., Blum, W. (2008). Secondary mathematics teachers' pedagogical content knowledge and content knowledge: validation of the COACTIV constructs. *ZDM*, 40(5), pp. 873–892.

Ohrndorf, Laura (2013). A taxonomy of errors for computer science education. In *World Congress on Computers in Education (WCCE)*, Torun.

Pohl, W., Hein, H.-W. (2013). Aufgabenqualität im Informatik-Biber. In Breier, N., Stechert, P., Wilke, T. (Eds.), *INFOS 2013, 15.GI-Fachtagung „Informatik und Schule" Praxisband* (pp. 51–58). Kiel.

Ragonis, N., Hazzan, O., Gal-Ezer, J. (2010). A survey of computer science teacher preparation programs in Israel tells us: computer science deserves a designated high school teacher preparation, *Symposium on Computer science*, pp. 401–405.

Reiss, K., Hammer, C. (2010). *Grundlagen der Mathematikdidaktik* (p. 147). Birkhäuser.

Riese, J., Borowski, A., Fischer, H., Gramzow, Y., Kulgemeyer, C., Reinhold, P., Schecker, H., Tomczyszyn, E., Walzer, M. (2013). Professional knowledge of physics student teachers – ProfiLe-P. In Blömeke, S., Zlatkin-Troitschanskaia, O. (Hrsg.), *KoKoHs Working Papers*, 3. Berlin, Mainz: Humboldt-Universität & Johannes Gutenberg-Universität.

Saeli, M., Perrenet, J. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education*, 10(1), pp. 73–88.

Schlüter, K. (2008). Je schwieriger die Aufgabe, desto klüger der Kopf? In *Didaktik der Informatik - Aktuelle Forschungsergebnisse* - 5. Workshop der GI-Fachgruppe „Didaktik der Informatik" (pp. 77–86). Bonn: Köllen.

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational researcher*, 15(2), pp. 4–14.

Thompson, E., Grove, H., Luxtonreilly, A., Whalley, J. L., Robbins, P. (2008). Bloom's taxonomy for CS assessment. In *Proceedings of the Tenth Australasian Computing Education Conference (ACE2008)*, pp. 155–162.

Biography

Laura Ohrndorf received her diploma in computer science (2011) from the University Duisburg-Essen, Germany. Since July 2011 is a research assistant at the University of Siegen. Her main research interests concern misconceptions in computer science education and their identification.

Programming for Non-Programmers – Fostering Comprehension Capabilities by Employing a PRS

Rachel Or-Bach
The Max Stern Yezreel Valley College
Emek Yezreel 19300, Israel
orbach@yvc.ac.il

Abstract: The study reported in this paper involved the employment of specific in-class exercises using a Personal Response System (PRS). These exercises were designed with two goals: to enhance students' capabilities of tracing a given code and of explaining a given code in natural language with some abstraction. The paper presents evidence from the actual use of the PRS along with students' subjective impressions regarding both the use of the PRS and the special exercises. The conclusions from the findings are followed with a short discussion on benefits of PRS-based mental processing exercises for learning programming and beyond.

Keywords: Novice programmers, comprehension, tracing, personal response systems

1 Introduction

Throughout the short history of personal computers there have been attempts to delineate possible benefits of teaching programming to non-programmers, from children to higher education students. The potential benefits range from motivation and creativity to logical thinking, problem solving skills and more (diSessa, 2000; Kay, 1991; Papert, 1980). Teaching programming to students who have no special interest or inclination towards programming and related topics requires at least two things: providing students with some motivation and carefully designing the instructional materials and strategies. A course titled "Design of computerized games and interactive stories" was developed to address these requirements. A multimedia interactive development environment and a goal of designing a working game were chosen to enhance students' motivation. Special emphasis in the instructional materials was given to the relationship between explaining, tracing and writing skills in introductory programming (Lister, Fidge, Teague, 2009). A previous paper (Or-Bach, 2013) dealt with the contribution of such a course to the development of higher order cognitive skills and ICT competencies. The study reported in the present paper involved the employment of specific in-class exercises using a Personal Response System (PRS). These exercises were designed to enhance students' ability to explain a given code in natural language with focus on abstracting the goal of the code. This paper presents some evidence from the PRS use along with students' subjective impressions regarding the use of the PRS and the special exercises.

2 Related Work

2.1 Learning to program – The role of comprehension exercises

A significant distinction in the literature (see for example Robins et al., 2003) is between studies that explore program *comprehension* and those that focus on program *generation*. No doubt that these two topics are correlated, as comprehension activities must be performed during development and debugging. Lister et al. (2009), in an effort to build and refine the SOLO taxonomy for programming learning, stress this issue and the use of assessment tasks of "explain in plain English" that require the ability to grasp the overall goal or meaning of the code. Studies they carried showed evidence of a relationship between explaining, tracing and writing skills in introductory programming. They claim with pedagogical implications that until students have acquired minimal competence in tracing and explaining, it may be counterproductive to have them write a great deal of code. Robins et al. (2003) identify the issue of mental models as an important factor in learning and teaching programming. They claim that writing a program involves maintaining different kinds of "mental model". They distinguish between the model of the program as was intended, and the model of the program as it actually is that requires the ability to trace code for predicting its behavior. In our study both skills of tracing and explaining played a central role.

2.2 Games, education and personal response systems

The use of computer games for learning is widely advocated. Computerized games can provide a context for developing various skills, but constructing such games seems to entails even wider educational benefits. Robertson and Howells (2008) argue that authoring a game can engage students in authentic rich tasks offering a good degree of learner autonomy. Vos et al. (2011) report on a study where they compared using a game versus constructing a game. The results suggest that constructing a game might be a better way to enhance student motivation and deep learning than playing an existing game. Game design is becoming a popular strategy for enhancing students' interest and skills with computer technology, deepening understanding of scientific principles, fostering critical media literacy and more (Hayes et al, 2008). Personal Response Systems (PRS) are technologies that facilitate interactivity with an audience. Such systems enable participants to instantly respond to posed questions using some end user device. The system can receive participants' responses and can provide a representation (or several ones) of the collected data. Instructors in variety of disciplines are increasingly using audience response systems to increase participation, engagement and learning (See for example, Beatty, Gerace, 2009; Mareno et al., 2010). SMS-HIT (Kohen-Vax et al., 2012) is a personal response system based on mobile devices for SMS and web response provision. The system is designed for teaching purposes enabling instructors to prepare and enact personal response activities in actual instructional setting for any subject domain. Readymade activities are stored in a repository and could be later on copied, modified and reused. The SMS-HIT system was used in our study for several types of in-class activities, as will be described in the next section.

3 The Course – General characteristics of the course

The rationale for the "Design of computer-based games and interactive stories" course was to provide a motivating and engaging context for introducing computer programming to students in the behavioral sciences departments in our college. This elective course is already offered for several years and is accompanied by an ongoing action research. The instructional approach, intermediate assignments and research tools were refined during the years. The programming environment chosen for this course is Scratch, which is a visual programming environment that lets users create interactive, media-rich projects (Resnick et al., 2009; Maloney et al. 2010). A key goal of Scratch is to introduce programming to those with no previous programming experience. Programming is done by snapping together command blocks to control sprites. Specific blocks can be placed on top of a stack of blocks to trigger that stack in response to some run-time event. Multiple stacks can run at the same time to show simultaneous acts by different sprites. The programmer can watch stacks in the scripting area high-lighted when the action unfolds on the stage thus showing how the constructed scripts are interpreted by the computer. The course final assignment is the implementation of a game or interactive story using Scratch. Classwork consists of examples that students explore by "reading" and/or by executing the program; as well as other examples where students have to modify or construct a program in order to produce prescribed outcomes. In the last semester a PRS was employed with specially designed in-class exercises dealing with tracing and explaining Scratch scripts.

4 The Study – Participants, setting and tools

During the semester when this study was conducted the course lasted 14 weeks with a 2 hours session each week in a computer lab and 23 students participated in the course. The SMS-HIT PRS was used in most of the sessions, sometimes more than once but with an attempt not to use it too often. The exercises were presented through our LMS and included a link to a site where the response can be written or selected. The goal of the study was to investigate the contribution of these specific exercises to the learning of Scratch programming. Two types of questions were used: short free text ("What will be seen on the screen after executing the following script?"), and multiple-choice ones ("Do the two following scripts act the same? (Yes/No)"). The responses and their relative frequency were presented to the class at the end of the activity for further discussion and/or follow-up learning activities. A survey was administered to the students at the end of the course to collect students' impressions from the use of the PRS. The survey included 10 Likert-type items and a free text question regarding the special in-class exercises. An additional item dealt with the student's evaluation of his/her mastery of Scratch with relation to the class. Two of the 10 items dealt with factors related to the use of PRS in general (motivation and engagement) and the rest dealt with specific factors of the PRS use that may contribute to the learning of Scratch programming (requirement of mental processing, presentation of different answers, requirement to compare between scripts, discussions based on students' answers etc.) The scale for the Likert-type items was designed to obtain a finer resolution on the positive side of the scale because a goal of the study was to distinguish between the contributions of the different factors and the instructor's impression was that students favored in general the PRS use. The resulted scale included the following categories: Do not agree, Agree slightly, Agree moderately, Agree, Agree strongly.

5 Findings

The average response rate for the PRS based in-class exercises was 15 out of 23 course participants. The fact that not all the students provided responses for these exercises might be because the chosen PRS allows anonymity (anonymity might also encourage participation). For most of the multiple-choice questions, the frequency diagram that was produced and presented to the students exhibited a distribution of opinions. Looking at the class results, students were asked to re-examine their response by re-reading the code. Only then they were asked to actually check the behavior of the script(s). It seemed that for many students the combination of mental tracing along with the actual execution is required for the ultimate conviction. For exercises with free-text short responses, the resulted variety of responses enabled discussions that dealt with what answers are actually the same, what is correct, and what might be the cause for other responses. These discussions were productive for stressing the need for accuracy, for re-examining the various programming structures and for presenting possible misconceptions. Twenty students (out of 23) completed the survey. One item of the survey dealt with students' perception about their relative mastery of Scratch programming. It seemed important to check this because over-confidence or under-confidence might give a different interpretation for the survey results. The results show that in general students felt good about their mastery of Scratch programming. The averages for the different survey items were in the range 3.4–3.95, indicating appreciation for the various contributions of using the PRS. The standard deviation was around 1. The highest average (3.95) was for item 5 – "Mental execution of a script helps to deeply understand specific commands". The lowest average (3.4) was for item 3 – "The fact that I saw additional answers made me re-think my answer". None of the students chose "Do not agree" for items 1 and 2 that dealt with general contributions (motivation and engagement respectively). The averages and the distribution for the survey items are quite similar, not showing differences between students' perceived contributions for the various aspects of the PRS use. Thus further analysis was carried out with regard to the students. It turned out that the standard deviation for a student (across items) was around 0.5, while the standard deviation for an item was around 1. We also checked for each student the differences between the average appreciation of using a PRS (items 1 and 2) and the average appreciation for the specific use of a PRS for learning programming (items 3–10). The differences ranged between -0.9 and 1.9, indicating that different students experienced the use of the PRS differently (or interpreted it differently).

6 Conclusions and Discussion

Investigating difficulties of non-programmers, as well as respective instructional strategies, can be informative also for CS educators. In this study we tried to deal with the tracing and explanation skills, building on the evidence that was found of the relationship between explaining, tracing and writing skills in introductory programming (Lister et al., 2009). The general impression from the course, in comparison to previous years, is that the extensive use of tracing and explanation exercises was productive for learning to program with Scratch. From the instructor point of view, the in-class PRS-based activities were helpful for discovering students' difficulties and addressing them more effectively in class. The motivation and engagement provided by the PRS cannot be separated from the experience. Students' final submissions indicate good mastery of programming in Scratch based on what was learned in class and some self-learning. Results from the survey showed that in general students felt good about their programming capabilities in relation to the class. Results show that students had positive attitudes towards the PRS use during the course. The item that dealt with the use of mental execution of scripts for better understanding had the highest average and the highest number of students choosing "agree strongly". This fits well with the goals for the PRS-based exercises design. The fact that the distribution for all the items is similar might be because the various factors are highly correlated and students cannot differentiate between them thus consolidating a general attitude. This might explain also the fact that the variance among the students was much larger than the variance among the survey items. The tracing and explanation exercises can be beneficial also to the development of higher order cognitive skills related to abstraction, reasoning and use of mental models. Exercising "mental processing" seems important for current students that use technology extensively and tend to solve problems by tinkering or by very short chains of reasoning.

References

Beatty, I. D., Gerace, W. J. (2009). Technology-enhanced formative assessment: A research-based pedagogy for teaching science with classroom response technology. *Journal of Science Education and Technology*, 18(2), 146–162.

diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*. Cambridge, MA: MIT Press.

Hayes, E. R., Games, I. E., (2008). Making Computer Games and Design Thinking: A Review of Current Software and Strategies. Games and Culture, 3(3–4), 309–332.

Kay, A. (1991). Computers, Networks and Education. *Scientific American*, 138–48.

Kohen-Vacs, D., Ronen, M., Bar-Ness, O. (2012). Integrating SMS Components into CSCL Scripts. *7th IEEE International Conference on Wireless, Mobile & Ubiquitous Technologies in Education* (WMUTE 2012), Takamatsu, Japan.

Lister, R., Fidge, C., Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *SIGCSE Bulletin*. 41, 3 (July 2009), 161–165.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 1–15.

Mareno, N., Bremner, M., Emerson, C. (2010). The use of audience response systems in nursing education: best practice guidelines. *International Journal of Nursing Education Scholarship*, 7, 1–17.

Or-Bach, R. (2013). Higher Education – Educating for Higher Order Skills. *Creative Education*, 4(7A2),17–21.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67.

Robins, A., Rountree, J., Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13,137–172.

Vos, N., van der Meijden, H., Denessen, E. (2011). Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education, 56*, 127–137.

Biography

Rachel Or-Bach is a senior lecturer in the Information Management Systems department in Yezreel Valley College. She received her Ph.D. from the Technion. Her main research interest is design of interactive learning environments: representations, interactivity, and intelligent support and adaptation. She publishes in journals of IT for education, Science Education, Computer Science and Information Systems Education, Informing Science etc.

Mentoring in a Digital World: What are the Issues?

Christina Preston[1], **Sarah Younie**[2]
[1]MirandaNet
University of Bedfordshire, UK
christina@mirandanet.ac.uk
[2]MirandaNet
University of De Montford, UK
younie@dmu.ac.uk

Abstract: This paper focuses on the results of the evaluation of the first pilot of an e-mentoring unit designed by the Hands-On ICT consortium, funded by the EU LLL programme. The overall aim of this two-year activity is to investigate the value for professional learning of Massive Online Open Courses (MOOCs) and Community Online Open Courses (COOCs) in the context of a 'community of practice'. Three units in the first pilot covered aspects of using digital technologies to develop creative thinking skills. The findings in this paper relate to the fourth unit about e-mentoring, a skill that was important to delivering the course content in the other three units. Findings about the e-mentoring unit included: the students' request for detailed profiles so that participants can get to know each other; and, the need to reconcile the different interpretations of e-mentoring held by the participants when the course begins. The evaluators concluded that the major issues were that: not all professional learners would self-organise and network; and few would wish to mentor their colleagues voluntarily. Therefore, the e-mentoring issues will need careful consideration in pilots two and three to identify how e-mentoring will be organised.

Keywords: MOOCs, e-mentoring, professional development, ICT skills, user-centred

1 The Context for the Handson ICT MOOC

The EU LLL programme funded Hands-On ICT[1] to explore the value of Massive Online Open Courses (MOOCs) and Community Online Open Courses (COOCs) in professional learning. In essence, Hands-On is a holistic environment that provides teachers from higher education, vocational education and schools with everything they need to learn about making the right choice of ICT tools for a given pedagogical activity. The Hands-On ICT team from England, Greece, Slovenia, Spain and the Netherlands based the design of the MOOC on the contexts and practices that were identified in a report about existing e-learning projects already underway in Europe (Riviou, Barrera, Domingo 2014). The need was clear: the *Survey of Schools: ICT and Education (2013)* found that no reliable progress had been made in teachers' access, use and attitudes towards ICT in since the previous survey in 2006.

The design of the e-mentoring unit was based on the knowledge and experience of The MirandaNet Fellowship[2] in the context of professional e-learning in 'communities of practice' (Lave and Wenger, 1991; Wenger, 1998). In particular MirandaNet concentrates on the collaborative and interactive potential of e-learning in terms of building and publishing new knowledge called Braided Learning (Leask and Younie, 2001; Haythornthwaite, 2007; Preston, 2007; Cuthell, 2012).

1.1 The Pilot Course design

The evaluation was designed to provide ideas for a design in progress. Nearly thirty teachers in schools, VE and HE volunteered to pilot the four unit. The first three self-contained units exploring creativity techniques lasted one week each: **'concept mapping'**, **'the six hats'** and **'triggering questions'**. The average number of participants in each unit was eight, a 25 % drop out rate (Florjani and Lesjak, 2014)

1.2 The e-mentoring unit

This paper focuses on the fourth unit, e-mentoring that was two-weeks long to allow for interactions to build up. Six modules: **Mentoring in a Digital World**, looked at the how and the why of e-mentoring and served as an introduction to the activities; **Activity One**, Access, Motivation, Socialisation was

1 Hands-On ICT Project handsonict.eu.
2 MirandaNet Fellowship www.mirandanet.ac.uk.

linked to the discussion forum Working Online – your virtual classroom. **Activity Two** looked at Information Exchange and Knowledge Construction, and the focus of that forum is on Knowledge Building. Knowledge Development and Braided Learning were the subjects of **Activity Three**, with work in the forum dealing with Collaborative Knowledge Construction. The final course component, **Activity Four**, focused on using digital technologies to enhance learning and teaching opportunities, with Improving the Learning Experience as the subject of the forum. In the **Conclusion** participants examined Tools for Personal Development. **The forums** focused on ways of moving on.

2 Findings

2.1 The cohort for e-mentoring

Only two teachers of the thirty who signed up for the first three units elected to do this final two-week course. As a result the MirandaNet Fellows drew on their MOOC research group of thirty-two members who had volunteered interest specifically in e-mentoring. Six registered to join two participants from the first three units: eight for the final two-week unit. All except one were expert members of communities of practice with a strong online element: one school teacher was already teaching 86 pupils that year online; four had a Masters in online learning; one had a Ph.D.; six were over forty five: three were looking for retirement opportunities in online teaching.

The evidence of their sophisticated knowledge and experience emerged in advice given to others in the pre-course questionnaire. Overarching factors they considered important in online course design included simplicity; clear structure; alignment to learning objectives; structured introductory activities; a variety of learning tasks introduced slowly. They were all clear about the value of facilitation to engender a community spirit, maintain interest and building learning.

2.2 The technical adjustments

In the beginning the software team who were committed to a user-centred approach to development made several adjustments to the design as the participants expressed their difficulties. The most important changes were to the registration obstacles, to the visibility of the units to all registrants and the cancelling of different keys for each course. The immediate attention given to these problems ensured that the eight registered participants stayed in the course. In fact, one participant, here called Roger, also intervened on the technical front because another participant, Andrew, had significant problems that he was blaming on the course design. His outspoken frustration was threatening to disrupt the course. Roger, an experienced user of Moodle realised Andrew's interface was corrupted and phoned him in order to explain the cause of his frustration. This action by an expert participant exonerated the software design team from blame.

Three changes to the general design of the environment were recommended by participants: A guide to using the software to include: a series of 'How do I?' screenshots linked to FAQ; an introductory section linked to Step One, Access & Motivation; addition of a profile box because the participants wanted to get to know each other at the start; and an initial activity unit on using Moodle, with screen shots to guide participants through a range of competences, from understanding the difference between replying to a thread in a forum to uploading files. Combining these skills with socialisation activities would be a good way for participants to introduce themselves.

2.3 E-mentoring activity

The course tutors started two discussions in order to develop collaborative thinking: *hosting a discussion in the dark*; and, *Marriage between Mentoring and Moodle: can it work?* One contributor felt that the participants had not had sufficient opportunity to introduce themselves to each other and set up a discussion called: *Online Learners. Getting Started*. Another forum was *Do MOOCs change our expectations?* Forums were a rich, lively and expert source of information.

Both tutors and participants suggested at the start of the e-mentoring unit that the profiles should be more visible and should perhaps pop up with the photos when comments were made. The tutors' forum, *'Hosting a party in the dark'* attracted twenty-five replies that mostly concentrated on the positives of online learning: for example: "In my experience, working online I get to know my guests in more depth more quickly than I can working with them face to face. The online environment potentially affords a level of intimacy which is a privilege to work with". Another participant said, "We should share our expectations, worries and concerns about starting this course together". He then set up a forum, *Online Learners. Getting Started*: all the students participated in this. Overall there were two forums started by the tutors and four by the students. Generally the quality of the forum discussions was high, although the students tended to lose collaborative focus. Students suggested that the interaction of more students will help in the next pilot and the tutors need to be more active in steering the discussion towards collaboration. Humour was seen to be important in keeping the forum learning lively.

The interactive forums proved to be more popular than the learning activity modules: two students did activity one; two different students did activity two; only one did activity three; only one did activity four. However, it seemed that three out of eight of the participants who were active in the forums did not see the units and thought that the forums were the only activity. This design fault will be remedied.

2.4 Discussion

It must be emphasized that participants for all these four pilot activities were a-typical because they volunteered for a pilot in the first place. This is an important factor in considering how this course should be altered because the majority of these teachers were already e-mentoring and were motivated by their professional interest in the design of the software and the practice of others. The evidence from the discussions in the e-mentoring unit shows clearly that the participants already had an understanding of concepts and performative competence that may not necessarily be applicable to subsequent participants.

In addition, further discussion need to be held about whether an e-mentoring unit should be academic, or practical, or both. The plan and rationale behind this e-mentoring module was to develop mentoring skills in the participants – all profession-als – and let them be peer mentors. The pedagogical assumptions are therefore subtly different from those that would underlie a course purporting to teach people how to be mentors. Participants perceived a need for greater scaffolding of concepts, skills and activities before the final versions of the HandsOn ICT courses are launched if the more academic learning activities were still to be offered as well as the interactive forums.

3 Conclusion

The participants questioned the underpinning e-mentoring principle of the course as well as perceiving a lack of clarity about the role of an e-mentor because each student had different views. Also the mentoring role implies responsibility for other students and a generosity with time that cannot always be relied on. Questions were raised about whether there should be tangible rewards for mentoring effort other than personal satisfaction like accreditation. Since no payment would be involved qualifications in e-mentoring were mooted. But how would success in mentoring be judged: test scores; ICT competence; the quality of responses in a forum or whether the teachers have implemented these ideas in the classroom? Tests can validate knowledge as evidence: however, there should also be a way to validate performative evidence. One way is for the participant to upload an ICT artefact used to support learning and teaching, together with a commentary and evaluation. In this context the Hands-On team is exploring partnerships with Learning Designer[3] and Ingots[4]. Global publication could be another route that would motivates the teachers to develop artefacts to share more widely with others like the Mapping Educational Specialist knowhow (MESH)[5] initiative.

The major conclusion from the participants was that the designers of the second pilot need to engage in some significant rethinking because the underlying theory of Hands-On ICT, that all students are the drivers in their education and will self-organise and network, is not necessarily the case. Some will only want an academic course. Should the Hands-On ICT team cater for both kinds of professional learner?

In her article, *Hits and Myths: MOOCs may be a wonderful idea but they're not viable* (2014) Laurillard raises several good points about the problems of sustaining the free MOOC offers that are emerging globally from major universities and multinational companies. Judging on the results of the HandsOn ICT evaluation a point that the team has to tackle in the design of the second and third pilot is Laurillard's assertion that it is a myth that students will support each other's learning. The next evaluations of the second and third pilot will need to address the challenges of e-mentoring in a MOOC in some depth in order to design a sustainable model.

Thanks to John Cuthell and Alison Bulbeck for their helpful comments.

3 Learning Designer https://sites.google.com/a/lkl.ac.uk/ldse/.
4 INGOTS http://theingots.org/community/about.
5 MESH http://www.meshguides.org/.

References

Cuthell, J. (2012) *Learning in liminality*. Retrieved May 5, 2014, from http://www.virtuallearning.org.uk/.

European Commission (2013). *Survey of schools: ICT in education: benchmarking access, use and attitudes to technology in Europe's schools*. Retrieved May 7, 2014, from http://bookshop.europa.eu/.

Lesjak, B. (2014). A MOOC as an opportunity for the development of competence of creative use of information technology in teaching. In *21. konferenca Dnevi slovenske informatike*, Portorož, 14.–16. April 2014. Informatika–neizkoriščeni dejavnik razvoja. Ljubljana: Slovensko društvo Informatika. pp. 66–76.

Haythornthwaite, C. (2007). *New international theories and models of and for online learning*. Chicago IL, USA, First Monday.

Lave, J. and Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK: Cambridge University Press.

Laurillard, D. (2014). Hits and myths: moocs may be a wonderful idea but they are not viable. *The times education supplement January 16th*.

Leask, M., Younie, S. (2001). Communal constructivist theory: pedagogy of information and communications technology and internationalisation of the curriculum. *Journal of information technology for teacher education*, 10(1&2), pp. 117–134.

Preston, C. (2007). *Braided learning: promoting active professionals in education*. New international theories and models of and for online learning. C. Haythornthwaite. Chicago IL, USA, First Monday. Volume 12, number 8. Retrieved May 8, 2014, from http://firstmonday.org/issues/issue12_8/.

Riviou, K., Barrera,C. F., Domingo, M. F. (2014). "Design principles for the online continuous professional development of teachers", *14th IEEE International conference on advanced learning technologies (ICALT 2014)*, Athens, Greece, IEEE Computer Society, 7–10, July 2014Online Learning. London: Kogan Page.

Wenger, E. (1998). *Communities of practice: learning, meaning, and identity*. Cambridge, UK: Cambridge University Press.

Biographies

Christina Preston, Professor of Education Innovation, Bedfordshire University, publishes internationally about the use of digital technologies as a catalyst for enriching teaching and learning. She founded the MirandaNet Fellowship in 1992 that now has nearly 1,000 members in 80 countries. Christina has won five international awards for the design of innovative professional programmes promoting collaboration across national boundaries.

Sarah Younie, Innovation and Research Director at the MirandaNet Fellowship, has been involved in international research on educational technologies and teaching for twenty five years. Principal Lecturer at De Montfort University, she is also Associate Editor-in-Chief for the international Journal of Technology, Pedagogy and Education, and Editor-in-Chief of the MESH guides for ICT.

Key Competences with Physical Computing

Mareen Przybylla[1], **Ralf Romeike**[2]
[1]Didactics of Computer Science
University of Potsdam
A.-Bebel-Str. 89
14482 Potsdam, Germany
przybyll@cs.uni-potsdam.de
[2]Computing Education Research Group
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3
91058 Erlangen, Germany
ralf.romeike@fau.de

Abstract: Physical computing covers the design and realization of interactive objects and installations and allows students to develop concrete, tangible products of the real world that arise from the learners' imagination. This way, constructionist learning is raised to a level that enables students to gain haptic experience and thereby concretizes the virtual. In this paper the defining characteristics of physical computing are described. Key competences to be gained with physical computing will be identified.

Keywords: Defining characteristics of physical computing, key competences in physical computing, physical computing tools

1 Introduction

Within the last decades computers have become ubiquitous and interactive media. Embedded systems play an increasingly important role in our everyday-lives, but only few students ever get the chance to investigate and understand them, although this persistent trend can be used in computing education. Constructionist learning can be strengthened and inspiring learning environments offered, where students can be creative. One way to address these issues is to implement physical computing in computer science classrooms. Everyone likes making stuff do things – just think of marble slides, domino effects, Rube Goldberg machines or flying paper airplanes. With physical computing such childhood dreams can be met in the classroom while at the same time understanding embedded and ubiquitous media.

This article will give an overview of the defining characteristics of physical computing and describe typical products, processes and tools of the discipline. The question will be pursued, what key competences students gain in physical computing.

2 Three Pillars of Physical Computing

Physical computing is quite a new and unfamiliar concept among informatics teachers and informatics didactic researchers. Literature review has shown that different authors put different emphases on the following three pillars of physical computing: typical products, tools and processes of physical computing need to be investigated in order to clarify the meaning. It will not be possible to clearly draw lines between the three fields. Processes involve tools and aim at particular products, so different perspectives will therefore certainly put a stronger focus on any of the three variables, but not neglect the other two. As a starting point it is assumed that physical computing means to creatively design tangible interactive objects or systems using programmable hardware.

2.1 Products

The term 'physical computing' in educational settings was first mentioned by O'Sullivan and Igoe (2004), who see it as a crucial element of such systems that they make use of transducers (sensors and actuators) to connect the virtual and the physical world. Typical products of physical computing are programmed tangible media, such as the example illustrated in Figure 1. Such media can be embedded, interactive, responsive, adaptive and many more. One feature all products of physical computing have in common is that they are not transformational. Conventional computing systems (e.g. text translation or batch processing) do not allow interference in the processing as they strictly collect input data, then process these data and finish the program by presenting the result. Physical computing devices run continuously and interact steadily with the environment. Siemers (2012) distinguishes interactive from reactive systems depending on the communication driver. In the following both are referred to as interactive objects:

> Interactive Objects are programmed, tangible media containing an integrated system that is invisible to the outside world. They perceive their

> environment with sensors, which in turn deliver data to be processed by the integrated system. According to the configuration of the system these data are passed on to actuators. Interactive objects can be part of networks of interactive installations. (cf. Przybylla, Romeike, 2012)

Examples for such interactive objects and installations range from interactive jewelry and clothes over intelligent toy pets and mood lamps to room-filling installation arts.

2.2 Processes

There are many different contexts in which physical computing is practiced and therefore also a variety of purposes that are pursued. O'Sullivan & Igoe (2004) strengthen the role of the physical body in computing. They encourage makers to forget what they know about computers when planning a new project and instead to focus on the needs of people and the environment that are to be supported by computers. Others have adapted physical computing and use it in a wider meaning: they see it as connecting computers to the physical world (e.g. Computer Laboratory, 2013; Libow Martinez, Stager, 2013). This shifts the focus from interaction of machines with humans (only) to interaction between machines and the physical world in general. Among artists and designers, physical computing is characterized by the use of electronics to prototype new materials. Another aspect is the process of tinkering: reusing and improving existing hard- and software in an experimental way, driven by curiosity, imagination and creativity is part of the process (Banzi, 2011). This is also reflected in many projects, where people have used existing toys or devices to make something new. Based on the suggestions of O'Sullivan and Igoe (2004), the common approach of physical computing can be split into two parts:

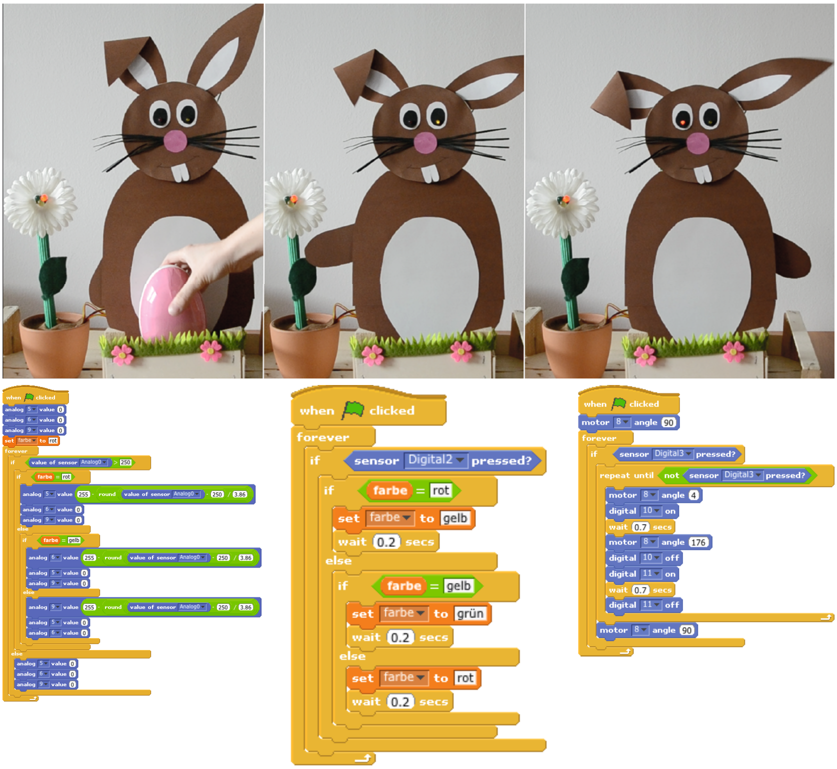


Figure 1: Vigilant Easter Bunny made with My Interactive Garden and S4A

- Description of what is supposed to happen from the point of view of the person who is going to experience it (e.g. “The thief who wants to steal the Easter egg will be chased away from the bunny, which makes noise, moves its arms ...”)
- Description of how this is supposed to happen (e.g. “When the thief takes the egg, this is noticed with a digital infrared sensor placed in the nest...”)

Physical computing thus focuses on ideas, not on technical limitations. This way, whim, imagination and creativity are fostered.

2.3 Tools

By now there is a large variety of good and affordable hardware on the market, which can be used for physical computing. Obviously, physical computing always involves the use of sensors and actuators and a computer in-between to control the behavior. Apart from this restriction, anything is possible. It is therefore not surprising, that tools are available in big numbers for every purpose and target group. Different types of hardware for physical computing activities were classified technically, based on their level of complexity and additional features such as modularity (Figure 2). Five main groups of hardware tools have been identified: programmable toys, I/O devices, programmable bricks, microcontroller boards and mini computers. Depending on how advanced the makers are, they either assemble sensors and actuators themselves or use preassembled hardware. For the above-mentioned "Vigilant Easter Bunny" the physical computing toolkit "My Interactive Garden" consisting of an Arduino microcontroller and plug and play components and the programming environment Scratch for Arduino (S4A) reduce the complexity of using and programming the interactive object. In some primary school contexts, programmable toys and bricks were used to introduce students to algorithmic thinking. For some purposes (e.g. physical measurements) and first interactions input and output devices are helpful. Strictly speaking, the latter cannot be associated with physical computing, because the resulting products cannot be described as interactive objects in the sense of physical computing. Input and output devices, however, can serve as sensor and actuator boards and be connected to the integrated system. This way, they become relevant hardware tools. It is also conceivable that whole laptop or tablet computers are integrated into interactive objects to make use of displays, cameras, et cetera.

Difficulty	Type	Features		Examples
easy	programmable toys			Finch, Beebot, Big Track
	programmable bricks (modular)			LEGO WeDo, LEGO Mindstorms, Pico Cricket
	I/O devices	everything on board with modules		Phidges, Theremino, Senseboard Kit
		everything on board without modules	sensing	PicoBoard, HCIs (e.g. MakeyMakey)
			acting	-
			sensing and acting	SenseBoard
		nothing on board		Vellemann Board
	micro controller boards	with modules		MyIG, Tinkerkit, Hummingbird, Gadgeteer
		without modules		Arduino-Family, Wiring Board, E-Textiles
	mini computers	with modules		Arduino TRE / Intel Galileo + Arduino-compatible modules, Raspberry Pi + PiFace
complex		without modules		Raspberry Pi, Beagle Board, Intel Galileo

Figure 2: Classification of suitable tools for physical computing

2.4 Physical Computing in the Computer Science Classroom

Summarizing, physical computing is an activity that involves creative arts and design processes and that, by bringing together hard- and software components, connects the virtual world of computers to the physical world of humans. All hardware components used in physical computing make use of transducers to interact steadily with their environment. Typical tools used for physical computing include microcontrollers and minicomputers. Physical computing projects are of an iterative nature and quickly bring forth working prototypes. In every iteration ideas are always in focus, the implementation of the projects will start only after the ideas have matured (Figure 3). The artistic approach in physical computing matches perfectly with the ideas of constructionism: According to the constructionist learning theory, learning is most effective when learners construct knowledge and develop competences from their own initiative and for a personally relevant purpose (Papert, Harel, 1991). Resnick (1996) added: “What’s important is that they are actively engaged in creating something that is meaningful to themselves or to others around them”. Further, in all physical computing projects prototypes are created and iterative processes are cycled. Sometimes, existing systems are reused or expanded. In a constructionist sense, learners will hereby develop meaningful products they can present to and discuss with friends and family.

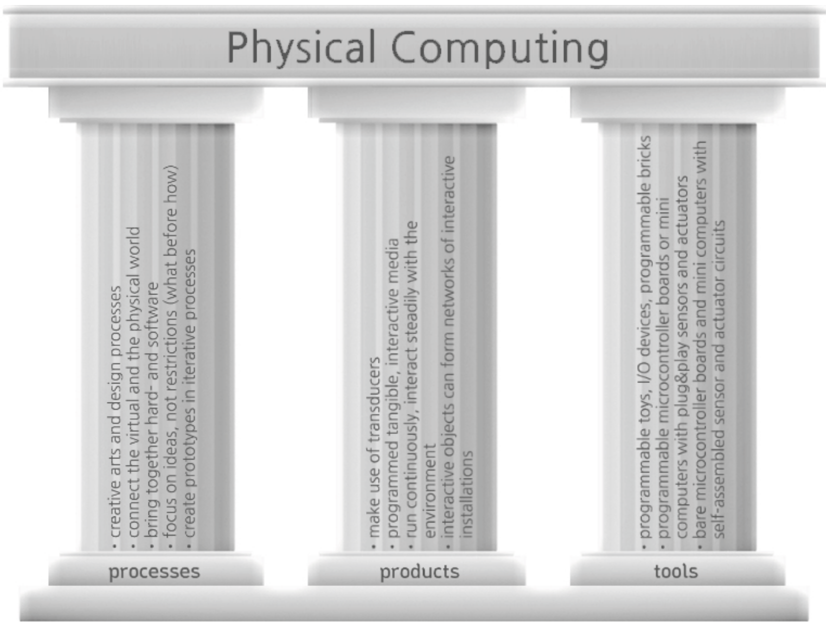


Figure 3: Three pillars of physical computing

3 Key Competences in Physical Computing

In Germany, throughout the last years there have been discussions about the introduction or abolishment of computer science as a compulsory subject in public schools. These discussions are often about the general value of the subject and key competences of which learners benefit in all areas of life as opposed to expert knowledge, which is not relevant for most pupils unless they decide to study computer science. In order to highlight these competences in the thematic area of physical computing, the "Operational Definition of Computational Thinking for K-12 education" (International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA), 2011), the "Uniform Test Requirements for Informatics" (KMK - Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland, 2004) (Standing Conference), the "Computer Science Education Standards" (Gesellschaft für Informatik (GI) e.V., 2008) as well as "A Model Curriculum for K-12 Computer Science" (Tucker et al., 2003) will serve as guidelines. In the following subsections, those areas are highlighted where physical computing makes an outstanding contribution.

3.1 Understanding Computing Systems

Interactive objects or installations made with physical computing are entire computing systems containing hard- and software components that students can assemble themselves and investigate further. Depending on the level of complexity they undergo in the particular setting, they can come all the way from an intuitive understanding (e.g. when controlling a programmable toy) to a deep understanding of interactive computing systems (e.g. when constructing an intelligent letterbox). Particularly aspects of hardware design help them to develop the abilities to identify and understand interactive systems in their every-day environments.

3.2 Formulating Problems

With physical computing, the basic ability to precisely formulate problems is formed and practiced as a first step in the process of designing and creating interactive objects. Students are required to unambiguously describe what is supposed to happen from an outside perspective, thus they focus on the problem formulation separately from thinking about possible ways of problem solving.

3.3 Organizing and Analyzing Data

In physical computing projects data can be collected automatically with self-made weather stations, a voting system to elect the next class representative or an automatic traffic recorder to count the number of cars passing outside school. This way, students learn with real-world data collected in their own environment by measurement objects they have designed and built themselves. They will find out about the coding and decoding of data and information while working with sensors that deliver data, which need to be interpreted and actuators that receive data, which has to be generated from information.

3.4 Algorithmic Thinking

Algorithmic thinking is also a crucial element of the physical computing process. Students at any level are required to precisely describe series of events, both serial and parallel. Physical computing in particular demands students to develop algorithms that allow their objects to run continuously and interact steadily with the environment.

3.5 Effectiveness and Efficiency

Key aspects of computational thinking include identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources. In physical computing projects, ineffective or ineffi-cient solutions are particularly evident. Interactive objects or installations should give immediate feedback, may include concurrent processes and should include intuitive interfaces. If they fail to meet the expectations, e.g. due to the choice of inappropriate sensors, excessive pauses or delayed responsiveness, this is immediately noticeable.

4 Conclusion and Discussion

Physical Computing is a complex activity that requires learners to be aware of hard- and software issues at the same time. On the introductory level (primary school), students may work with programmable toys or programmable bricks and drag & drop programming environments to learn the fundamentals of algorithmic thinking. Construction kits that have pre-assembled sensors and actuators to be plugged into a microcontroller either directly or with a shield allows older children to come to visible and tangible achievements very quickly. With advancing in physical computing both, on the hard- and the software side, students undergo learning processes that strengthen computational thinking and key competences that are necessary for all aspects of life. Physical computing can enrich future informatics classrooms with valuable competences that are focused in computer science education better than in many other subjects, as they are innate in the subject and not to be imposed artificially. Future research on this topic will therefore investigate how students of different age groups experience physical computing activities and what learning processes they undergo.

References

Banzi, M. (2011). *Getting Started with Arduino* (2nd Edition). Sebastopol, CA: O'Reilly Media/Make.

Computer Laboratory, University of Cambridge (2013). Physical Computing with Raspberry Pi. Retrieved October 24, 2013, from http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/robot/

Gesellschaft für Informatik (GI) e.V. (2008). Grundsätze und Standards für die Informatik in der Schule. Retrieved April 15, 2014, from http://www.sn.schule.de/~i-standard/docs/bildungsstandards_2008.pdf

International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA) (2011). Operational Definition of Computational Thinking. Retreived April 15, 2014, from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

KMK – Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland (2004). Einheitliche Prüfungsanforderungen Informatik, 1–71. Retreived April 15, 2014, from http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf

Libow Martinez, S., Stager, G. (2013). *Invent to Learn – Making, Tinkering, and Engineering in the Classroom. Torrance,* CA: Constructing Modern Knowledge Press.

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

Papert, S., Harel, I. (1991). Situating Constructionism. In S. Papert, I. Harel (Eds.), Constructionism. Norwood: Ablex Publishing Corporation. Retrieved from http://web.media.mit.edu/~calla/web_comunidad/Reading-En/situating_constructionism.pdf

Przybylla, M., Romeike, R. (2012). My Interactive Garden – A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In Kynigos, C., Clayson, J. E., Yiannoutsou, N. (Eds.), *Constructionism: Theory, Practice and Impact. Proceedings of Constructionism 2012* (pp. 395–404).

Resnick, M. (1996). Distributed Constructionism. In *ICLS '96 Proceedings of the 1996 international conference on Learning sciences* (pp. 280–284). International Society of the Learning Sciences.

Siemers, C. (2012). Handbuch Embedded Systems Engineering V 0.61a. Retrieved May 26, 2014, from http://www.in.tu-clausthal.de/uploads/media/Embedded_Systems_Engineering_Handbuch_V0_61a.pdf

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2003). A Model Curriculum for K–12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. (Tucker, A. Ed.). New York: ACM. Retrieved from http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf

Biographies

Mareen Przybylla is research associate and doctoral student at the professorship for Didactics of Computer Science at the University of Potsdam, Germany. Her main research interest is on physical computing in computer science education and its effects on students.

Ralf Romeike is the head of the Computing Education Research Group at the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany.

Music Technology and Computational Thinking: Young People displaying Competence

Nicholas Reynolds, Andrew Swainston, Faye Bendrups
Melbourne Graduate School of Education
The University of Melbourne, Australia
{nreyn, swaa}@unimelb.edu.au, fayebendrups@hotmail.com

Abstract: A project involving the composition of a number of pieces of music by public participants revealed levels of engagement with and mastery of complex music technologies by a number of secondary student volunteers. This paper reports briefly on some initial findings of that project and seeks to illuminate an understanding of computational thinking across the curriculum.

Keywords: Computational Thinking, Music Technology, ICT Competence, Young People

1 Introduction

This paper reports a project that was part of a one day birthday event in Melbourne. The activity, entitled Sound Escape, involved the creation of a 'crowd-composed' piece (or pieces) of music. This paper presents very early findings from the project and introduces two young people and presents their involvement with technology. It provides an opportunity to describe computational thinking in practice and to highlight the possible connection between computational thinking, music technology, artistic and creative pursuits, and young people. In a time when there is much concern about young people's ICT competency, their capacity for computational and critical thinking, their reported inability to engage for prolonged periods of time and, amongst other things, their role as ICT consumers rather than producers, this paper reports that in certain circumstances it is very apparent that young people (certainly those in this project) have the necessary skills and competencies of 21st Century citizens.

In the structuring of the event it was decided that while there was room for the ‘traditional’ approaches of compositional software and a piano keyboard, a range of music making sources that didn’t require specific musical technical skill would provide a richer and more inclusive experience. Consequently, a suite of products that included laptops with piano keyboards, DJ equipment, midi drum pad/patch controllers, and a large collection of loops (pre-recorded short excerpts of music (single instrument) or drum rhythms) was developed. The final list of equipment comprised:

- 10 Mac laptops (microphones included) with Garageband and Logic ProX software installed
- 4 Mixtrack Pro2 DJ controllers (with Serato multitrack software)
- 2 Maschine Mikro drum pad and midi controllers (with Maschine software)
- 10 headphones
- 10 midi keyboards

A consequence of using modern approaches and technologies was that these technologies were unfamiliar to both facilitators; they each understood the function and purpose of the technologies but had no real practical experience using them. In particular this applied to the Maschine and to the Mixcraft Pro. These technologies were chosen not only because they were state of the art but also because they are relatively inexpensive, a consideration when working to a strict budget. As it happened the student volunteers were also unfamiliar with some of the technologies. All had used GarageBand but only a few had used Logic Pro, three of the seven had used Mixcraft but none had used Maschine.

2 Computational Thinking and Computer Use

As long ago as 1980 Seymour Papert envisioned a world in which computers were the carriers “of cultural ‘germs’ or ‘seeds’ whose intellectual products will not need technological support once they take root in an actively growing mind” (Papert, 1993, p. 9). It is interesting in this case to note that the technological support had become part of the creative process and the intellectual product rather than just, as Papert thought, a way of supporting active minds. Perhaps this is a reflection of the development of technologies beyond even Seymour’s vision. Papert’s notion of technology as “objects to think with” (p. 11) is very much apparent in the use of technologies present at the Sound Escape event. While he was talking specifically about Logo Turtles as these objects, the parallels with modern technologies are clear.

In what might be seen as contradictory statements Wing (2006) defines computational thinking as "a way that humans, not computers think" yet calls for university subjects entitled "Ways to Think Like a Computer" (p. 35). In 2008, she clarifies this by talking about "mental tools" and "metal tools" (computers), where "the power of our 'mental' tools is amplified through the power of our 'metal' tools" (Wing, 2008, p. 3718) but here it is stressed that is the ability to think computationally (a human quality) is paramount in achieving outcomes not achievable without those metal tools. She does describe computational thinking as "a universally applicable attitude and skill set everyone, not just computer scientists, would be willing to learn and use" (Wing, 2006, p. 33).

We return to Papert now for more sage advice, which echoes (if it is possible to echo from the past) Wing's notion of universal application. He uses the term "think like a computer" quite freely but qualifies the term so that it does not mean to only or always think like a computer, rather as "a powerful addition to a person's stock of mental tools" (Papert, 1993, p. 155). When Papert asks himself to think like a computer, he does so knowing that "it does not close of other epistemologies. It simply opens new ways for approaching thinking" (p. 155).

Using descriptions of computational thinking by Wing and the National Research Council (*Report of a Workshop on The Scope and Nature of Computational Thinking*, 2010), Woltz et al. define computational thinking as "a mode of problem solving that emphasizes the processes necessary to express a computing-intensive solution in a structured, dynamic way" (Wolz, Stone, Pearson, Monisha Pulimood, Switzer, 2011). The NRC's definition is a little broader and reflects Wing's definition, saying that "computational thinking is a fundamental analytical skill that everyone, not just computer scientists, can use to help solve problems, design systems, and understand human behaviour" (*Report of a Workshop on The Scope and Nature of Computational Thinking*, pp. viii-ix).

In its most basic, but possibly its most universally accepted form, computational thinking requires a mindset or thinking approach that applies an understanding of the way computers work (think, act, function, are programmed) in order to solve complex contemporary problems. The actions and approaches of the young people in this study are real world examples of this kind of computational thinking; actions that occurred not in the world of computer science or programming but in the artistic and creative world of musical composition.

Music education has been challenged by researchers such as Lucy Green (Green, 2002, 2008) who through the investigation of the ways in which popular musicians learnt their craft proposed a way of learning she termed 'informal learning'. This is hardly a revolutionary term and it exists in many disciplines. What is relevant to this paper is the way Green describes some of the attributes of informal learning and the ways in which these expert musicians perceive their own skill development.

She talks about how knowledge is often 'discovered' rather than learnt through theory instruction. In one example she uses popular musicians' knowledge of chords and their harmonic properties. Quoting one of her colleagues she states:

> You discover A-augmented-6 because you want to play a Stevie Wonder song; you discover A-augmented-9 because you want to play a Jimi Hendrix song; you discover A-major triad over a B bass-note because you want to play a Carole King song (Charlie Ford in Green, 2008, p. 207).

This development of an understanding of complex harmonic structure without formal instruction can be seen as a way of thinking that fits with the ways in which the young people presented in this paper interacted with the complexities of the technologies and their ways of working musically and creatively.

3 Method

Data in the form of observational notes were collected throughout the day and analysed to produce a narrative of events. A photographic record of the day was kept and all computer files saved on the day were collected and stored safely on separate disks. More than 150 people participated in the activity. Many of those were parents who watched or assisted their children. The youngest participant was aged three with the oldest admitting to being "in her seventies".

In this paper observational field notes are used to present actions of the young people in context.

4 Complex Music Technology

Maschine is described by its manufacturer as a "compact groove production system", designed to allow the creation and editing of "grooves" (drum patterns and feels) that can be used in live performance or to enrich recorded performances. The software has the capacity to record live audio, called 'sampling, which can also be used in groove production. One typical use is to record a phrase or even a word and to splice it up so that segments of words can be treated rhythmically. For example the word 'yes might be represented as 'ye, ye, ye, yes'. The hardware itself consists of 16 touch pads that can be assigned a series of controls through controller buttons and knobs. It also contains an LCD multifunction screen and standard transport buttons for play, record and so on. The way that pads are linked to controls creates a complex set of multifunction possibilities.

This complexity is increased through its interaction with its software. The software offers a complex and complete set of tools for the editing, recording and mixing of sound. It can be driven by the Maschine hardware or by the computer.

As the feature set and possible combinations of pad to controller and software function increase, the complexity of the task at hand becomes apparent. The requirement for significant operator skill appears to be a given.

> J and T arrive together; they are early and ready to get started. They look around the space a little uncertain and are introduced to Author 1 and Author 3. It is clear that they are very interested in the Mixtrack and the Maschine. Author 1 asks about their expertise with this equipment (it is new to him and he has little idea of how it all actually works). The boys respond that they haven't worked with this gear and haven't seen the Maschine software before but that it shouldn't be a problem. They sit down and start working.

The approach they both used (they were working at different workstations located next to each other) was to click and see. They had familiarity with an earlier version of Mixtrack so that created very few problems but the Maschine hardware and software was completely new to them. First they attempted to work out what its primary purpose was. This was a purpose according to their needs not necessarily what the manufacturer stated. They decided that its primary purpose would be to work as a drum generator but were intrigued by its capacity to work with and create samples.

> J looks at the maschine with interest, he starts pressing buttons and moving knobs. These actions do not appear to be random, they have purpose. He has the software open on his screen and shifts his attention there. Author 1 interrupts him to ask how it is going on. He replies that

> it is really interesting how each of the pad buttons appear to be able to be assigned completely different functions.

The actions above occurred over a very short time period. In that time J had made an appraisal of the hardware and had correctly identified the complexities of the interface. This complexity did not daunt him, rather it excited him. He was keen to keep learning.

> While Author 1 is talking to J, he (J) discovers the sample recording capacity of the equipment. He records a brief section of the conversation between him and Author 1. He shows Author 1 what he has done and then proceeds, through the use of the hardware, to play with the sample, rubbing over sections to produce the broken speech described earlier. He is very excited about this. He is not immediately sure exactly how the recording took place – through the Maschine hardware or through the computer microphone – but he will work that out very quickly.

The serendipitous discovery of the ability to record samples was very interesting to J. He had no idea that the software would do that but as soon as he had realised that this was the case he commenced editing the recorded track. The question of 'how' appeared somewhat irrelevant to him. The software performed a function. That function made perfect sense to him and afforded him an opportunity for increased artistic flexibility.

> T is working next to J, he has not said much but has focused on working with samples that he has found. He has not asked any of the authors for advice or assistance. There are brief discussions between him and J as they show each other what they've done (these are not audible), but most of the focus is on independent exploration.

The rapid mastery of a highly complex and multifaceted piece of software and hardware is apparent in the actions of these two boys. This mastery is demonstrated when participants begin expressing interest in the technologies. A number of young children want to press the buttons and make sounds. They are also very attracted to the Mixcraft DJ devices. The two boys, J and T, became confident and competent users of this technology in a very short period of time. They achieved such mastery that they could also confidently and competently provide guidance and support for those participants who wished to compose on those technologies.

5 Conclusion

This short paper can only begin to report on the project. It is presented here as a way of highlighting the ways in which young people can, when interested, when presented with authentic tasks, and when left on their own to learn what they feel is necessary to learn, can demonstrate high levels of competence in dealing with complex and unseen technologies. Here we have examples (brief as they are) of young people deeply engaged in critical and creative thinking, who are solving problems and applying 21st Century skills in order to solve them.

References

Green, L. (2002). *How popular musicians learn: a way ahead for music*. Surrey, England: Ashgate.

Green, L. (2008). *Music, informal learning and the school: a new classroom pedagogy.* Surrey, England: Ashgate.

Papert, S. (1993). *Mindstorms: Children, computers and powerful ideas* (2nd ed.). New York: Basic Books.

Report of a Workshop on The Scope and Nature of Computational Thinking. (2010). The National Academies Press.

Wing, J. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33–35.

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 366*, 3717–3725.

Wolz, U., Stone, M., Pearson, K., Monisha Pulimood, S., Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education*, 11(2), 9:1–9:22.

Biography

Nick Reynolds (see picture) is a senior lecturer in ICT in Education at the University of Melbourne. He is the Australian representative to TC3 and in 2014 was awarded the IFIP Silver Core Award.

Andrew Swainston lectures in creative technologies and music at the University of Melbourne.

Faye Bendrups is a musician, composer and performer. Her research interests and practice include the works of Astor Piazzolla and modern Tango forms.

Think logarithmically!

Maciej M. Sysło[1], **Anna Beata Kwiatkowska**[2]
[1]Faculty of Mathematics and Computer Science
University of Wrocław
Joliot Curie 15
50-383 Wrocław, Poland
syslo@ii.uni.wroc.pl
[2]Nicolaus Copernicus University
Chopin 12/18
87-100 Toruń, Poland
{syslo, aba}@mat.umk.pl

Dedicated to **John Napier** on the occasion of
the 400th anniversary of inventing logarithm in his book
Mirifici Logarithmorum Canonis Descriptio

Abstract: We discuss here a number of algorithmic topics which we use in our teaching and in learning of mathematics and informatics to illustrate and document the power of logarithm in designing very efficient algorithms and computations – logarithmic thinking is one of the most important key competencies for solving real world practical problems. We demonstrate also how to introduce logarithm independently of mathematical formalism using a conceptual model for reducing a problem size by at least half. It is quite surprising that the idea, which leads to logarithm, is present in Euclid's algorithm described almost 2000 years before John Napier invented logarithm.

Keywords: Logarithm, binary search, binary representation, exponentiation, Euclid's algorithm, Fibonacci numbers, divide and conquer, complexity

1 Introduction

Logarithm is a very important operation and function in informatics, not only because 'logarithm' is an anagram of 'algorithm'. Today it is difficult to imagine how one could study and work in informatics not knowing this concept.

Logarithm is formally introduced in high schools as a mathematical concept and we demonstrate here how to introduce logarithm in informatics and in mathematics using conceptual models, which have a computing flavour and are related to practical applications.

We consider here five themes/questions:

- How many times do we have to read a page in a paper dictionary to find a word?
- How many bits does an integer occupy in a computer?
- How many multiplications are needed to calculate the value of the exponential function?
- How fast can we find the greatest common divisor of two numbers?
- How many steps does a divide-and-conquer algorithm need when applied to a problem of size *n*?

A common feature of the answers to these questions is that all of them touch logarithm, directly or indirectly, and moreover they contribute to better understanding this concept and its role in designing practical algorithms and computations.

In our presentation, as much as possible, we avoid to use any reference to logarithm as a mathematical concept. Then, after informal introduction of the logarithmic function, we define logarithm in an algorithmic (operational) way.

We consider **logarithmic thinking** in problem solving as one of the most important facets of algorithmic and computational thinking and as a key competence to master high school and academic informatics as well as ICT studies. We refer the reader to our books (Sysło, 1997, 1998) and papers (Sysło, Kwiatkowska, 2006, 2014) for detailed presentations of the topics discussed here.

2 Logarithm as a Mathematical Concept

Logarithm appears in the core curriculum of mathematics in high schools in Poland on both levels, basic and extended. On the basic level, students are expected to be able to apply formulas which involve logarithm of product, quotient, and power of numbers, and on the extended level – also the formula for changing the base of logarithm. The logarithmic function appears only on the extended level and students are expected to draw a graph of this function and to use this function in modelling some phenomena and also in some practical situations. However, no applications of logarithm in informatics are included in the mathematics curriculum.

The paper (Webb et al., 2011) reports on promoting student understanding of logarithm using the instructional design theory of Realistic Mathematics Education, based on a principle that engagement in mathematics for students should begin within a meaningful context. However, no informatics context is considered.

This situation motivated us to write a paper (Sysło, Kwiatkowska, 2006) on the contribution of informatics education to mathematics education in schools, where we discuss several problems of mathematical flavour included in the informatics curriculum which can contribute to better understanding and appreciating mathematical concepts and their use in computing. In this paper we restrict our attention to one of such concepts – logarithm.

Most of the concepts and topics discussed in this paper belong to discrete mathematics, known as mathematics of our times, mathematics of the computer era, mathematics of computing. We use the approach presented here in schools K-12 as well as in teaching at university level.

3 Find a Word, guess a Number

A paper telephone book consists of 1000 pages. Find the page, which contains the telephone number of Mr. Smith, checking the smallest number of pages. Searching for a right page, students discover a **binary search**, that is to keep splitting the remaining pages into two equal-size parts and to eliminate the part which does not contain Smith, until only one page remains, which should contain Smith's telephone number. Instead of a telephone book one can choose a dictionary.

The game of guessing an integer hidden in a given interval may serve the same purpose and can be used to activate the whole class. We encourage students to play several rounds of this game in pairs and to fill in a table such as seen in Tab. 1.

Table 1: A table for the results of the game to find a hidden number

Interval	Interval size	Hidden number	Number of questions asked	LOG	$\log_2$(Interval size)
[1, 80]	80	65	7	7	7
[51, 180]	130	100	7	8	8

Regardless of the hidden number, LOG is equal to the number of times the interval size is divided by 2 to obtain 1 (when an odd number is divided by 2 we take 'a bigger half'), for instance: 30, 15, 8, 4, 2, 1. The last column could be filled in later. For any game, numbers in the last three columns should be close to each other.

Related topics and questions discussed with students:

- The importance of order among the elements in a dictionary and in an interval: how many times do we have to read a page in a paper telephone book of 1000 pages to find the owner of the phone number 1234567?
- In the case of a dictionary, when we have to find a word, which begins with one of the initial letters in the alphabet, we usually try to find this word on initial pages – such a strategy is called an **interpolation search**. We ask students to find in the Internet more information about this type of search and its complexity.

4 Binary Representation – A size of a number

Table 2: Binary representation

n	***q***	***r***
23	11	1
11	5	1
5	2	1
2	1	0
1	0	1

Students usually know how to find a **binary representation** for a given non-negative integer number n – such a representation is generated in successive divisions of n and the resulting quotients by 2. They divide n by 2 and take the remainder r (0 or 1) as the least significant digit of the representation. Then, apply this procedure to the quotient q and continue as far as the quotient is nonzero. For $n = 23$ we get $(10111)_2$, as in Table 2.

Then we ask students, how many binary digits has a decimal number n in its binary representation or equivalently, how much space in the computer memory we need for storing n. To answer this question let us assume that n needs k bits (however at this point we do not know the value of k). To find k, now we first have to determine the smallest and the largest numbers which can be represented on exactly k bits. The largest number has all k bits equal 1, hence we have:

$$(111\ldots1)_2 = 2^{k-1} + 2^{k-2} + \ldots + 2^2 + 2^1 + 2^0 = 2^k - 1$$

It is easy to see that when we add 1 to this binary number we get 2^k, the next power of 2, hence we get the last equality above. On the other hand, the smallest integer, which needs k bits, has only 1 on its most significant position, therefore equals 2^{k-1}. Therefore we have the following inequalities:

$$2^{k-1} - 1 < n \leq 2^k - 1.$$

Now, adding 1 to all sides of these inequalities and taking $\log_2$ of all sides we get the inequalities:

$$k - 1 < \log_2 (n + 1) \leq k.$$

Since the number of bits k is an integer number, we have:

$$k = \lceil \log_2 (n + 1) \rceil,$$

where $\lceil x \rceil$ is the ceiling function and is equal to the smallest integer number greater than or equal to x.

We may conclude now that an integer number n occupies about $\log_2 n$ bits in a computer memory – this number is sometimes taken as the **size of *n*** in a computer. Moreover, since a binary search in an interval of size n corresponds to finding the binary representation of n, we conclude also, that the number of steps in a binary search in an interval of size n equals about $\log_2 n$.

Finally we may reverse our arguments and **define** $\log_2 n$ **algorithmically**:

> **Logarithm** $\log_2 n$ is equal to the number of steps in which, successive divisions of n and the resulting quotients by 2 lead to 1.

5 Logarithm

Now we have to convince students that the logarithmic function is very important in informatics – its importance lies in its rate of growth – although it tends to infinity with n going to infinity but it is incomparably slower function comparing with the linear growth of n. Table 3 is the best illustration of our words.

Table 3: Linear versus logarithmic growth

n	$\lceil \boldsymbol{log_2 n} \rceil$
1024	10
1 048 576	20
10^{10}	34
10^{100}	333
10^{300}	997

6 Exponentiation

Fast exponentiation x^n is a crucial step in many real-world computations, such as compound interest, and public key cryptography (e.g., RSA). Practical values of n, for instance in RSA, are really very big numbers having hundreds of digits. We first ask student to calculate, how long a PFLOPS super computer (it performs 10^{15} multiplications per second) will compute x^n for a 'small' exponent n consisting of 30 digits, e.g. $n = 123456789012345678901234567890$, using the 'school' method which depends on performing $n - 1$ multiplications. Using the Windows calculator students can easily find that it will take more than … 10^7 years.

Our task now is to direct students to a faster exponentiation, which, as in the previous two cases, reduces the exponent by half at each step. When the exponent is even, they quickly come up with the formula $x^2k = (xk)^2$ and when the exponent is odd we suggest to transform this case to the even case and they quickly find that $x^{2k+1} = (x^{2k})x$. Then they use these observations to find how to calculate x^{23}. Repeated application of these rules leads to the following calculations:

$$x^{23} = (x^{22})x = ((x^{11})^2)x = (((x^{10})x)^2)x = ((((x^5)^2)x)^2)x = (((((x^4)x)^2)x)^2)x = ((((((x^2)^2)x)^2)x)^2)$$

Therefore, to compute x^{23}, only 7 multiplications are needed, instead of 22.

Next task is to estimate how many multiplications are used by this algorithm for arbitrary n. We ask students to compare the binary representation of $n = 23 = (10111)_2$ with the order of multiplications, going from right to left. It becomes clear that except the left most position, each bit 1 corresponds to multiplication by x and each position corresponds to squaring. Therefore, the number of multiplications in computing x^n by the above algorithm is equal to the number of binary positions in the representation minus 1 plus the number of 1's in the representation minus 1. Since the length of the binary representation of n is about $\log_2 n$, the number of multiplications needed to calculate x^n is at most $2\log_2 n$.

Finally we can estimate how many multiplication performs the above algorithm for $n = 123456789012345678901234567890$. We have $2\log_2 n < 194$. It is tremendous achievement in complexity – it takes a moment to perform 194 multiplications instead of waiting 10^7 years to get the result. Table 3 shows that calculating x^n for n with hundreds of digits takes a few thousands of multiplications.

The exponentiation algorithm described above can be expressed as a recursive procedure, see (Sysło, Kwiatkowska, 2014) for further discussion:

$$x^n = \begin{cases} 1 & \text{for } n = 0 \\ (x^{n/2})^2 & \text{for } n \text{ – even} \\ (x^{n-1})x & \text{for } n \text{ – odd} \end{cases}$$

7 Euclid's algorithm

We begin this section with the main observation of this paper:

> Euclid was very close to invent logarithm, almost 2000 years before John Napier did it!

We first ask students to apply **Euclid's algorithm** to find the greatest common divisor GCD (n, m) of n and m ($n \geq m$), for instance for $n = 34$ and $m = 21$. The algorithm generates a sequence of remainders (in the third column of Table 4):

$$r_{-1}, r_0, r_1, r_2, \ldots, r_k$$

which begins with the given numbers $r_{-1} = n$, $r_0 = m$ and terminates when the remainder becomes equal 0, $r_k = 0$. The remainders are generated according to the following equations:

$$\begin{aligned} r_{-1} &= q_1 r_0 + r_1, && \text{where } 0 \leq r_1 < r_0 \\ r_0 &= q_2 r_1 + r_2, && \text{where } 0 \leq r_2 < r_1 \\ &\ldots \\ r_{k-2} &= q_k r_{k-1} + r_k, && \text{where } 0 \leq r_k < r_{k-1} \end{aligned}$$

and GCD(n, m) = r_{k-1}. The first equation corresponds to the first row in Table 4, and the last equation – to the last row. The quotients q_i and remainders r_i satisfy:

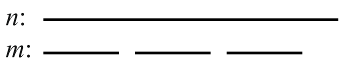
$$q_i = r_{i-2} \text{ div } r_{i-1}, \text{ and } \qquad r_i = r_{i-2} \text{ mod } r_{i-1}$$

Table 4: GCD

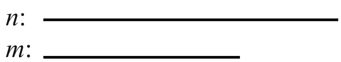
n	***m***	*ri*
34	**21**	13
21	13	8
13	8	5
8	5	3
5	3	2
3	2	1
2	1	0

Now we want to investigate with students how many steps needs Euclid's algorithm to find GCD(n, m). We suggest first to compare the numbers in columns 1 and 3 in Table 4. Students should notice that in the same row, the number in the third column is at least twice smaller than the number in the first column, that is, in the equation $r_i = r_{i-2}$ mod r_{i-1}, r_i, ri is at least twice smaller than r_{i-2}. Therefore we want to show, that in general the remainder r from dividing n by m is not greater than $n/2$. We usually provide a geometric proof of this property in which there are two cases.

A. $m \leq n/2$. In this case, when n is divided by m, then the remainder is not greater than m, which is at most $n/2$.



B. $m > n/2$. In this case, the remainder equals $n - m$ and since $m > n/2$, then $n - m$ is not greater than $n/2$.



Therefore in the sequence of numbers generated by Euclid's algorithm, each number is at least two times smaller than the number that appears two positions earlier. It reminds a sequence generated by a binary search except a sequence of the resulting numbers could be twice longer before it reaches 0 (the number smaller than 1). Hence we may conclude that:

> Euclid's algorithm finds GCD(n, m), where $n \geq m$ in'at most $2\log_2 n$ steps.

A challenging question for our students is to find n and m, for which Euclid's algorithm makes the largest number of steps. We have used such a pair above.

8 Fibonacci Numbers

The tendency of replacing a linear-time algorithm by a logarithmic-time algorithm, illustrated by the exponentiation, is also present e.g. in computing Fibonacci numbers. The recurrence relation defining Fibonacci numbers can be used to implement a linear-time algorithm. To obtain a logarithmic-time algorithm we have to use a system of two recurrence relations in which recursive calls have indices reduced by about half, see (Sysło, Kwiatkowska, 2014).

9 Divide and Conquer

The algorithmic methods discussed in this paper are based on divide-and-conquer technique in its broad sense – at each step of a method the problem size is reduced by at least half and there are a number of sub problems, which are to be solved on each level of the problem decomposition. In such situations complexity formula contains some logarithmic components and some linear terms. We usually illustrate such behaviour of divide and conquer using a merge sort together with its complexity analysis (for the problem size equal to a power of 2) see (Sysło, 1997).

10 Conclusions

In this paper we illustrate how we introduce logarithm, the most important concept in informatics, and show its properties and applications using a number of very popular building bricks of computer science. Logarithmic thinking is one of the most important key competencies when designing efficient solutions to real world problems.

References

Sysło, M. M. (1997). *Algorithms* (in Polish), WSiP, Warszawa.

Sysło, M. M. (1998). *Pyramids, Cones and Other Algorithmic Constructions* (in Polish). WSiP, Warszawa.

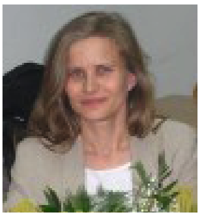
Sysło, M. M., Kwiatkowska, A. B. (2006). Contribution of Informatics Education to Mathematics Education in Schools. In Mittermeir, R.T. (Ed.) *ISSEP 2006*. LNCS, vol. 4226 (pp. 209–219). Springer, Heidelberg.

Sysło, M. M., Kwiatkowska, A. B. (2014). Introducing Students to Recursion: a Multi-facet and Multi-tool Approach, accepted for *ISSEP 2014* (Istanbul).

Webb, D. C., van der Kooij, H., Geist, M. R. (2011). Design research in the Netherlands: introducing logarithms using realistic mathematics education, *Journal of Mathematics Education at Teachers College*, Vol. 2, pp. 47–52.

Biographies

Maciej M. Sysło, mathematician and computer scientist, author of informatics curricula, educational software, textbooks and guidebooks for teachers, member of national committees on education, Polish representative to IFIP TC3, recipient of awards: Steinhaus, Car (Poland), Mombusho (Japan), Humboldt (Germany), Fulbright (USA), Best Practices in Education Award, IFIP OSA.



Anna Beata Kwiatkowska – academic and school teacher of informatics and mathematics, instructor at in-service courses for teachers, supervisor of school students gifted in informatics who participate in various competitions and in the Olympiad in Informatics, organizer of the Bebras Competition in Poland.

Using *Arduino*-Based Experiments to Integrate Computer Science Education and Natural Science

Holger Zieris, Herbert Gerstberger, Wolfgang Müller
Dept. of Physics Education, Dept. of Media Education and Visualization
Univ. of Education
Kirchplatz 2
88250 Weingarten, Germany
{zieris, gerstberger, muellerw}@ph-weingarten.de

Abstract: Current curricular trends require teachers in Baden-Wuerttemberg (Germany) to integrate Computer Science (CS) into traditional subjects, such as Physical Science. However, concrete guidelines are missing. To fill this gap, we outline an approach where a microcontroller is used to perform and evaluate measurements in the Physical Science classroom.
Using the open-source *Arduino* platform, we expect students to acquire and develop both CS and Physical Science competencies by using a self-programmed microcontroller. In addition to this combined development of competencies in Physical Science and CS, the subject matter will be embedded in suitable contexts and learning environments, such as weather and climate.

Keywords: Computer Science Education, Natural Science Education, Inquiry-based Learning, Physical Science, Measurement, Arduino, Sensors

1 Introduction

Studies on K12 education, such as PISA, have revealed widespread deficits in learning outcomes. These disappointments have led to a shift in objectives from content knowledge to skills and competencies. Learning scenarios that integrate various subjects have become increasingly important. More and more, school curricula are demanding interdisciplinary approaches to education. For example, in the German state of Baden-Wuerttemberg, at the level of secondary schools, Physical Science and CS Education are to be combined. Moreover, this combination will soon be cancelled and CS will be placed as a central theme in secondary schools. Central themes are themes that are not assigned to one specific subject; they have to be taught in an integrated manner.

Although current educational standards describe a general framework for interdisciplinary learning, they fail to supply concrete contents and methods. Consequently, teachers find it hard to integrate scientific subjects. This is especially true regarding a proper integration of learning objectives in the fields of Physical Science, Information Technology, and CS. In this paper, we address two consequences of this educational dilemma, specifically in the fields of Physical Science and Computer Science Education:

- The teaching of Physical Science in integrated science tends to be poor and fragmentary.
- Current and future educational standards require CS to be taught as an integral part of the established subjects (Ministerium für Kultus, Jugend und Sport, 2014). Some teachers simply subtract some teaching time from science to teach CS in a mostly isolated way. Moreover, course contents are mostly limited to the handling of application software such as Microsoft Word® and PowerPoint®. Higher-level competencies can hardly be achieved this way.

This paper outlines a new approach to integrating Physical Science and Computer Science. We suggest specific scenarios involving *Arduino* as a measurement tool. Furthermore, we propose competence areas to be promoted thereby.

2 Approach

What would be a possible approach to integrating CS and Physical Science education in a balanced manner? Science education in both fields has been broadly investigated (Coll, Taylor, 2008; Pientka, 2008). Following worldwide trends, the integration of Physical Science and Computer Science education has become a crucial element of integrated STEM (Science, Technology, Engineering, and Math) curricula (Berlin, Lee, 2005; Asghar et al., 2012). However, when integrated, Physical Science and Computer Science are mostly treated unequally:

In Physical Science lessons, computer-based technologies are typically used as mere tools to solve physics-specific problems. They are hardly used to increase CS competencies. The computer acts as a black box with several functions. For example, it simulates the lift in different liquids or helps students create models (such as the atomic models). Software is used as an interactive learning environment, for example to balance forces, or to collect data from ready-made sensors, delivering a well-formatted output.

In CS teaching, there has been some research in the area of robotics, for instance Lego Mindstorms®. Although these technologies have been very successful in teaching programming skills, their potential for teaching Physical Science concepts is low. After all, modern physics is more than the motion of a robot.

To avoid this unbalanced treatment of physics and CS, we suggest a Physical Science scenario where the design and application of CS instruments stimulates students to deal with both the informatics and the physical principles involved.

Specifically, we suggest using a microcontroller to record experimental data, handle the recorded data, and process them for presentation.

Our teaching method is based on three principles:

1. **Principle**: In order to go beyond mere knowledge toward application-centered skills, we suggest establishing a learning environment where students are responsible for most part of their learning process and outcome. **Theory**: Our approach builds on theories of problem-based learning and inquiry-based learning (Dewey, 1910). **Example**: The microcontroller is not treated as a black box. Instead, it has to be designed, constructed and programmed by the students themselves.
2. **Principle**: To promote a sense of purpose within the students, Physical Science is to be taught in a natural context and in a way that reflects the nature of science. **Theory**: Our approach draws on the concept of situated learning (Lave, Wenger, 1990) and learning in real-world contexts (Muckenfuss, 2006). **Example**: Within the context of weather, students may solve the question of how to acquire weather data. The use of *Arduino* allows students to observe and record processes in the real world.
3. **Principle**: The students themselves design, construct and perform computer-based experiments, guided by the instructor as necessary. **Theory**: In particular, we follow the idea of a guided-inquiry lab (Colburn, 2000). **Example**: Students would be asked to plan and build the experiments in a way that the sensors accurately measure the required data. Then they would write an *Arduino* program to

apply and convert the data into useable formats. These formats would allow the students to create a spread-sheet or other graphical forms for their presentation.

Because students' skills vary, individual scaffolding is important. Depending on the individual skill level of the students, the teacher can guide and support them by offering a set of ready-made elements for the construction and programming of the micro-controller.

3 Technological Approach: Measurements with Arduino

Media for schools should be reasonably cheap. The only way to have a dozen micro-controllers in a classroom is to use an electronic prototyping platform that is open-source, where no licenses have to be paid for. There is a large variety of relevant and interesting products on the market. The most common platforms are *Arduino* and *Raspberry Pi*. Both are single-board computers (the size of a credit card), with enough peripherals to connect the sensors to. Although *Raspberry Pi* is more focused on net-working and multimedia than *Arduino*, the latter is cheaper and more suitable for handling data. In contrast to other, less common platforms, *Arduino* comes with a large supply of accompanying material, such as tutorials, examples, and other resources. This encourages students to learn autonomously, possibly beyond what is expected by the teacher. A remarkable advantage of *Arduino* is the freedom in choosing the different sensors to get the data needed. Therefore, one can use sensors for experiments in all areas of Physical Science education.

Moreover, the *Arduino* integrated-development environment is specially designed to introduce newcomers to software development.

4 Scenarios

Using *Arduino* to analyze Physical Science experiments is possible in almost all scenarios. For beginners it is recommended to confine the subject matter to a basic domain of thermodynamics or mechanics, and to a related context. As an example, we introduce a weather station as a scenario, see Figure 1. We also worked with scenarios where measuring g-forces or measuring temperature in four different units are the central tasks.

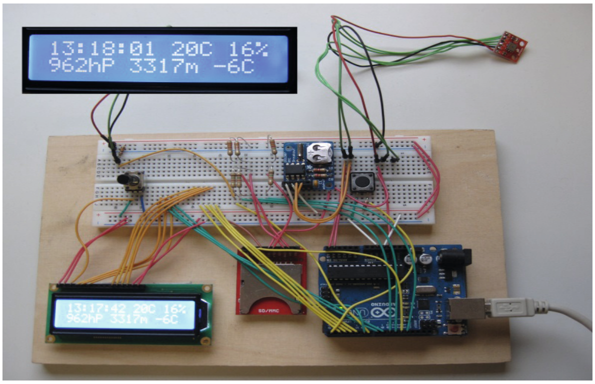


Figure 1: Weather station

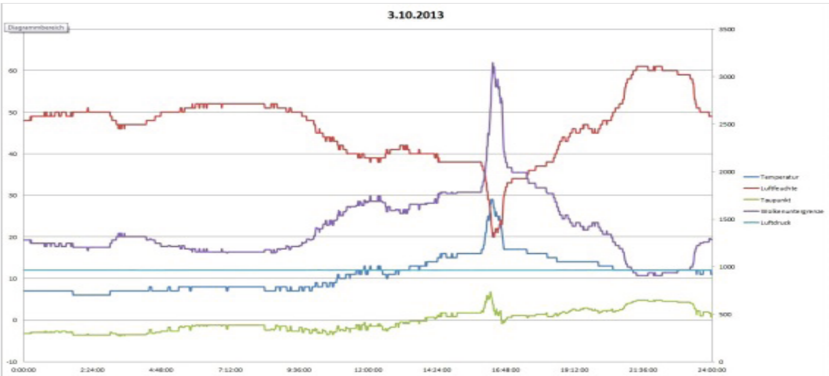


Figure 2: Weather diagram

4.1 Weather station

In meteorology (the science of weather), the main measurable variables are pressure, t3emperature, and humidity. For these three thermodynamic variables, there are numerous sensors available. Moreover, the timing of the measuring process need not be very fast. This makes it relatively easy to create a working code. Nonetheless, even simple temperature measurements, for example, require a large set of competencies in order to get correct values. Analogue sensor values, ranging from 0 to 1024, have to be mapped onto a suitable scale. Here, the students need to know (or find out) how a temperature scale is defined. Other thermodynamic variables can be calculated on the basis of the three main variables, and visualized as in Figure 2. The weather station may provide the actual values of temperature, humidity and pressure. Furthermore, as examples of calculated values, it can provide the cloud level and the dew point. Data can be visualized by a simple processing application created by the students.

5 Skills and Competencies

With the outlined approach, key competencies in both Physical Science and Computer Science will be addressed.Specifically, we aim to address some of the key competencies in the natural sciences, such as: Breaking down complex issues into simpler parts, planning experiments, collecting data, documenting and presenting experimental results and working autonomously with measurement systems. A more comprehensive list can be found in the German educational standards (Ministerium für Kultus, Jugend und Sport, 2004), or on the website of Next Generation Science Standards (Achieve, Inc., 2013).

Note that competencies in Physical Science and CS are to be trained with equal importance. For the purposes of KEYCIT, however, our presentation will be focused on the CS competencies. Not only do we aim to achieve the German media competence standards as formulated for Baden-Wuerttemberg (see above), but also the more comprehensive CS-related standards, *cf.* ACM K-12 (Tucker, 2003).

Overall, we address competencies in accord with the Guiding Ideas as put forward in Ministerium für Kultus, Jugend und Sport (2004):

Basic knowledge should be comprehensive, sustainable and experience-based. Therefore, teaching needs to be contextualized, taking into account students' pre-concepts, school equipment, and curricular organization.

It is a supreme goal of general school education to develop the ability to use information in a purposeful, responsible, and creative way. Important competencies in this respect are especially the sensible acquisition, choice, processing, and delivery of information.

Specifically, based on the German educational standards (Ministerium für Kultus, Jugend und Sport, 2004), we cover the following competencies and subject matters:

Table 1: Examples (right) for meeting curricular expectations (left)

Curricular requirements: The students are able to…	**Suggested realization: The students are able to…**
establish quality features of computer systems and software	specify necessary features of a weather station to get weather data that are as accurate as possible.
present the structure of a data processing system	describe and present the steps that are required to show sensor data
use the computer for measuring and controlling	connect sensors (for humidity, temperature and pressure) to a microcontroller to acquire weather data
solve a problem using a simple programmed algorithm	design a working code to display the value of the measured variable, such as temperature
handle the basic items of digital coding	translate analog into digital data when using analog sensors
use a wide range of basic IT-applications in an independent and purposeful way	use standard Microsoft Office ® applications to process and present the collected data
present data and facts vividly and clearly	create a weather diagram for the collected and the calculated data (*cf*. Fig. 2)

6 Summary and Outlook

Our research focuses on the various competencies that could be acquired and consolidated by using a self-programmed *Arduino* microcontroller in a Physical Science context. In addition to the competencies, we expect a positive effect on students' motivation and interest, both in CS and Physical Science.

In the upcoming two years, the first prototypes of an *Arduino* Measurement Box will be evaluated in close relation to live situations. This evaluation will take place both at college and high school level. The increase in Physical Science and Computer Science competencies will be measured and evaluated.

With our interdisciplinary teaching approach, we hope to foster key competencies in both Physical Science and Computer Science.

References

Achieve, Inc. (2013). Nature of Science. *Next Generation Science Standards*. Retrieved March 26, 2014, from http://www.nextgenscience.org/sites/ngss/files/Appendix%20H%20-%20The%20Nature%20of%20Science%20in%20the%20Next%20Generation%20Science%20Standards%204.15.13.pdf

Asghar, A., Ellington, R., Rice, E., Johnson, F., Prime, G. M. (2012). Supporting STEM Education in Secondary Science Contexts. *Interdisciplinary Journal of Problem-based Learning, 6*(2), 85–125. Retrieved March 01, 2014, from http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1349&context=ijpbl

Berlin, D. F., Lee, H. (2005). Integrating Science and Mathematics Education: Historical Analysis. *School Science and Mathematics, 105*(1), 15–24. Retrieved March 01, 2014, from http://dx.doi.org/10.1111/j.1949-8594.2005.tb18032.x

Colburn, A. (2000). An Inquiry Primer. *ScienceScope, 23*(6), 42–44. Retrieved March 03, 2014, from http://www.experientiallearning.ucdavis.edu/module2/el2-60-primer.pdf

Coll, R. K., Taylor, N. (2008). *Science Education in Context. An International Examination of the Influence of Context on Science Curricula Development and Implementation*. Rotterdam, Netherland: Sense Publishers.

Dewey, J. (1910). Science as Subject–Matter and as Method. *Science, 31*(787), 121–127. Retrieved March 01, 2014, from https://archive.org/details/jstor-1634781

Lave, J., Wenger, E. (1990). *Situated Learning: Legitimate Periperal Participation*. Cambridge, UK: Cambridge University Press.

Ministerium für Kultus, Jugend und Sport (Ed.). (2004). *Bildungsplan 2004 Realschule*. Retrieved March 15, 2014, from http://www.bildung-staerkt-menschen.de/service/downloads/Bildungsplaene/Realschule/Realschule_Bildungsplan_Gesamt.pdf

Ministerium für Kultus, Jugend und Sport (Ed.). (2014). *Bildungsplanreform 2015*. Retrieved March 15, 2014, from http://www.kultusportal-bw.de/,Lde/Startseite/schulebw/bildungsplanreform2015

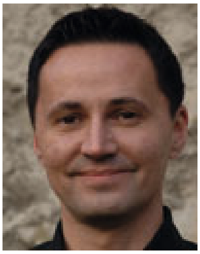
Muckenfuss, H. (2006). *Lernen im sinnstiftenden Kontext*. Berlin, Germany: Cornelsen.

Pientka, H. (2008). Lernen in Kontexten. *Praxis der Naturwissenschaften. Physik in der Schule, 57*(1), 4.

Tucker, A. (2003). *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee*. New York, USA: CSTA.

Biographies

Herbert Gerstberger is professor of physics and science education at the University of Education Weingarten. His research interests cover several aspects of interdisciplinarity connected to the teaching and learning of physics.



Wolfgang Müller is a professor for media education and visualization with the University of Education Weingarten. His research interests are in the areas of technology-based learning, edutainment, information visualization and visual analytics. He serves as a delegate to IFIP TC3 WG3.3, as a co-editor for chief, for the Springer Journal Transactions on Edutainment (ToE), and as an associate editor for the Journal on Technology for Education and Learning.

Holger Zieris is PhD candidate of physics education at the University of Education Weingarten. His research interests are in the areas of context based physical education such as flying, weather and energy.

Posters

Empirical and Normative Research on Fundamental Ideas of Embedded System Development

Steffen Büchner
Didactics of Informatics and E-Learning
University of Siegen
Höderlinstraße 3
57068 Siegen, Germany
steffen.buechner@uni-siegen.de

Keywords: Theory, Embedded Systems, Fundamental Ideas

Motivation

Embedded systems are seen as one of the main innovation drivers in the international computer science industry (Eggermont, 2002). Educating future experts is particulary difficult, because of the interdisciplinary nature of the development process, relying on competences from informatics, electrical engineering, physics, maths, and oftentimes in depth knowledge of the specific application area. Embedded systems correlate to terms like the internet of things, ubiquitous as well as pervasive computing, making the real impact on our every day life hard to grasp. Determining the fundamentals of embedded systems development may ease this task by providing students with a core set of competences that are useful in their professional careers, but in addition, can serve as a guideline to decide whether the student should investigate into a new technique later or not.

Schwill has described an approach to extract fundamental ideas for general computer science with a focus on school education (Schwill, 1997). The author describes a different approach which connects empirical and normative proceedings to determine fundamental ideas for embedded system developers. The DFG-funded research project *competence development with embedded micro- and nano systems* (KOMINA) provides empirical data for university teaching, derived from an experts survey and a laboratory course observation. These results have been published as an empirical refined competence structure model (ECSM) (Schäfer et al., 2012). This competence structure model is meaningful for many educators concerned with embedded system development, because it gives insights to the importance of the most common competence descriptions in this field of application.

The project members used these competence descriptions to exemplify the reconstruction of a typical hardware-focused laboratory at the University of the Siegen. The observation of this laboratory course is the second source of empirical data, which provides insights into common mistakes and detours of students when developing embedded systems.

The competence descriptions in the ECSM are sometimes application specific (FPGA development, usage of the C-Language) and other times very generic (interplay between hard- and software). This is problematic, because the competence descriptions offer plenty of room for interpretation, which restricts their use as a guideline for course creation. The inspection of the ECSM results with an approach like the fundamental ideas helps to create unified descriptions, useful for the specific audience and application area. In order to do that, modifications of the original criteria are unavoidable because Schwill's research methods have been created under different conditions (school education, computer science in general). Starting with Schwill's four criteria, the author has been researching proposed additions and own thoughts on how to separate fundamental from common ideas. The poster will give an overview of five criterions with their descriptions. Those are the advanced training criterion, the horizontal criterion, the criterion of time, the criterion of sense and the criterion of variance. The advanced training criterion is similar to the vertical criterion by Schwill. It, however, omits references to the spiral curriculum and instead focuses on basics needed for the students' career, justifying topics, which serve as groundwork for many other topics. The horizontal criterion has been adopted with minor, but very important changes. Due to the interdisciplinary nature of embedded systems development, fundamental ideas may be found at the borders of computer science and even other disciplines, too (e.g. electrical engineering or physics). While the criterion of time is without change, the criterion of objective conflicts with the authors understanding of the criterion of sense and is therefore dismissed.

The criterion of sense makes sure that an idea has a significant relevance in practice or in science, which is in contrast to Schwill's definition not the everyday life, but the life as a professional developer or researcher. In a first proof of concept where all criterions have been applied to a catalogue of techniques and methodologies currently discussed in science, the author noticed, that some fundamentals occur multiple times with only minor differences. The criterion of variance makes sure, that every potential fundamental idea connects existing ideas in a new and significant way or that it is a completely new idea by it self.

The current work encompasses the revision of Schwill's criteria with regard to higher education. A subject specific collection of ideas has been derived in a normative and empirical way. All criteria have been applied to this collection.

References

Schwill, A. (1997). Computer science education based on fundamental ideas. In B. Samways (Ed.), *Information Technology–Supporting change through teacher education. London: Chapman & Hall*, (pp. 285–291).

Schäfer, A., Brück, R., Büchner, S., Jaschke, S., Schubert, S., Fey, D., ... Schmidt, H. (2012). The empirically refined competence structure model for embedded micro-and nanosystems. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (pp. 57–62). ACM.

Eggermont, L. D. (2002). *Embedded systems roadmap 2002. Vision on Technology for the Future of PROGRESS*.

Biography

Steffen Büchner received his diploma in informatics (2011) from the University of Siegen, Germany. Since August 2011 he has been employed as a research assistant at the University of Siegen. He focuses on research in the field of competence oriented teaching for embedded system design and development.

On the Way to a "General Model of Contextualised Computer Science Education" – A Criteria-based Comparison of "Computer Science in Context" and the Concept of "Learning Fields"

Simone Opel
Didactics of Informatics
University of Duisburg-Essen
Schützenbahn 70
45127 Essen, Germany
simone.opel@uni-due.de

Extended Abstract: Vocational and general secondary computer science education in Germany pursue different aims – on the one hand, general secondary education targets to educate its students in order to attend either university or vocational education and training (with the consequence that also the learning content in computer science education should support general education of the students). On the other hand, vocational computer science education focuses on the development of skills and competencies for professional usage. For this reason, several teaching methods and approaches have been separately developed to encourage the students in gaining the competencies which are regarded as necessary with their school type. Although general secondary and vocational education often use different concepts, there is also some accordance. One such accordance can be seen in the usage of contextualised teaching methods. In general secondary computer science education the concept "Computer Science in Context" (CSiC) follows the idea of implementing contextualised teaching units by using contexts from the everyday life of the students (Koubek et al., 2009), whereas the concept of "Learning Field-orientated Computer Science Education" (LFCS) in vocational secondary education uses contexts from the professional life of the students (CMECA, 2011). These contexts should be implemented into activity-orientated lessons, called "learning situations" (Sloane, 2001).

Up to now, both approaches could be seen as promising to improve computer science education, but they have not been successfully and completely implemented into practice. To promote their implementation, it seems to be useful combining these different context-based approaches of secondary and vocational education by developing a *"General Model for Contextualised Computer Science Education"*, in order to have secondary, vocational and in prospective expansion also higher education benefit from each other.

As first step on the way to such a model, we compared in detail the concepts of CSiC and LFCS concerning their respective target groups and foundation in curriculum as well as underlying theoretical principles, competency models and superordinated aims. Therefore we established a set of criteria in an inductive way by analysing basic documents and descriptions of CSiC and LFCS. Afterwards, we selected the necessary criteria for the description of such a model.

As a result of this evaluation, we found accordances as well as differences between CSiC and LFCS. Both concepts are based on the idea of contextualisation as a way to promote the interest in and understanding of complex topics by the students. However, the target groups differ significantly – while CSiC has been developed for general secondary schools, the concept of LFCS is part of vocational computer science education and an obligatory part of the curriculum (ISB, 2007). For this reason, the basis for contextualisation is also quite different – CSiC uses contexts from everyday life and the social environment of the students, whereas LFCS uses contexts directly from their professional life. Another difference is the underlying competence model – CSiC is indirectly based on the cognition-theoretical competence model by Weinert (2001), whereas LFCS has been defined by the CMECA based on the action-theoretical, outcome-orientated model by Roth (1971).

Regarding these differences, LFCS seems to have a broader theoretical basis than CSiC. Therefore, LFCS could be the main basis for the theoretical framework of a *"General Model of Contextualised Computer Science Education"*. This prospective model can be described by different requirements: For a theoretical foundation of the model *relevant basic concepts of computer science* and *computer science education* have to be selected. Additionally, a suitable *competency model* – inclu-

ding the needs of general and vocational computer science education – has to be defined. These two parts of the model will be complemented by a *set of criteria for decision-making* based on whether a context idea is suitable to be implemented. The last – but not inherent – part consists of a *collection of guidelines* on how to implement a context idea into a teaching unit or learning situation. Since several models, standards and guidelines still exist, they have to be reviewed whether they could be suitable for the prospective model description. Our next steps will be to integrate the named requirements into a formal model description of a *"General Model of Contextualised Computer Science Education"* to promote contextualised teaching methods and facilitate the development of contextualised teaching units.

Keywords: Vocational Education, Secondary Education, Computer Science Education, Learning Fields, Contextualisation, Computer Science in Context, Activity-orientated Learning

References

CMECA – Standing Conference of Ministers of Education and Cultural Affairs (Sekretariat der ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland) (2011). *Handreichung für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe*. Bonn, Germany.

ISB – Staatsinstitut für Schulqualität und Bildungsforschung (2007). *Lehrplanrichtlinie für die Berufsschule. Fachklassen: Fachinformatiker/-in*.

Koubek, J., Schulte, C., Schulze, P., Witten, H. (2009). *Informatik im Kontext. Ein integratives Unterrichtskonzept für den Informatikunterricht (in German)*. In Körber, B. (Ed.), Proceedings of the 2009 German conference on Informatics and Schools (INFOS 2009), 268–279, Bonn: Köllen.

Roth, H. (1971). *Pädagogische Anthropologie*. Hannover: Schrödel.

Sloane, P. F. E. (2001). *Lernfelder als curriculare Vorgabe*. Hohengehren: Schneider.

Weinert, F. (2001). *Leistungsmessung in Schulen*. Weinheim: Beltz.

Biography

Simone Opel studied Information Technology at the University of Applied Sciences of Nuremberg and Vocational Education for Electrical Engineering and Computer Science at the University of Erlangen-Nuremberg. She worked as trainer for computer science and teacher at several vocational schools. Since 2010, she is working as a scientist in the "Didactics of Informatics" groups at the Universities of Erlangen-Nuremberg (until Oct. 2012) and Duisburg-Essen (since Nov. 2012).

Teaching Information Security (as Part of Key Competencies): The Situation in Austria

Thomas Schiller
BG/BRG Ramsauerstraße
Ramsauerstraße 94
4020 Linz, Austria
Pedagogical University of Upper Austria
Kaplanhofstraße 40
4020 Linz, Austria
th-schiller@gmx.net

Abstract: The poster and abstract describe the importance of teaching information security in school. After a short description of information security and important aspects, I will show, how information security fits into different guidelines or models for computer science educations and that it is therefore on of the key competencies. Afterwards I will present you a rough insight of teaching information security in Austria.

Keywords: Teaching information security, key competencies, computer science education, Austria

1 Information Security

Information security (definition in (Praxiom, 2013)) is very important, also in classroom, especially in times of heavy usage of smartphones and social networks. Do I have my (posted) data under control? Is it really my friend behind a certain account? There are further more questions to deal with, for example cyber mobbing as a possible consequence of easily taking snapshots of persons everywhere in any embarrassing and inconvenient situation and because of the "spatial distance" between offender and victim. Information security covers a wide range of (potential) problems, which cannot all be mentioned in detail here. In classroom it is also necessary to deal with technical basics to understand the used techniques like encryption and verification mechanisms.

Modern teaching approaches in every subject should be competence-oriented. According to Fuchs and Landerer (2005) important competencies in the field of computer science education could be: (C1) system competence, (C2) application competence, (C3) modelling expertise, (C4) communication skills, and (C5) problem-solving skills. System competence (C1) covers structure, function, limitations, safety and effects of (networked) computer science systems (ibid., p. 8). Information security belongs to system competence (including safety and effects of (networked) computer science systems (ibid., p. 8)). Students should also be able to deal with technical basics to understand the techniques behind a user interface of a system (e.g. encryption and authentication) as part of their application competence and communication skills. So, competence-oriented teaching in computer science should certainly emphasize aspects of information security.

2 Situation in Austria

At the AHS (allgemein bildende höhere Schule, Gymnasium, a wide spread secondary school type in Austria) computer science education is diverse because of decentralization and autonomy as possible reasons. In general, there are no obligatory computer science lessons in lower secondary education. Pupils in upper secondary education (ages 15 to 18) have two lessons weekly in 9th grade (age 15) that are obligatory. That is the only invariant in computer science education at a Gymnasium in Austria (Micheuz, 2009). Therefore, almost all relevant computer science topics have to be taught in this single course. Teaching information security should be included in this year, as mentioned (partially) in the curriculum (cf. BMUKK, 2003, p. 1). One of the objectives is to "understand key measures and legal principles related to data security, privacy and copyright, as well as learn about the impact of technology on individuals and society" (ibid., p. 2, translated by the author).

Troubles due to lack of awareness about information security are starting much earlier, long time before the compulsory computer science lessons begin. Therefore in Austria there exist different initiatives to raise awareness about information security, such as Saferinternet.at (Saferinternet.eu, co-founded by the European Commission (Saferinternet, 2013)) as well as the "Click&Check" workshops (Polizei, 2013).

Currently, models for digital literacy skills emerge in Austria (cf. EduGroup, 2013). In addition to these competence models, on (DigiKomp, 2013) a collection of ready-to-use teaching examples can be found, a part of them dealing with information security, which opens up the chance of bringing information security issues also to lessons of different subjects.

References

Web ressources were retrieved on 26/5/14.

BMUKK (2003). Lehrplan Informatik (2003), *Der nach wie vor gültige aktuelle Lehrplan ab 2003* (The still valid current curriculum from 2003), http://www.bmukk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf

DigiKomp (2013). *Digitale Kompetenzen – EduGroup*, http://www.digikomp.at

EduGroup (2013). digi.komp4/8/12 – Das Kompetenzmodell (Informatik 5. Klasse), http://www.edugroup.at/praxis/portale/digitale-kompetenzen/digikomp4-volksschule/kompetenzmodell.html, …/digikomp8nms-ahs-unterstufe/ kompetenzmo-dell.html, …/digikomp12ahs/kompetenzmodelle/informatik-5-klasse.html

Fuchs, K. and Landerer, C. (2005). *Das mühsame Ringen um ein Kompetenzmodell* (The struggle over a competency model), CD Austria 12/2005, pp. 6–9, infobild-web.pdf

Micheuz, P. (2009). *Zahlen, Daten und Fakten zum Informatikunterricht an den Gymnasien Österreichs* (Figures, data and facts about the computer science teaching in secondary schools in Austria), http://workspace.digikomp.at/pluginfile.php/30/mod_resource/content/0/Materialien_Ergebnisse/micheuz-infos-beitrag-2009.pdf

Polizei (2013). Click&Check (a project of the police) http://www.clickundcheck.at/

Praxiom (2013). ISO IEC 27000 2014 Information Security Definitions, Praxiom Research Group Limited, http://www.praxiom.com/iso-27000-definitions.htm

Saferinternet (2013). Saferinternet – *Sicherer Umgang mit dem Internet* (Safe use of the Internet), http://www.saferinternet.at/ and http://www.saferinternet.eu/

Biography

Thomas Schiller is teaching in the BG/BRG Ramsauerstraße (a secondary school) in Linz and at the Pedagogical University of Upper Austria and already taught at the Paris Lodron University in Salzburg.

ProtoSense – Interactive Paper Prototyping with Multi-Touch Tables

Christian Wegner, Raphael Zender, Ulrike Lucke
University of Potsdam
August-Bebel-Str. 89
14482 Potsdam, Germany
{christian.wegner, raphael.zender, ulrike.lucke}@uni-potsdam.de

Keywords: Interface design, paper prototyping, NUI

1 Motivation

The design of user interfaces with the paper prototyping method allows software developers to identify the customer's design and workflow related requirements in an early and cost-efficient manner. The basic idea is to sketch ideas by paper work (e.g. by cutting-out and painting elements) to visualize a user interface design at a very early stage of the design process. The result is a prototype which is simple and whose development did not need much time. The advantages of paper prototyping are obvious: They are easy to use, allow extensive control over details of the design and encourage team design because many people can draw at the same time. The main disadvantage of those prototypes is that they are not executable (Szekely, 1994). Their workflow must be simulated to customers by the designer who changes between multiple paper views.

Inspired by paper prototyping with so called "low-fidelity" (Snyder, 2003), several digital interface builders allow the creation of digital (high-fidelity) prototypes that are rudimental executable. But, these solutions lack in the ease of use since only experienced programmers are able to use them in their whole functionality.

With the increasing availability of Natural User Interfaces (NUI), the borders between the physical world and IT systems disappear more and more. The ProtoSense system introduced in this poster is a NUI-based wireframe prototyping solution that allows the creation of simple and executable user interfaces without the skill of an experienced programmer.

2 The Protosense System

ProtoSense runs on a table-size display (Microsoft PixelSense) and allows teams the creation of simple paper prototypes in an intuitive manner. Elements can be easily placed by physical stamps and they can be arranged by hand gestures directly on the table. The resulting wireframes are executable and support developers in the presentation of their early work to clients. Figure 1 shows an example of a prototyping result created with the help of the ProtoSense tool.

Early evaluations with experienced interface developers already showed the practicability of this solution, but also the demand of more elements and degrees of freedom with ProtoSense. An evaluation with computer science students led to the finding that ProtoSense is overall helpful for learning paper prototyping in this target group, especially in lower semesters, but for now clearly behind the experience, quality and usability of original paper prototyping.

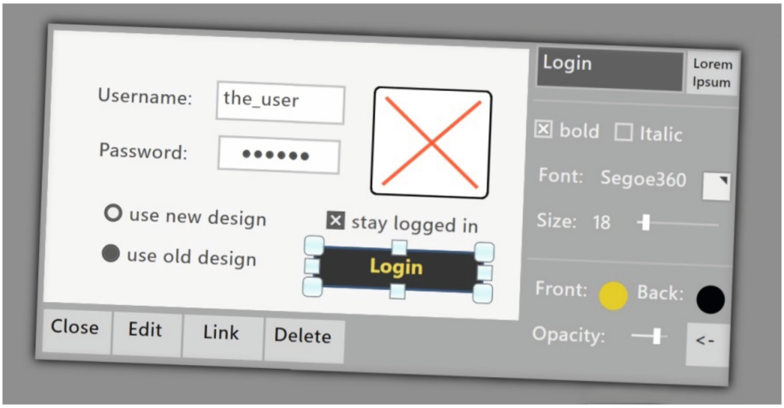


Figure 1: A sample prototyping project on the PixelSense table.

References

Snyder, C. (2003). Paper Prototyping. San Francisco, CA: Morgan Kaufmann.

Szekely, P. (1994). User Interface Prototyping: Tools and Techniques. In Ashlund, S., Henderson, A., Hollnagel, E., Mullet, K., White. T. (Eds.), *ICSE '94 Proceedings of the Workshop on Software Engineering and Human-Computer Interaction* (pp. 76–92). Amsterdam, The Netherlands: ACM.

Biographies

Christian Wegner completed his B.Sc. in the practical studies "IT Systems Engineering" at the HPI in Potsdam, Germany. Also, he studied "Design Thinking" at the d.school. In 2014, he achieved his M.Sc. in Computer Science.

Raphael Zender studied Computer Science at the University of Rostock, Germany. He wrote his Ph.D. thesis in Rostock and worked as scientific project assistant at the Chair of Computer Architecture. Today, he is scientific assistant at the Chair of Complex Multimedia Application Systems at the University of Potsdam, Germany.

Ulrike Lucke studied Computer Science at the University of Rostock, Germany. She finished her Ph.D. and habilitation on the development of IT infrastructures for research and education. Since 2010, she is Professor for Complex Multimedia Application Architectures as well as Chief Information Officer (CIO) at the University of Potsdam, Germany.

Workshops

Let's talk about CS! – Towards a suitable Classroom Language and Terminology of CS for Teaching

Ira Diethelm, Jörn Syrbe
Computer Science Education
Carl von Ossietzky University
26111 Oldenburg, Germany
{ira.diethelm, joern.syrbe}@uni-oldenburg.de

Abstract: To communicate about a science is the most important key competence in education for any science. Without communication we cannot teach, so teachers should reflect about the language they use in class properly. But the language students and teachers use to communicate about their CS courses is very heterogeneous, inconsistent and deeply influenced by tool names. There is a big lack of research and discussion in CS education regarding the terminology and the role of concepts and tools in our science. We don't have a consistent set of terminology that we agree on to be helpful for learning our science. This makes it nearly impossible to do research on CS competencies as long as we have not agreed on the names we use to describe these. This workshop intends to provide room to fill with discussion and first ideas for future research in this field.

Keywords: Terminology, classroom language, CS concepts, competencies, tools

1 Motivation

In natural sciences there is a long tradition to think about the usage of everyday and special language in class and also there is well-developed area of research about their role in science education, e.g. (Rincke, 2010). They also have a long tradition to investigate students' and teachers' perceptions (Duit, 2007).

At conferences regarding CS education and ICT for teaching there are an uncountable number of papers focusing on the use of tools for CS or other subjects. The number of papers that discuss the relationship between these tools, the basic concepts of CSE they are built for and the intended learning outcomes are much less. In tool-papers the authors often don't distinguish between tool and concept at all, for example it is often not clear if an "introduction to java" means to introduce into object oriented programming and algorithms or special aspects of the programming language java. Actually, the meanings of the terminology for our science Computer Science/Informatics/ICT are not yet resolved definitively.

One phenomenon that comes along with that is that the language students and teachers use to communicate about their CS courses is deeply influenced by tool names as well: e.g. "I had a Java course", "Last year we learned Greenfoot". In other disciplines this is unusual. Imagine our colleagues would hear from their students sentences like "I had a Casio Calculator Course" in math or in biology: "I learned about ph-testers and centrifuges". Sentences like these sound strange to us because calculators, ph-testers and centrifuges are clearly identified as tools and not as the learning objective. But in CS the distinction between tool and concept in learning objectives, intended competencies and related principles taught is not that easy. It depends much on the teacher's perspective on the certain course. And we are at least one century of tradition building discussion behind the natural sciences.

According to Ni and Guzdial (2012) CS teachers "have different perceptions related to CS teaching." They often feel not self-confident in teaching CS and in their choice of the topics and terminology to teach, even more if there is no teacher community to talk about CS in class available. Coming from different domains CS teachers it is coherent that they have a wide range and mixed terminology they use in class. Even in textbooks certain key terms like "algorithm" are frequently not defined clearly. Therefore, it results to be very difficult for teachers to decide upon suitable teaching material or literature for their own lifelong learning. It is even more difficult to judge on the value of certain teaching material for a given set of competencies if the terminology is floating ground. So the best teaching material provided is useless if the terminology used in the material does not fit to the teacher's one.

These fuzzy-terminologies of the CS field are an additional challenge for students, too. A reflected and unified special language during lessons and grads is indispensable for students to obtain competence and self-confidence in CS, especially for CS in general education and therefore to bridge the digital divide.

Many purposes will benefit from a unified classroom language for CS. It also would strengthen CS in class, its research and promote a better understanding of CS outside class. If we had a clear set of terminology for the use for teaching CS and ICT in classrooms many things would get much easier, for teachers, students and researchers.

2 Outline of the Workshop

This workshop does not provide a solution for this problem. Hence, we would like to lay the land of the problem domain "classroom language for CS" and discuss with other researchers from the field of CSE and CS teachers who came across the same difficulties during their practise recurring to the lack of a clear and accepted terminology for CS in class. We'd like to reflect about questions like these: What terminology do I use to describe my intended learning outcomes/competencies for my CS courses? What terminology do I use in class to introduce the key concepts of CS? What terminology do my students use in class to talk about CS and what terminology do I want them to use (instead)?

We'd like to conduct a small survey before the workshop to become aware of the different perceptions CS professionals and teachers could have about a small set of terms, first. The analysis of these data will be an exercise during the workshop and will serve as a starting point for discussing the questions stated above. Therefore the participants will be grouped by their mother language.

To shape the problem area we will make use of several perspectives and generate some hypotheses for further research. For the first set of perspectives we like to use parts of the approach of Educational Reconstruction. There, amongst others the students' and the teachers' perspective are taken into account and compared with the scientific view on the subject matter (Diethelm, Hubwieser, 2012). Discussing the intended competencies and the structuring of the courses it is necessary to reflect the different roles of ICT in class: what is used as a tool or as a learning environment (media) and what is the intended subject matter knowledge addressed? Planning to teach a special subject matter these questions should be answered in order to find suitable definitions of terms we would use in class.

With these perspectives we will try to create a first set of terms and visualize their relations to make differences between them transparent and ready to handle. These differences will occur comparing the perceptions of the participants and comparing them with definitions used in scientific publications and textbooks. They might also differ regarding the use by teachers and students in class during courses about the same topic. We expect that the meanings of these terms will not match entirely but will definitely overlap, but possibly with differing interpretations of each term related to the intended teaching contexts.

References

Diethelm, I., Hubwieser, P., Klaus, R. (2012). Students, Teachers and Phenomena: Educational Reconstruction for Computer Science Education. In McCartney, R., Laakso, M. J. (Eds), *12th Koli Calling conference on computing education research*. Tahko.

Duit, R. (2007). Science education research internationally: conceptions, research methods, domains of research. *Eurasia Journal of Mathematics, Science & Technology Education*, 3(1), 3–15.

Ni, L., Guzdial, M. (2012). Who AM I?: understanding high school computer science teachers' professional identity. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 499–504). ACM.

Rincke, K. (2010). Alltagssprache, Fachsprache und ihre besonderen Bedeutungen für das Lernen (Everyday and special language and their role in science education). *Zeitschrift für Didaktik der Naturwissenschaften*; 16, 2010.

Biographies

Ira Diethelm is full professor and leader of the research group for Computer Science Education at the Carl von Ossietzky University of Oldenburg, Germany. Most of her students are becoming CS teachers at high schools and vocational schools. She is also developing training programs for in-service teachers and teaching material.

Jörn Syrbe is research assistant and PhD student of Ira Diethelm at the University of Oldenburg, Germany. His research addresses the definition of a classroom terminology for CS Education and algorithms for automatic identification and classification of teaching material for CS at school.

Tackling Educational Challenges in a Digitally Networked World: Strategies developed from the EDUSummIT 2013

Kwok-Wing Lai[1], Niki Davis[2], Birgit Eickelmann[3], Ola Erstad[4], Petra Fisser[5], David Gibson[6], Ferial Khaddage[7], Gerald Knezek[8], Mary Webb[9]
[1]University of Otago; New Zealand
wing.lai@otago.ac.nz
[2]University of Canterbury, New Zealand
niki.davis@canterbury.ac.nz
[3]University of Paderborn, Germany
brigit.eickelmann@upb.de
[4]University of Oslo, Norway
ola.erstad@iped.ulo.no
[5]National Institute for Curriculum Development, The Netherlands
p.fisser@slo.nl
[6]Curtin University, Australia
david.c.gibson@curtin.edu.au
[7]Deakin University, Australia
fayekhaddage@gmail.com
[8]University of North Texas, USA
gknezek@gmail.com
[9]King's College London, UK
mary.webb@kcl.ac.uk

Keywords: Assessment, computational thinking, digitally-enabled pedagogies, educational systems; informal and formal learning, mobile learning, mobile technologies and apps, organisational evolution

1 Background

EDUSummIT (Education Summit on ICT in Education) is a global community of researchers, policy-makers, and educators committed to support the effective integration of ICT in education by promoting active dissemination and use of research. Supported by SITE, ISTE, Kennisnet, IFIP, ATE, and UNESCO, EDUSummIT has been held three times in the past, in The Hague (2009), Paris

(2011), and Washington D.C. (2013). These Summits have generated a large number of position papers, conference presentations, and journal articles. Information about the EDUSummIT is available at www.edusummit.nl.

The last EDUsummIT was held in Washington D.C. on Oct 1–2, 2013. Over 100 leading researchers, policy makers, and practitioners spent two days discussing educational challenges and strategies to address these challenges in eight working groups.

1. Towards new systems of schooling in the digital age
2. Advancing mobile learning across formal and informal contexts
3. Professional development for policy-makers, school leaders and teachers
4. Digital equity and intercultural education
5. Assessment as, for and of 21st century learning
6. Advancing computational thinking in 21st century learning
7. Observatories for researching the impact of IT in education
8. Placing Global Digital Citizenship and Literacy

A briefing paper for each group was published prior to the Summit and recommendations made by each working group to researcher, policy makers, and practitioners were published as summary reports and an action agenda (http://www.edusummit.nl/resources/results-edusummit-2013/) after the Summit.

In this workshop we will have four presentations to discuss major recommendations of the EDUSummIT 2013 by four working group leaders and members. We will also report on research conducted by the working groups subsequent to the EDUSummIT 2013, to be published in a special issue titled Research-Informed Strategies to address Educational Challenges in a Digitally Networked World, by the IFIP journal Education and Information Technologies in 2015. The next EDUSummIT will be held in Asia in 2015. During this workshop suggestions will also be sought with regard to the themes and format of this upcoming Summit. The presenters will make short presentations (10–15 minutes each) during the first part of the workshop, followed by at least 30 minutes of discussion with the audience.

2 Abstracts

2.1 Towards New Systems for Schooling in the Digital Age

B. Eickelmann, N. Davis and O. Erstad

The aim of this EDUSummIT 2013 Thematic Working Group was to identify the most effective policies and strategies to promote transformative and sustainable ICT-enabled changes in educational systems. Different perspectives on new systems of schooling in the digital age could be identified as relevant approaches. These perspectives are related to institutions, actors, and practices. Aiming for an expedient approach, Davis, Eickelmann & Zaka (2013) indicate the relevance of considering the co-evolution of pedagogy and technology. Because both education and digital technologies are evolving rapidly, the term co-evolution is adopted to describe the changing ICT applications and services as well as the changing scenarios leading to new systems and forms of schooling. Examples of new technology developments that could have an influence on new systems of schooling include OER (Open Educational Resources), MOOCs (Massive Open Online Courses), video-based learning settings, or flipped classrooms illustrate how the use of new technologies enables more flexible forms of teaching and learning as well as new systems of schooling. Furthermore, a need has been identified to move beyond traditional conceptions of formal vs. informal learning, online vs. offline activities, and to develop new conceptions of what defines learning spaces across different locations and contexts (Erstad, Sefton-Green, 2013; Fullan, 2012).

2.2 Towards a framework of criteria for identifying best practices and models of mobile learning

K.-W. Lai, F. Khaddage and G. Knezek

In this presentation we will discuss some of the key challenges and issues that teachers and students are facing today when using mobile devices in their classes, while high-lighting the urgency of identifying best practices, design guidelines, and models of mobile learning as a resource to support the design, development, and implementation of mobile learning in education. We will then propose a set of criteria as a framework for identifying best practices and design guidelines for integrating mobile technologies in learning. These criteria will include being evidence-based, culturally sensitive, curriculum centred, flexible and scalable, allowing adaptable pedagogy, student directed, and applicable in formal and informal contexts. Examples of best practices and design guidelines will be provided to illustrate how this framework can be used.

2.3 Technology enhanced assessment of collaborative learning

M. Webb and D. Gibson

This presentation examines the challenges and opportunities for improving assessment of collaborative learning through the use of technology. Our previous analysis of challenges for information technology supporting assessment (Webb, Gibson, Forkosh-Baruch, 2013), following discussions at EDUSummIT 2011, identified student involvement in assessment and digitally-enhanced assessment as critical for 21st century learning. Digitally-enhanced assessments were defined by the Working Group at EDUsummIT 2011 as those that integrate: 1) an authentic learning experience involving digital media with 2) embedded continuous unobtrusive measures of performance, learning and knowledge, which 3) creates a highly detailed (high resolution) data record which can be computationally analyzed and displayed so that 4) learners and teachers can immediately utilize the information to improve learning. This unobtrusive measuring approach provided a vision of "quiet assessment" whose volume can be turned up by learners and teachers whenever they wish in order to check their progress. There are now a number of projects working on developing a new generation of assessments including the OECD PISA Project which is planning to assess collaborative problem-solving skills in 2015 through computer-based assessment (see: http://atc21s.org/index.php/oecd-conceptual-framework-for-2015-pisa-assessment-of-problem-solving/). We will review recent developments in assessments and focusing particularly on approaches and challenges for assessing collaborative learning in order to identify:

1. Which current examples of computerised assessments embody our vision fully or partially?
2. For what purposes are computerised assessments particularly useful and where should other (non-computerised) approaches be retained or developed?
3. How can digitally enhanced assessments be designed to be transparent for teachers and learners?

2.4 Computational Thinking: A Conceptual Framework for Research, Teaching and Teacher Education

(Presenter: P. Fisser)

Computational Thinking has been receiving a great deal of attention lately – as being a particularly important skill that all students need to have to be successful in the future. Despite this there is much that we still do not know regarding the specifics of what the core concepts/attributes of CT are; how CT can be learned/taught; how CT can be integrated in the curriculum; and how the development of CT can be assessed/evaluated. In this presentation we offer an extended review of the idea of computational thinking, connect it to previous and current digital technology based educational initiatives (as well as point what differentiates it from the others). Most importantly through this we seek to develop a conceptual framework that would allow researchers, educators and policy makers to work together with a shared vocabulary. We end by identifying opportunities for both future research and practice.

References

Davis, N. E., Eickelmann, B., Zaka, P. (2013). A co-evolutionary perspective on the restructuring of schooling systems in the digital age. *Journal for Computer-Assisted Learning*, 29(5): 438–450. http://dx.doi.org/10.1111/jcal.12032 Special issue linked with EduSummIT.

Eickelmann, B. (2011). Supportive and hindering factors to a sustainable implementation of ICT in schools. *Journal for Educational Research Online*, 3(1), 75–103.

Eickelmann, B., Erstad, O. (2013). *Towards New Systems for Schooling in the digital age. Summary Report and Action Agenda. Results EduSummIT 2013*, Thematic Working Group 1

Erstad, O. (2013). *Digital Learning Lives. Trajectories, Literacies and Schooling*. New York: Peter Lang.

Hsi, S. (2007). Conceptualizing learning from the everyday activities of digital kids. *International Journal of Science Education, 29*, 1509–1529. Hsi, S. (2008). Information technologies for informal learning in museums and out-of-school settings. In Voogt, J., Knezek, G. (Eds.), *International handbook of information technology in primary and secondary education* (pp. 891– 900). Berlin, Germany: Springer.

Ito, M., Horst, H., Bittanti, M., Boyd, D., Herr-Stephenson, B., Lange, P., . . . Robinson, L. (2008). *Living and learning with new media: Summary of findings from the Digital Youth Project*. Retrieved from http://www.macfound.org

Kerka, S. (2000). *Incidental learning*. Trends and Issues Alert No. 18. Retrieved from http://www.ericacve.org/fulltext .asp

Khaddage, F., Baker, R., Knezek, G. (2012). If not now! When?A mobile badge reward system for K-12 teachers. In Resta, P. (Ed.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2012* (pp. 2900–2905). Chesapeake, VA: AACE.

Khaddage, F., Knezek, G. (2011). Device independent mobile applications for teaching and learning: Challenges, barriers and limitations. *Proceedings of Global Learn Asia Pacific* (pp. 1–7). Chesapeake, VA: AACE.

Knezek, G., Lai, K.-W., Khaddage, F., Baker, R. (2011). *TWG 2: Student technology experiences in formal and informal learning*. Retrieved from http://archief.vak-community.nl/edusummit/attachments/session%3dcloud_mmbase%2b2375683/2_EDUsummIT2011_student_technology_experiences_expanded_brief _paper.pdf

Khaddage, F., Lanham, E., Zhou, W. (2009). A mobile learning model for universities: Reblending the current learning environment. *International Journal of Interactive Mobile Technologies, 3*(1), 18–23.

Khaddage, F., Lattemann, C. (2013a). iTeach we learn via mobile apps 'a Case Study in a Business Course'. In McBride, R., Searson, M. (Eds.), *Proceedings of Society for*

Information Technology & Teacher Education International Conference 2013 (pp. 3225–3233). Chesapeake, VA: AACE.

Kukulska-Hulme, A., Sharples, M., Milrad, M., Arnedillo-Sanchez, I., Vavoula, G. (2009). Innovation in mobile learning. A European perspective. *International Journal of Mobile and Blended Learning, 1*(1), 13–35.

Lai, K.W., Khaddage, F., Knezek, G. (2013). Blending student technology experiences in formal and informal learning. *Journal of Computer Assisted Learning, 29*(5), 414–425.

Lai, K.W., Khaddage, F., Knezek, G. (2013). *Advancing mobile learning across formal and informal contexts*, International Summit on ICT in Education, October, 2013, Washington, D.C. Retrieved from: http://www.edusummit.nl/fileadmin/contentelementen/kennisnet/EDUSummIT/Documenten/2013/Pre-summit_brief_paper_TWG2_-Mobile_learning.pdf

Biographies

Kwok-Wing Lai is Professor of Education and Director of the Centre for Distance Education and Learning Technologies at the University of Otago College of Education in Dunedin, New Zealand. He is also the editor of *Computers in New Zealand Schools*. His current research focuses on the fields of knowledge building and mobile learning.

Niki Davis is the University of Canterbury Professor of e-Learning, IFIP outstanding service award; current member of WG 3.3 for research and WG3.6 for distance education. She recently developed a postgraduate course in Computer Science Education with Tim Bell, leading author of CS Unplugged and the CS Field Guide.

Birgit Eickelmann is professor for educational science with a focus on ICT in teaching and learning contexts, especially in schools and teacher education. She is IFIP member of WG 3.3.

Ola Erstad is professor at the Department of Education, University of Oslo, Norway. He has been working both within the fields of media and educational research, especially on the implementation and use of ICT inside and outside of schools.

Petra Fisser is involved in research and curriculum design at the National Institute for Curriculum Development in the Netherlands with a focus on technology integration in education.

David Gibson, Associate Professor of Teaching and Learning and Director of Learning Engagement at Curtin University in Perth, Australia. His research focuses on complex systems analysis, design and improvement of cyberlearning applications, games and simulations in education, and the use of technology to personalize learning via cognitive.

Ferial Khaddage is a Lecturer of Mobile Apps at Deakin University/MIBT Melbourne Australia. Holds a Ph.D. degree in I.T. (Mobile Technology) from Deakin University, her research interest is in mobile technologies and apps in education and has published widely in the area, and has presented her research findings locally and internationally.

Gerald Knezek is Regents Professor of Learning Technologies at the University of North Texas, in Denton, Texas, USA.

Mary Webb is Senior Lecturer in Information Technology in Education at King's College London and Chair of IFIP Working Group 3.3 on Research. Mary has developed and researched computing and the use of digital technologies in learning and teaching since computers first appeared in schools. She has taught Computer Science in secondary schools and runs a PGCE Programme for new Computer Science teachers.

Discussing Educational Standards for Digital Competence and/or Informatics Education at Lower Secondary Level

Peter Micheuz
Institute for Informatics Didactics
Alpen-Adria University Klagenfurt
Universitätsstraße 65-67
9020 Klagenfurt, Austria
peter.micheuz@aau.at

Abstract: Participants of this workshop will be confronted exemplarily with a considerable inconsistency of global Informatics education at lower secondary level. More importantly, they are invited to contribute actively on this issue in form of short case studies of their countries.
Until now, very few countries have been successful in implementing Informatics or Computing at primary and lower secondary level. The spectrum from digital literacy to informatics, particularly as a discipline in its own right, has not really achieved a breakthrough and seems to be underrepresented for these age groups. The goal of this workshop is not only to discuss the anamnesis and diagnosis of this fragmented field, but also to discuss and suggest viable forms of therapy in form of setting educational standards. Making visible good practices in some countries and comparing successful approaches are rewarding tasks for this workshop.
Discussing and defining common educational standards on a transcontinental level for the age group of 14 to 15 years old students in a readable, assessable and acceptable form should keep the participants of this workshop active beyond the limited time at the workshop.

Keywords: Educational Standards, Digital Competence, Informatics Education, Computing, Lower Secondary Level

1 Introduction

Since computers have been implemented in schools in the 1980s, the house of Informatics education, comprising the spectrum from digital media to computer science in its core played in the course of time an increasing but still ambiguous role. This applies especially to general education at secondary level including the primary level as well. Actually, more than 30 years after an unprecedented development of computers from primitive and programmed calculators to ubiquitous, pervasive and connected endpoints of networks, Informatics as the underlying science of our digital society is still not recognized widely as a core discipline in its own right. In contrast to traditional subjects such as (native) languages and Mathematics (a global language) as the common core and corner stones of educational systems worldwide, and other obligatory educational areas with elaborated curricula for each age group, Informatics (computing) education is underrepresented. The current situation of this fragmented field is due to an unclear terminology and a comparatively short history, not to mention the inherent inertia in educational systems.

However, an overview of worldwide endeavours gives hope that Informatics will play a more significant role in lower secondary education in a foreseeable future. Widely accepted definitions related to Informatics (computing), information and communication technologies, digital literacy and technology enhanced learning, and the acceptance of existing frameworks, competence models, curricula and teaching aids should support this process.

Recently, an increasing number of position papers, frameworks and country reports explicating the wide field of Informatics at schools have been published. These activities should remedy the unacceptable situation of big distortions of computing in education even within countries, regions and schools. Incoherence from country to country, state to state and even from school to school, is not the exception but the norm. Informatics education (standards) varies widely and its picture especially at lower secondary education shows distortions and inconsistencies referring to

- different perceptions of the term Informatics which often serves for every activity with computers,
- formal Informatics education between obligation and freedom of choice within autonomous decisions of schools and regions,
- an antagonistic view on approaches to develop students' digital competence and basic Informatics education, in an integrated way across the disciplines or as a discipline in its own right,

- different structures of reference frameworks in many countries, and
- different preconditions, cultural backgrounds and requirements worldwide.

2 Structure of the Workshop

2.1 Discussing Terminology

Among many meaningful combinations of relevant keywords, ranging from media literacy to a rigorous computer science education, digital competence and Informatics education seem to be prevalent. A short discussion about Informatics and its Anglo-Saxon equivalent computer science respectively computing should be conducted.

Table 1: List of Keyword Combinations

Field	Level of Proficiency
Digital, Media IT, ICT Computer, Computing Informatics, Computer Science	Skills, Literacy, Fitness, Fluency, Knowledge, Qualification, Competence, Pedagogy, Education

2.2 Overview of Frameworks

- The seminal European Reference Framework for Key Competences for Lifelong Learning (ERF, 2007) consists of the key competences communication in the mother tongue, communication in foreign languages, mathematical competence and basic competences in science and technology, digital competence, learning to learn, social and civic competences, sense of initiative and entrepreneurship and cultural awareness and expression.
- The DIGCOMP project (DIGCOMP, 2014), initiated by the EU commission with representants from many countries, published a framework for all citizens in our increasingly digitalised society. But has it the potential to serve as an important reference model like the prominent and influential Common European Framework for Foreign Languages? It comprises the main competence areas information, communication, content-creation, safety and problem solving with each area

consisting of 3 to 6 competences and the proficiency levels A (foundation), B (intermediate) and C (advanced).

- Further overviews of current models curricula and frameworks will be given and compared, e.g. the CSTA K-12 (CSTA, 2011) Curriculum comprising Computational Thinking, Collaboration, Computing Practice and Programming, Computer and Communications Devices, Community, Global, and Ethical Impacts,
- Principles and Standards for School Informatics in Germany (Gesellschaft für Informatik, 2008), comprising Information and Data, Algorithms, Languages and automata, Informatics systems, Informatics, man, and society,
- Digital Technologies within Australian curriculum development, comprising Digital Systems, Representation of data, Collecting, managing and analysing data, Creating solutions by Defining, Designing, Implementing, Evaluating and Collaborating and Managing (Australian Curriculum, Assessment and Reporting Authority (2014)).
- Other current approaches (cmp. new Computing curriculum in UK).

2.3 Discussing a Comprehensive Competence Model

As a common denominator of many regional, national and international curricula and frameworks, the following competence model can be seen as a starting point and compromise of core Informatics and interdisciplinary media education. It can be applied to nearly all stages of lower school level and can serve as a solid fundament and a preliminary basis for further Informatics education at upper secondary level.

Table 2: Reference Model for Digital Competence in Austria (Micheuz, 2010)

	Content	Levels of Competences		
		Knowing Understanding	Applying Designing	Reflecting Evaluating
Media Reflexion Related Topics	**Information Technology, Human and Society**			
	Impact of IT in Society			
	Responsibility in Using IT			
	Privacy and Data Security			
	Developments and Vocational Perspectives			
Digital Media Knowledge	**Informatics Systems**			
	Technical Components and their Use			
	Design and Use of Personal Information Systems			
	Data Exchange in Networks			
	Human-Machine Interface			
Use and Production of Digital Media	**Software Applications**			
	Documentation, Publication und Presentation			
	Calculation and Visualization			
	Search, Selection and Organisation of Information			
	Communication and Cooperation			
Principles and Computational Thinking	**Informatics Concepts**			
	Representation of Information			
	Structuring of Data			
	Automatization of Instructions			
	Coordination and Controlling of Processes			

3 From Abstract Frameworks to Concrete Tasks

Competence models play a well-defined and central role in the spectrum from abstract objectives to their implementation, leading to intended learning activities and students' outcomes. Typically, they are deduced from and refer to a core curriculum, and thus form the foundation for so called educational standards. However, a competence-oriented approach aims at concrete learning outcomes and has to be substantiated by age-appropriate and illustrating tasks. In order to make abstract formulations concrete, at the end of the workshop exemplary tasks will be presented for further discussion.

4 Requirements for this Workshop and Expectations

After a compact and comparing overview of approaches provided in a comprehensive way by the organizer of the workshop, additional contributions from the participants – who preferably should be aware of the situations in their countries – are appreciated. Preparational work and tentative results of this workshop could/should be the basis for further discussions among the participants beyond the limited time at the conference. This cooperative work could result in a widely accorded position paper about educational standards and (minimal) requirements at the end of K-8, preferably together with strong recommendations for a (possibly interdisciplinary oriented) subject (area) Informatics, Computing or Digital Technologies in its own right.

References

Australian Curriculum, Assessment and Reporting Authority (2014). *Digital Technologies Australia*. Retrieved from http://www.australiancurriculum.edu.au/technologies/rationale-aims/digital-technologies

CSTA (2011). *K-12 Computer Science Standards*. Retrieved from http://csta.acm.org/Curriculum/sub/K12Standards.html

DIGCOMP (2014). *A Framework for Developing and Understanding Digital Competence in Europe*. Retrieved from http://www.openeducationeuropa.eu/en/download/file/fid/35432

ERF (2007). *Key Competences for Lifelong Learning, European Reference Framework. European Commission Luxembourg*. Retrieved from http://www.britishcouncil.org/sites/britishcouncil.uk2/files/youth-in-action-keycomp-en.pdf

Gesellschaft für Informatik e.V. (2008). *Grundsätze und Standards für die Informatik in der Schule*. LOGIN-Verlag, Berlin, Arbeitskreis „Bildungsstandards" der GI.

Micheuz, P. (2011). *A Competence-Oriented Approach to Basic Informatics Education in Austria*, in Lecture Notes in Computer Science Volume 7013, pp. 43–55, Informatics in Schools. Springer: Heidelberg

Biography

Peter Micheuz is since 1979 an Austrian teacher at the Alpen-Adria-Gymnasium Völkermarkt and since 2000 in charge of teachers' education for Informatics at the Alpen-Adria-University Klagenfurt.

Tutorial

What about the Competencies of Educators in the New Era of Digital Education?

Carlos Delgado Kloos
Universidad Carlos III de Madrid
Av. Universidad, 30
E-28911 Leganés (Madrid)
cdk@it.uc3m.es

Abstract: A lot has been published about the competencies needed by students in the 21st century (Ravenscroft et al., 2012). However, equally important are the competencies needed by educators in the new era of digital education. We review the key competencies for educators in light of the new methods of teaching and learning proposed by Massive Open Online Courses (MOOCs) and their on-campus counterparts, Small Private Online Courses (SPOCs).

Keywords: Massive Open Online Courses, Small Private Online Courses, Competencies, Digital Education, Digital Revolution, Big Data

Summary

Salman Khan showed the world how it is possible to teach millions with just a video-capturing tool, a tablet, and some wit. His model was copied and extended by MOOC providers such as Udacity, Coursera, edX, and others. And this released a revolution in the education sector. Now there is of course much more than that: learning analytics tools, gamification features, animations and simulations to illustrate concepts, social tools for sharing questions and responses, etc. Moreover, similar technology has been used to improve on-campus education in the form of SPOCs (Small Private Online Courses). This new multimedia educational content can be used to complement lectures, help for remediation, expose top students to advanced content, and personalize the learning experience for all. And it has also given rise to new pedagogies, such as the Flipped Classroom model.

In a similar way that Information and Communication Technologies have disrupted the music industry, the news industry, and many other industries, these technologies are already disrupting educational practices. This means that blackboard and chalk, or projectors and slides are not the only tools for lecturing. And this implies that the educator has to acquire new competencies. Apart from being an expert in the domain, the educator has to master the new eco-system of knowledge transmission. This eco-system has many components. Here are some basic ones:

- creation of engaging multimedia material (media literacy),
- design of online quizzes with adequate hints,
- management of an online community of learners,
- definition of educational scenarios (Schuck, Aubusson, 2010) and useful learning designs (Mor et al., 2013).

Each of these components has many aspects for which educators have to be prepared.

Moreover, the possibility of teaching thousands of students at the same time through MOOCs comes at a time when digital data analytics tools are being applied with success in many industries. In less than a quarter of a century digital data has come from being 1 % of total data stored to become 99 % and in big quantities. Since technology allows processing large data sets in a reasonable time, and all the clicks can be captured when a student interacts with a digital platform, one can learn about how students learn and apply the best strategies for every learner individually. To be able to harness the findings of learning analytics tools to improve the teaching strategies is another relevant competency.

Does the faculty of the future need to be more like an actor/actress, become an engaging storyteller, have knowledge of video production, know how to prepare (possibly parametric) formative evaluations, be an expert community manager, and also a big data analyst, and all without losing the domain expertise? The objective of the presentation is not to give responses to all possible questions open at the moment with the digital revolution, but to pose some questions in order to encourage discussion.

It is clear that support personnel can take some of these roles. How this is done will depend to a large extent on what is possible in each individual institution. But this fact has another important implication: educators will not be so much an individual agent as they have been before, than a member of a larger team with whom they have to collaborate. The educators thus lose power and control over their teaching. Negotiation and collaboration skills will be additional assets of good educators.

Finally, the design of a MOOC or a SPOC is a multifaceted endeavour. There are multiple parameters to take into account to create a rich learning experience. It is not enough to master the different components independently. One also needs to have a good overview of the interplay of these different components (Alario-Hoyos et al., 2014).

The context of educators is changing in the Internet era. To be effective and efficient they need to master the tools that are available and be able to act in an environment of multiple stakeholders. How well they are able to adapt to the new context will determine the quality of their teaching, and, on the long run, the welfare of society.

References

Alario-Hoyos, C., Pérez-Sanagustín, M., Cormier, D., Delgado-Kloos, C. (2014). Proposal for a Conceptual Framework for Educators to Describe and Design MOOCs, *Journal of Universal Computer Science, 20* (1), 6–23.

Ravenscroft, A., Lindstaedt, S., Delgado Kloos, C., Hernández-Leo, D. (2012). *21st Century Learning for 21st Century Skills* (Vol. 7563). Springer, Berlin Heidelberg. http://www.springerlink.com/index/10.1007/978-3-642-33263-0

Schuck, S., Aubusson, P. (2010). Educational scenarios for digital futures, *Learning, Media and Technology, 35* (3), 293–305.

Mor, Y., Craft, B., Hernández-Leo, D. (2013). The art and science of learning design. *Editoral. Research in Learning Technology*, 21 article 22513.

Biography

Carlos Delgado Kloos is full Professor and the Holder of the UNESCO Chair about *"Scalable Digital Education for All"* at the *Universidad Carlos III de Madrid*. He is also *Vice-Rector for Infrastructures and Environment*, an as such, in charge of defining the online strategy for the university. He has led the development of online flipped-class 0-courses at the university that used the *Khan Academy* platform for freshmen to review STEM subjects. Under his leadership the first three MOOCs of the university were carried out, which were deployed through the *MiríadaX* platform. He directed and participated in one of the MOOCs himself. He is also the Director of a Master's program on e-learning and the Coordinator of the Excellence network *eMadrid* about Technology-Enhanced Learning in the Region of Madrid <www.emadridnet.org>. He has further been General and Program Chair of the EC-TEL conference about Technology-Enhanced Learning <www.ec-tel.eu>, and more recently the Chair of the Experience Track of the *eMOOCs2014* conference <www.emoocs2014.eu>. He is mem-ber of the German *Gesellschaft für Informatik*, board member of *TEL-Spain*, senior member of *IEEE*, editorial board member of the journals *MOOCs Forum*, Springer *Formal Aspects of Computing* and *IEEE*. *Transactions of Learning Technologies*, and the Spanish representative at *IFIP TC3*.

Previous Publications in this Series:

1 Schwill, A. (Hrsg.): Hochschuldidaktik der Informatik. HDI2008 – 3. Workshop des GI-Fachbereichs Ausbildung und Beruf / Didaktik der Informatik 2008
2009 | ISBN 978-3-940793-75-1

2 Stechert, P.: Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht
2009 | ISBN 978-3-86956-024-3

3 Freischlad, S.: Entwicklung und Erprobung des Didaktischen Systems Internetworking im Informatikunterricht
2010 | ISBN 978-3-86956-058-8

4 Engbring, D., Keil, R., Magenheim, J., Selke, H. (Hrsg.): HDI2010 – Tagungsband der 4. Fachtagung zur „Hochschuldidaktik Informatik“
2010 | ISBN 978-3-86956-100-4

5 Forbrig, P., Rick, D., Schmolitzky, A. (Hrsg.): HDI 2012 – Informatik für eine nachhaltige Zukunft
2013 | ISBN 978-3-86956-220-9

6 Diethelm, I., Arndt, J., Dünnebier, M., Syrbe, J. (Eds.): Informatics in Schools – Local Proceedings of the 6th International Conference ISSEP 2013 – Selected Papers
2013 | ISBN 978-3-86956-222-3

7 Brinda, T., Reynolds, N., Romeike, R., Schwill, A. (Eds.): KEYCIT 2014 – Key Competencies in Informatics and ICT
2014 | ISBN 978-3-86956-292-6

This series publishes conference proceedings and selected research reports in the field of computer science and informatics education from primary school to university level.

In our rapidly changing world it is increasingly important not only to be an expert in your chosen field of study but also to be able to respond to developments, master new approaches to solving problems, and fulfil changing requirements in the modern world and in the job market.

In response to these needs key competencies in understanding, developing and using new digital technologies are being brought into focus in school and university programmes.

The IFIP TC3 event Key Competencies in Informatics and ICT (KEYCIT 2014) provided a forum to present and to discuss research, case studies, positions, and national perspectives in this field.

The conference was organized into strands focusing on secondary education, university education and teacher education and accompanied by parallel conference streams of IFIP-WG3.4 and IFIP-WG3.7.

ISSN 1868-0844
ISBN 978-3-86956-292-6