A New Interactive Computer Science Textbook in Slovenia

Nataša Mori[1(✉)] and Matija Lokar[2]

[1] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
natasa.mori@fri.uni-lj.si
[2] Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
matija.lokar@fmf.uni-lj.si

Abstract. Informatics is only a mandatory course in the first year and elective in the remaining three years in Slovene general secondary schools (grades 9–12). The course curriculum lists 100 learning objectives for all four years, but it does not specify the ones for the first (mandatory) year. Although this gives Computer Science teachers the freedom to choose the topics to be covered, they usually choose the ones covering digital literacy. One of the reasons is that this is the most important topic in the eighteen-year-old textbook that is currently used to teach Computer Science. In this paper we discuss a new interactive textbook that was introduced and the areas of knowledge it covers. We also discuss e-textbooks as a technology in general and give some feedback from teachers after the first year of using the new textbook.

1 Introduction

For the last 18 years Computer Science (CS) in Slovene secondary schools (grades 9–12) has not changed much. We have a mandatory first year of Computer Science for all pupils, the next three years are optional, as well as the Matura exam in Informatics. The paper talks about the four-year-long Computer Science subject as a whole, because in Slovenia, we have a single curriculum. The curriculum represents the topics and learning objectives (100 of them!) of the Computer Science subject, spread over four years. This means that teachers can be very flexible at preparing each year's course. The curriculum has pretty much stayed the same since 1998, with minor updates made in 2008. Updating the curriculum would definitely be the proper step towards better CS, but in our case, it is not possible. Fortunately, the committee of National Examination Centre followed the ACM trend in improving CS, and they revised the national exam (Matura) in Informatics. A new Subject examination catalogue was published in 2013 for the 2015 Matura examinations. This led to concern about the current textbook - does it support the change? Unfortunately, the answer is no. The CS textbook was first published in 1997, centered mainly on technology and its use (*Digital Literacy*). *Information Technology* and *CS as a discipline*, as defined in [19], was mostly left out. It was big and heavy, as it was supposed to cover all

A. Brodnik and F. Tort (Eds.): ISSEP 2016, LNCS 9973, pp. 167–178, 2016.
DOI: 10.1007/978-3-319-46747-4_14

four years. It was then re-published every two years, with minimal corrections. In 2015, teachers used the textbook, which was last revised in 2008, and contains some ridiculous content. Therefore, we saw the need for a new CS textbook.

This paper describes the development of the new textbook and presents the teachers first impressions after using it. First there is the Introduction section, followed by the section where the challenge is described in more detail. The alternative to heavy textbooks is presented and the progress in CS Education outlined. In the third section the new textbook is described, explaining its technology and content. The fourth section represents the results of a survey among the teachers. The paper finishes with the conclusion and intentions for the future.

2 The Challenge

The main challenge was a new educational tool. Why do we need it and what kind? Should it be a textbook or educational software? While virtual learning environments seem efficient and motivational enough, we decided to create an interactive textbook. The first reason for the decision is a necessary leap from the existing textbook, which was the Holy Grail for the majority of the teachers and emphasized computing literacy mostly. The second reason is the effort teachers are willing to put into their teaching. We noticed, that teachers prefer to use a textbook than educational software, because it usually takes less time to prepare for the teaching. For instance, project TOMO (https://www.projekt-tomo.si/, [17]) is a Slovene online educational software for learning programming, where teachers could create their own courses, they could copy or insert programming tasks and much more. In three years time they only created four courses, which are all empty, though. On the other hand using new virtual environments, however, run the risk of taking the *e-textbook* to become a collection of digital items, missing the main essence of what constitutes a textbook, as a *defined unit of content* with a clear message [18].

There are some online interactive textbooks: payable like **TeenCoder** for AP Computer Science A course in Java (http://www.compuscholar.com/teencoder/teencoder_jv_series.php) and **Interactive Java** for elementary Java course (http://ijava.cs.umass.edu/index.html), free like **Computer Science Circles** for programming in Python (http://cscircles.cemc.uwaterloo.ca/) and even open source like **IMI Python** (http://imi.pmf.kg.ac.rs/imipython/) and **How to Think Like a Computer Scientist**, both for programming in Python (http://interactivepython.org/runestone/static/thinkcspy/index.html).

In this section the background of the textbook is explained and an alternative is offered. Then the development of CS through important documents worldwide is described, and the situation in Slovenia is shown.

2.1 E-textbooks

Before we look at some characteristics modern e-textbooks have or should have, let us recall briefly what a textbook is. In [11] a textbook is defined as a *part of*

methodologically-didactical materials, and it cooperates with the teacher in the education process and [20] emphasises that *The definition of a textbook depends on the nature of the school system. A textbook is one of the means that help the teacher and the student to achieve those goals.*

Besides the obvious additions and improvements to the *paper* textbook such as ease of access, lower weight and costs, speed of delivery, portability and ease of navigation [14], e-textbooks should provide content adapted to human to computer interaction with interactive elements, multimedia, instant feedback [9]. But this is not enough. An important aspect is missing. As the need for individual approach towards each student is becoming more and more accentuated, one of the crucial changes that is expected is that an e-textbook should allow for customization and personalization. Therefore, e-textbooks should be designed to be adaptable to the pedagogical situation and to the user, be it a learner or a teacher. It should be a given that an e-textbook allows for and enables uncomplicated customization and personalization.

What are the desired characteristics of a good e-textbook? According to [15,16] they should be:

- **Accessible:** an e-textbook should be available online and there should be the possibility of transferring it to other locations.
- **Adaptable:** an e-textbook should be adaptable to the needs of individual teachers, learners and groups of learners.
- **Cost effective:** an e-textbook should increase the efficiency and productivity by cutting the time and money spent on the whole lifecycle of a textbook, including future revisions, adaptations.
- **Durable:** an e-textbook should be adaptable to the changes in technology without costly redesign and re-encoding.
- **Interoperable:** an e-textbook should have the option of being used in different learning environments and with different tools. Poor examples of this feature are some existing e-textbooks that require the use of a specific type of interactive whiteboards.
- **Reusable:** an e-textbook should have the option to use its parts in different contexts. For instance a teacher can use an applet in a frontal type lecture, and a student who makes a certain mistake while solving an exercise, is directed to that same applet. The exercises can be used as homework or as part of an exam. But the major point of being reusable is to use parts of different e-textbooks to produce a customized version of the e-textbook.

Perhaps the most important aspect of a future e-textbook is its adaptability. An e-textbook must be adaptable to the needs of individual teachers, learners and groups of learners. There is no real reason why the textbook used in class 7a should be the same as the one in class 7b, or even within the same class, why Joes textbook should be exactly the same as Janes. Some steps have already been taken in this direction. Several publishers offer the possibility of changing the order of the chapters; skipping and adding topics or even changing the contents.

Adaptability is the core idea and the key feature separating future e-textbooks from their paper (as well as from digitally enhanced) versions.

Moreover, it is the role of the teachers to exploit this adaptability. The teachers are the ones who must adapt the e-textbooks to an actual teaching situation and to a particular student. The authors create e-textbooks having a particular ideal situation in mind. The teachers, however, teach in the real world. Therefore, e-textbooks should be flexible. They should enable the teachers to change and recombine various parts from various sources. Unfortunately, in the majority of the existing e-textbooks, this is a mostly unrealized goal, although advantages in technology made this goal possible. However we should clarify the role of teachers in this adaptations. As textbooks reflect the academic standards, specific objectives, and ideologies commonly found in public curricula the role of the authors is to provide various models according to the foreseen pedagogical situation and thus provide teachers choices [22]. Techers will pick the most appropriate version of the e-textbook and personalize/adapt it further.

2.2 Curricula

Although Computer Science is a relatively young science, its evolution was quite turbulent. From the theoretical Turing machine, through electronic general-purpose computer and programming languages, to personal computer and the goal to teach people how to use this new technology. Schools started to introduce Computer Science (CS) to students. Unfortunately, with the creation of new technology, it was necessary to learn how to use it. In most schools the CS course was slowly replaced by an ICT course, and in some cases that even escalated to a course in the use of office tools. Various researchers opposed the new trend and in the last 10 years CS concepts aim to find a way back to the curricula.

One of the first important documents about CS Education is probably IFIP's ICT curriculum in secondary education from 1994 (updated in 2000). While it basically introduces ICT literacy and basic skills into the school, it mentions creating and supporting of ICT, but is not intended for general education, but for professional education [4]. For the next decade schools mainly thought ICT literacy.

The first attempt to break this period was A Model Curriculum for K-12 Computer Science in 2003. Its main goal was to introduce the principles and methodologies of CS to all students. They specify the distinctions between CS and information technology, and recommends structure for K-12 curriculum [7]. For 5 more years, nothing much has changed until the breakthrough, which was almost simultaneous in 3 different continents.

In New Zealand prof. Tim Bell with his colleagues made a collection of free learning activities that teach CS through engaging games and puzzles that use cards, string, crayons and lots of running around, named *CS Unplugged* [3].

In the USA they published a national report *Running on Empty: The Failure to Teach K-12 Computer Science in Digital Age*. In the report they present findings about the poor situation of CS in US schools and list recommendations for improving CS education [23]. Soon after the report, CSTA published renewed

K-12 Computer Science Standards, where they emphasize the role of CS as a core discipline and restructured K-12 curriculum [8].

In the UK they decided on a more aggressive approach with the report called *Shut down or restart? The way forward for computing in UK schools*. The main findings of this comprehensive report include the unsatisfactory CS education, the role of CS as an academic discipline and poor qualifications. They thoroughly describe issues and recommendations, and suggest terminological reform ICT is divided to CS, Information Technology and digital literacy [19]. In the following years, a new compulsory subject Computing was introduced to all schools. With the help of Computing At School (CAS) organization, new materials for teachers and students were made, along with a new curriculum [5,6].

The latest project of renewing the CS begun in the end of 2015 by a group of various participants, from organizations ACM, CSTA and Code.org, to schools and technology companies. Their goal is to create a general framework for K12 CS, which would identify core concepts and practices. It is not meant to be a curriculum or standards document, but instead to provide the guidelines for designing a new curriculum, assessments or teacher preparation programs. In the beginning of 2016 they published *A Framework for K-12 CS Education*, a draft of framework, consisting of five core concepts and seven practices. The draft was published for open reviewing by anyone who wanted to support the development of the framework and help to improve it. The final version will be finished and published in the summer of 2016 [12].

In Slovenia we also came to a conclusion that the situation in schools is alarming and change needs to be done. The Government and the National Education Institute agreed to create a new (elective) Computer Science subject in grades 3, 4 and 5. The new subject covers topics from CS Unplugged. We still have an elective CS-like subjects in grades 6, 7 and 8, which remain the same, a mandatory CS subject in grade 9 (gymnasium) and an elective CS Subject in grades 10, 11 and 12. Overall we have one mandatory year of CS and nine optional years of CS in K-12 school system.

While the curriculum itself could not be revised, it is fortunately quite open. Its openness actually goes in two directions a positive and a negative one. The positive side of the curriculum is of course the fact, that teachers can include all the desired topics into the subject, with no set limits. They can be creative, they can introduce the students to physical computing, they can do all sort of things. Our curriculum is made for the whole secondary school (4 years) with no distinction between the years. A teacher can therefore decide what he/she will teach and when he/she will teach it. Unfortunately, in practice that means, that a teacher teaches "easier topics", such as the use of programs and/or computer in the mandatory subject in the first year, and leaves the important topics, such as algorithms, programming and networks for later (elective) courses. This means that only a small percentage of secondary school students get familiar with the proper content. Why there is such a small percentage of these students, is a story for another article. The first step toward a better subject was made with the publication of the Computer Science examination catalogue for Matura, which

included ACM's knowledge areas. However, teachers were struggling with the new content, because the old textbook did not cover all the topics. The next logical step was a new textbook, an interactive online textbook.

3 New E-textbook

We immediately decided that the form of the new textbook will be interactive, it will be online and free - quite the opposite from the old traditional one. We had the support of the ministry as well as the National Education Institute.

We faced the first challenge, where we had to decide which topics will be included in the textbook. We quickly saw, that all of the areas cannot be included in just one textbook, because that would be too much. Therefore we decided, that the four most important topics: programming and algorithms, systems, networks and distributed systems and informatics and society, need to be in the first part of the textbook.

We wanted to make a clear and concise textbook, with less text, but with comprehensive explanations. Therefore, the second challenge was to balance the explanations and the limited space in the textbook, and to include just the right amount of interactive elements [2].

The new interactive online textbook is meant for students, but can also be used by teachers. In the following sections we will explain the technology behind the textbook and its limitations, the importance of learning objectives from the curriculum and the grouping of areas.

3.1 Technology

The interactive textbook was created with a tool called *exeCute* (http://execute.fnm.uni-mb.si/). The tool is an adapted version of *eXeLearning tool* (http://exe-learning.org/), which originates in New Zealand. It is a freely available Open Source authoring application to assist teachers and academics in the publishing of web content. The tool was upgraded by a Slovene team into an *exeCute tool*. They used their own XML scheme that describes the content in a neutral XML format called E-learning object XML.

As seen in Fig. 1, it supports different kinds of interactive elements which can be made with JavaScript, some are even predefined:

- low-level interactivity elements, such as images, video, sound, animation, simulation (multimedia components);
- medium-level interactivity elements, such as various tests (true/false, multiple-choice questions, gap-fill);
- high-level interactivity elements, such as applets and educational games.

The tool itself did not support built-in code interpreter, therefore we needed to add it ourselves. With the help of *CodeMirror* (https://codemirror.net/) and *Skulpt* (http://www.skulpt.org/) libraries, we added an interactive and a non-interactive interpreter directly into the e-textbook, so pupils can code alongside

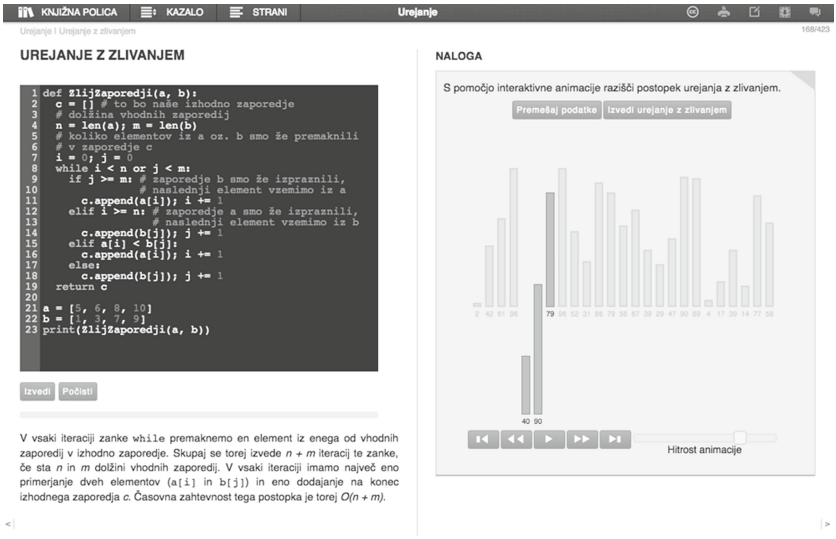

Fig. 1. Explanation of MergeSort, code and example of simulation.

the content or the exercise. *CodeMirror* serves as the base for the editor (line numbers, syntax highlighting, etc.), while *Skulpt* interpret Python code into the JavaScript code, execute it and return the result.

ExeCute has an index structured content covering each learning set that contains learning units - which are the base of the textbook. An e-learning unit consists of the title, the introduction (motivation and presentation of the content), the body (main content and interactive elements), the conclusion (summary of what the student learned and a set of exercises), and the sources [21].

A lot of interactive textbooks had already been made with the *ExeCute* tool by various authors. In Table 1 we describe the positive and the negative properties of the tool, based on our experience.

Moreover, the ExeCute technology supports modularization of learning objects and their grouping. Since it is XML based it is easy extendable by introducing new attributes. Due to lack of time we were not able to fully employ them, but the plan is to use additional attributes to support the personalization of the learning path and consequently make our e-textbook fully adaptable.

3.2 Grouping of Areas and Learning Outcomes

In Slovenia we have a CS curriculum for secondary school (gymnasium) from 1998, which was revised in 2008. There are merely 15 pages, although it covers all four years (from grade 9 to grade 12 in K12 system). The most interesting part is the distribution of topics with their 100 learning objectives. As mentioned in the beginning, the teacher can choose which topics he/she will teach in the one mandatory year, and which ones in the later optional years. Or in other words,

Table 1. Properties of the *ExeCute* tool.

PROS	CONS
- simple graphical user interface - easy to use - online support of the authors - nice design - predefined medium-level interactivity elements	- run only on Windows OS - poor documentation - slow, if you have a large e-learning unit - you can open only one e-learning unit per session - if you want to close and stop the program, you need to kill its process and Mozilla Firefox processes by hand - the prosess of inserting JavaScript applets is not straight-forward - complicated process of exporting the e-learning unit with all the included functionalities

which learning objectives should be presented to all students. The question is, how could one choose the most important topics and the most important learning objectives out of 100? In practice, the easier topics always win. Of course, the formal reason for choosing easier topics is much smarter - with the easier topics the teachers cover 54 % of all CS content, in one year. Sadly, 67 % of the easier topics are about the use of different programs.

Although we agree, that *digital literacy* is also important part of CS, this should not be the focus of the CS course. Computing is focused on three components: Digital literacy, ICT and Computer Science as a discipline. The last one is the least represented or even left out, so we will focus mainly on this component. A very good set of the knowledge areas in CS is presented by ACM and IEEE Computer Society [1]. After analysing other curricula and standards documents, authors of the e-textbook, including several university professors and secondary school CS teachers, incorporated programming and algorithms, systems, networks and distributed systems, and informatics and society into the new e-textbook. Throughout the e-textbook there is an emphasis on computational thinking.

The new e-textbook was published online (http://lusy.fri.uni-lj.si/ucbenik/) in the middle of 2015 and presented to all CS teachers in Slovenia. We were really satisfied with the result, especially when we found out that the content of the new e-textbook corresponds to the concepts of the new guidelines of A Framework for K-12 CS Education [12].

4 First Impressions

The new e-textbook was made and it was presented to most of Slovene CS teachers. We wanted to know, if the teachers use the e-textbook, how they use it, and what they think about it. We created a short online survey and received 61 replies. The complete questionnaire with analysis is available online (http://lusy.fri.uni-lj.si/ucbenik/survey2016).

The survey was divided into four sections - **Basic information**, **Content**, **Technology**, **Impact on learning process**, and **Comparison with the old textbook**.

Out of 61 replies, there were 35 (57,4 %) male and 26 (42,6 %) female teachers. 7 (11,5 %) of them were less than 30 years old, 14 (23 %) between 30 and 40, 20 (32,8 %) between 40 and 50, and 20 (32,8 %) more than 50 years old. 11 (18 %) teachers have just started teaching and have less than 5 years of experience, 6 (9,8 %) teachers have between 5 and 10 years of experience, 12 (19,7 %) teachers have between 10 and 15 years of experience, 14 (23 %) teachers have between 15 and 20 years of experience, and 18 (29,5 %) teachers have more than 20 years of experience. 5 (8,2 %) of them were not familiar with the e-textbook before, all others (91,8 %) were familiar with it.

We can see that the majority of replies was made from senior teachers, over 40 years old (65,6 %) and with more than 15 years of experience (52,5 %).

As expected, more than half - 34 (55,7 %) teachers teach at gymnasium, 18 (29,5 %) teach in primary school, 7 (11,5 %) teach in secondary professional schools and 2 (3,3 %) teach elsewhere.

The Content and Technology parts included three questions each; one question with a scale and two open questions. In the **Content** part, the question with a scale read *The content is explained well* and teachers could choose numbers between 1 (Strongly disagree) and 4 (Strongly agree). We chose the short Liker scale (4-point) in order to eliminate the *I do not know* answer. We wanted the teachers to decide whether they agreed or not. Most of them, 95,1 %, agreed that the content was explained well.

We asked the teachers which topics they considered important and felt they needed to be added. Most teachers missed topics about digital literacy (22 %), a lot of them suggested all the missing topics from the curriculum (20 %), only one teacher suggested the topic about physical computing and only one teacher suggested the topic about mobile applications.

We expected them to miss the digital literacy part from the curriculum, but did not expect so few different suggestions.

In the **Technology** part, the question with a scale read *E-textbook is technologically well made and has proper interactive elements*, where we chose the short Likert scale again. The majority agreed with the statement (93,5 %). The interactive element that the teachers liked most was animation (39 %), the second one was the built-in Python interpreter (28 %) and the third element was tasks with feedback (25 %). We got a few interesting answers to the open question if they would change anything from the aspect of technology. Two teachers suggested better support for mobile and tablet browsers, one teacher suggested a built-in search field, one teacher suggested support for adding notes. However, four teachers suggested more interactive elements, as the current version was too serious and boring, and one teacher suggested a printed version of the e-textbook, so that the students would not look into the display for too long.

In the **Impact on learning process** part, we first asked the teachers if they use the e-textbook in class, and then, based on the answer, provided sub questions.

Half of them, 31 (50,8 %), use the e-textbook, and 30 (49,2 %) of them do not use it. From those who use it in class the most common two descriptions of usage were exercises and revision. We grouped similar answers into five groups, as seen in Fig. 2. 19 % of the teachers use the e-textbook for explaining the concepts with interactive elements, 38 % of them use it for exercises and for revision of the content knowledge, 24 % of them improve their own learning plans and slides, 9,5 % use it for flipped learning, and 9,5 % for additional explanation. From those teachers, who don't use the e-textbook in class, 7 (23,3 %) of them said they didn't know about it yet but plan to use it in the future.

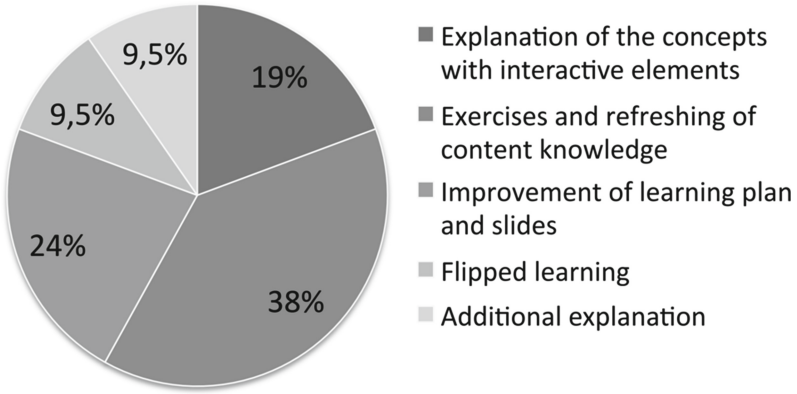

Fig. 2. Distribution of usage of the e-textbook.

The last part was a **Comparison with an old textbook**. 18 (30 %) teachers did not use any textbooks, they prefer to find information on the internet, 40 (66 %) teachers used the previously mentioned textbook and 3 (4 %) teachers used other textbooks. The majority of teachers - 51 (84 %) find the new e-textbook more useful, and the content better explained, although in the open question section they emphasize the lack of topics from the curriculum.

We can conclude, that the teachers liked the new e-textbook, both the content and the technology. However, they need to follow the curriculum and explain all the topics, which means the e-textbook is not yet complete.

5 Conclusion and the Future Steps

As teachers are mostly the ones who decide which textbooks are to be used in their classrooms, they are the ones who influence textbook development [13]. On the other hand, this also means that a careful choice of the approach used in a

textbook towards the topics prescribed by curriculum can realize some shifts in the way computer science is taught in Slovene schools.

According to the results of the preliminary study of the teachers' opinion described in this paper the authors of a new Slovene Computer Science e-textbook are on the right track. The topics covered are in accordance with the latest developments of curricula of computer science, the teachers prefer the technological solutions used with plenty of interactivity, especially in programming and algorithms.

However, the present version should be mostly seen as the first step towards a really modern e-textbook for Computer Science. The aspect most needed, and unfortunately missing in the current version, is adaptability. As argued in the introduction, e-textbooks should be designed to be adaptable to the pedagogical situation. There are several reasons why this goal has not yet been achieved in the current version of the Slovene Computer Science e-textbook. The most important is the fact that the Computer Science e-textbook was meant to be included in the Slovene portal of e-textbooks [10]. This brought some requirements and limitations of the layout as well as of the technological solutions that could be used. Due to some constraints the textbook described here was not actually included in the portal, so in the future versions some more advanced approaches in technological preparations will be used. Fortunately, as content and technological solutions currently used are based on HTML, the adaptations are possible. Besides adding adaptability we are planning to add support of other programming languages, not just Python.

References

1. ACM/IEEE-CS Joint Interim Review Task Force: Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science (2013). https://www.acm.org/education/CS2013-final-report.pdf. Accessed 4 May 2016
2. Anželj, G., Brank, J., Brodnik, A., Bulič, P., Ciglarič, M., Djukič, M., Fuerst, L., Kikelj, M., Krapež, A., Medvešek, H., Mori, N., Pančur, M., Sterle, P.: Computer science and informatics 1. e-textbook for informatics in gymnasium (2015). http://lusy.fri.uni-lj.si/ucbenik/. Accessed 4 May 2016
3. Bell, T., Alexander, J., Freeman, I., Grimley, M.: Computer science unplugged: school students doing real computing without computers. J. Appl. Comput. Inf. Technol. **13**(1), 20–29 (2009)
4. Bosler, U., Gumbo, S., Taylor, H., Wati Abas, Z., Duchteau, C., Morel, R., Waker, P.: Information and communication technology in secondary education. A Curriculum for Schools. UNESCO (1994)
5. Computing At School: Computing in the national curriculum. A guide for primary teachers, CAS (2013)
6. Computing At School: Computing in the national curriculum. A guide for secondary teachers, CAS (2014)
7. CSTA: a model curriculum for k-12 computer science (2003). http://www.acm.org/education/curricula-recommendations. Accessed 4 May 2016

8. CSTA: Csta k-12 computer science standards (2011). http://csta.acm.org/Curriculum/sub/K12Standards.html. Accessed 4 May 2016
9. Daniel, D., Woody, W.D.: E-textbooks at what cost? performance and use of electronic v. print texts. Comput. Educ. **62**, 18–23 (2013)
10. iUčbeniki: iučbeniki - spletno mesto interaktivnih učbenikov(itextbooks - portal of interactive textbooks) (2016). http://eucbeniki.sio.si/index.html. Accessed 4 May 2016
11. Jurman, B.: Kako narediti dober učbenik na podlagi antopološke vzgoje. Jutro, Ljubljana (1999)
12. K12CS.org: a framework for k-12 computer science education (2016). https://k12csdotorg.files.wordpress.com/2016/03/k-12-cs-framework-draft-march-18-2016.pdf. Accessed 4 May 2016
13. Knecht, P., Najvarov, V.: How do students rate textbooks? A review of research and ongoing challenges for textbook research and textbook production. J. Educ. Media Memory Soc. **2**(1), 1–16 (2010)
14. Lai, J., Chang, C.: User attitudes toward dedicated e-book readers for reading: the effects of convenience, compatibility and media richness. Online Inf. Rev. **35**(4), 558–580 (2011)
15. Lokar, M.: The future of e-textbooks. Int. J. Technol. Math. Educ. **22**(3), 101–106 (2015). Burnham: Research Information Ltd
16. Lokar, M.: E-textbook of the future. In: Proceedings of the Time 2014, R&E-SOURCE (2016). http://journal.ph-noe.ac.at/index.php/resource/article/view/135. Accessed 4 May 2016
17. Lokar, M., Pretnar, M.: A low overhead automated service for teaching programming. In: Proceedings of the 15th Koli Calling Conference on Computing Education Research, pp. 132–136. ACM, Koli Calling 15 (2015)
18. MindCET: The future of digital textbooks (2012). http://www.mindcet.org/wp-content/uploads/2012/10/Digital-Textbooks.-A-literature-review1.pdf. Accessed 24 July 2016
19. The Royal Society: Shut down or restart? The way forward for computing in uk schools (2012). http://royalsociety.org/education/policy/computing-in-schools/report. Accessed 4 May 2016
20. Turk Škraba, M.: Učbenik kot sredstvo za kakovostno učenje in poučevanje družboslovja. Ljubljana: diploma thesis (2005)
21. Čuk, A., Drakulič, D., Flogie, A., Jelen, S., Kaučič, B., Lipovec, A., Milekšič, V., Mohorčič, G., Novoselec, P., Pesek, I., Prnaver, K., Regvat, J., Repolusk, S., Senekovič, J., Šenveter, S., Vrtačnik, M., Zmazek, B., Zmazek, B., Zmazek, E., Wassermann, A.: Slovenian i-textbooks. The National Education Institute Slovenia, Ljubljana (2014)
22. Väljataga, T., Fiedler, S.H.D.: Going digital: literature review on E-textbooks. In: Zaphiris, P., Ioannou, A. (eds.) LCT 2014, Part I. LNCS, vol. 8523, pp. 138–148. Springer, Heidelberg (2014)
23. Wilson, C., Sudol, L.A., Stephenson, C., Stehlik, M.: Running on empty: the failure to teach k-12 computer science in digital age (2010). http://www.acm.org/runningonempty/. Accessed 4 May 2016