Metadata of the chapter that will be visualized in SpringerLink

Book Title	Proceedings of the Future Technologies Conference (FTC) 2018
Series Title	
Chapter Title	Some Barriers Regarding the Sustainability of Digital Technology for Long-Term Teaching
Copyright Year	2019
Copyright HolderName	Springer Nature Switzerland AG

Corresponding Author	Family Name	**Svetsky**
	Particle	
	Given Name	**Stefan**
	Prefix	
	Suffix	
	Role	
	Division	Faculty of Materials Science and Technology in Trnava
	Organization	Slovak University of Technology
	Address	Trnava, Slovakia
	Email	stefan.svetsky@stuba.sk
Author	Family Name	**Moravcik**
	Particle	
	Given Name	**Oliver**
	Prefix	
	Suffix	
	Role	
	Division	Faculty of Materials Science and Technology in Trnava
	Organization	Slovak University of Technology
	Address	Trnava, Slovakia
	Email	oliver.moravcik@stuba.sk

Abstract

Computer support of teaching is linked to terms like e-Learning, Technology-enhanced learning and Educational technology. Despite the very high level of the global IT services, networks, and actual clouds, which are also used in education, from a personalized teacher point of view, this is mostly only technological infrastructure that can be used for handling a curriculum content. A set of successfully used educational IT tools could be also mentioned, however, these are often only single-purposed static solutions. One needs not to do a research review – he can simply ask his colleagues – how many barriers are caused by incompatibility of the software formats, and short life cycle of software, hardware and networks solutions. Paradoxically, it is automatically supposed in the actual scientific papers that digital technology functions without any problems. Additionally, many questions arise regarding the IT sustainability for long-term teaching within engineering education. This paper describes some categories of the barriers to embedding the digital technology into teaching and shows some key points derived from around 12 years of the practical experience related to solving the personalized IT support of teachers within teaching bachelors students. As for the long-term teaching, it also demonstrates that state-of-the-art of the IT support for university teaching is not yet suitable including indicated lower IT skills of students. In real life, to be sustainable, a teacher should be teacher, programmer and researcher in one person. Regarding elimination of the barriers within long-term teaching, an all-in-one universal approach is presented by using the in-house educational software BIKE(E), based on design of a “virtual knowledge”. This specific default data structure enables solving knowledge (educational data) transmission through the off-line and online environments. In comparison with other solutions, it seems to be the most universal and sustainable solution for personalized IT support.

Keywords (separated by '-')	Computer support of teaching - Technology enhanced learning - Educational technology - Educational IT tools - Educational software - Digital technology - Personalized IT support

Some Barriers Regarding the Sustainability of Digital Technology for Long-Term Teaching

Stefan Svetsky(✉) and Oliver Moravcik

Faculty of Materials Science and Technology in Trnava, Slovak University of Technology, Trnava, Slovakia
{stefan.svetsky,oliver.moravcik}@stuba.sk

Abstract. Computer support of teaching is linked to terms like e-Learning, Technology-enhanced learning and Educational technology. Despite the very high level of the global IT services, networks, and actual clouds, which are also used in education, from a personalized teacher point of view, this is mostly only technological infrastructure that can be used for handling a curriculum content. A set of successfully used educational IT tools could be also mentioned, however, these are often only single-purposed static solutions. One needs not to do a research review – he can simply ask his colleagues – how many barriers are caused by incompatibility of the software formats, and short life cycle of software, hardware and networks solutions. Paradoxically, it is automatically supposed in the actual scientific papers that digital technology functions without any problems. Additionally, many questions arise regarding the IT sustainability for long-term teaching within engineering education. This paper describes some categories of the barriers to embedding the digital technology into teaching and shows some key points derived from around 12 years of the practical experience related to solving the personalized IT support of teachers within teaching bachelors students. As for the long-term teaching, it also demonstrates that state-of-the-art of the IT support for university teaching is not yet suitable including indicated lower IT skills of students. In real life, to be sustainable, a teacher should be teacher, programmer and researcher in one person. Regarding elimination of the barriers within long-term teaching, an all-in-one universal approach is presented by using the in-house educational software BIKE(E), based on design of a "virtual knowledge". This specific default data structure enables solving knowledge (educational data) transmission through the off-line and online environments. In comparison with other solutions, it seems to be the most universal and sustainable solution for personalized IT support. AQ1

Keywords: Computer support of teaching · Technology enhanced learning
Educational technology · Educational IT tools · Educational software
Digital technology · Personalized IT support

1 Introduction

In principle, each strategic policy is dedicated to integration of digital technology to education. However, in contrast to long-term teaching needs, the life-cycle of

K. Arai et al. (Eds.): FTC 2018, AISC 880, pp. 1–12, 2019.
https://doi.org/10.1007/978-3-030-02686-8_71

technological tools in real life is shorter. This fact is rarely discussed in scientific literature, where it is automatically supposed that any technology or computer support of teaching works perfectly. However, from the teacher's point of view, it works not perfectly. He needs a sustainable, i.e. long-term personalized support. Such personalized support was a key issue of some ICT calls of the EU-FP7 research program, and its importance is very realistically emphasized in [1]: "Ideally, personalization is relevant to all the stages of the learner's journey". Similarly, its importance for lifelong competence development is discussed in [2], or e.g. in the field of information and knowledge processing is discussed in [3]. In view of this, teachers still use for their teaching mostly global solutions (learning management systems, WEB-services, cloud computing, even social networks). Around ten years ago, the global learning management system Moodle was indicated as the most frequently used TEL - tool at European universities [4]. It is still used, although nowadays, universities already use their own learning management systems. One should be aware that such global solutions are static and not personalized enough (they could be considered rather as a kind of an information or knowledge management). It is emphasized similarly in [5] that "the most of the existing e-learning platforms offer just a single way to organize the course contents (book structure)" and "whoever is interested in organizing the course contents in a different way must use specific tools that are external to the e-learning environment".

Any support of uncertain and unstructured teaching processes by digital technology is very complex, and it is much more difficult than a common IT support of technical processes, which are well structured. This results into a certain terminological chaos and different approaches like e-Learning, Technology-enhanced learning (TEL), Educational technology, Computer supported collaborative learning. Nowadays, e-Learning is mostly considered to be subject of distance learning e.g. as it was indicated by University of Oxford, and "confusing" in general. It was replaced by the term "Learning and technology" (the terms Technology-enhanced learning and Educational technology were not accepted) [6]. From the real teaching practice point of view, the subject of TEL was very realistically described in [7]. As for the practice, the actual situation in TEL was discussed in the UCISA report [8]. In this report, it is argued that lack of time, departmental/school and institutional culture, including internal funding remain "the leading barrier to TEL development".

It is not possible to present here a complex review because the integration of digital technology into teaching and learning covers many fields of ICT and Computer science. However, as could be generalized from the above mentioned, developing personalized approaches and universal, all-in-one software is still a big challenge. To confirm the lack of software solutions, one can quote from [9]: "a system design on the model basis has been widely ignored by the community until now, and software engineering is missing in TEL system development". In light of the knowledge principles of teaching and learning, such missing models should be developed based on the subject of knowledge. In this context, semantic and ontological approaches should also be taken into account, e.g. as a connection of technology and education [10], or knowledge representation and ontologies [11]. The high level of ICT can be a basis for development of other specific approaches, e.g. for solving ontologies dedicated to visually impaired people [12], educational cloud computing [13], or combinations with machine learning or evolutionary algorithms [14].

Based on the idea that computers can be the means of automation of teaching processes (when knowledge is accepted as a regulated process parameter), author's approach is based on modeling a knowledge representation using the so-called virtual knowledge as a specific default data structure. They already published this approach [e.g. in 15, 16]. To better understand such approach, it must be mentioned that the authors' personalized computer support of teaching was empirical at the beginning. Step by step it evolved into systematic research under the umbrella of TEL (it was the official term of the FP7-ICT calls in period of 2007–2013). Within this research a paradigm of batch (balk) information and knowledge processing was designed, which is performed by the in-house developed software BIKE(E) (Batch Information and Knowledge Editor and Environment). Its selected part WPad is installed on classroom computers. The software is written by the main author of this paper.

By using BIKE(E), teaching material, informatics training tools, and personalized virtual learning environment with communication channels were produced. Together with personal cloud and network spaces, this created a background infrastructure for integrating digital technology into teaching. The personalized approach to teaching processes as knowledge based enabled authors to research the automation of teaching processes by using a model of virtual knowledge, including solving the knowledge transmission between computers and networks (registered at the patent office). In addition, this virtual knowledge as default data structure enabled teachers to research new approach to collaborative teaching and collaborative activities of teachers, students and researchers on shared virtual spaces or clouds. Authors' current research is focused on (multi-lingual) human knowledge processing, modelling a new generation of educational packages, and working on a vision of educational robot. It should be emphasized that this approach does not use any mathematical model. Technological design is simply based on a simulation of mental processes running within teaching, self-study and associated activities (e.g. writing paper or research project). In principle, the virtual knowledge only "switches" between human and machine as it is needed for (educational) knowledge processing and transmission, including informatics activities - to assure compatibility and adaptation to Windows and networks.

The authors' target is that all must be low-budget, effective and user friendly from the users' point of view, including using minimum of interfaces and software. And of course, knowledge processing is in natural language. To the best of authors' knowledge, such approach (knowledge representation, data transmission, personalized educational software, batch knowledge processing paradigm) does not exist yet. It seems to be beyond the state of the art, i.e. there are no analogical results in the literature on modelling the automation of teaching processes. It should be also emphasized that the authors' complex approach covers many informatics areas in comparison with common solutions of computer support of teaching, which are mostly single-purpose and monothematic. In other words, many barriers of digital technology integration into teaching must be overcome, especially from the long-term teaching point of view. Some of them are discussed in the next section that relates to internet browsers, changes of hardware and software, servers and networks, including human factor and behavior of students. The significance of didactics and informatics algorithms and off-line and online data transmission for automation of teaching processes is discussed in the separate sections (it is important for collaborative learning activities).

2 Barriers of ICT Integration into Teaching

2.1 Internet Browsers

Internet browsers are needed because the teaching material was designed by the teacher and produced by the continually developed in-house software BIKE(E)/WPad as a set of html-files. They were used for browsing both in offline mode (classroom computers, notebooks) and online mode (faculty's virtual learning space). At the beginning of research on TEL, identical learning content was used by students in classrooms, faculty's server and teacher's private internet domain. Within 10 years of authors' research, only internet browsers and WPad were used, despite the fact that mostly PowerPoint is commonly used for lectures and exercises. Each student or group of one to three students had always the study material at their own computer screen (it was combined with using sheets of paper or blackboard, e.g. when writing chemical formulas). From the pedagogical point of view, the OPERA browser under Windows XP was the most suitable browser. It enabled students to create sessions from study materials (something like a personal library), however, only until version 9.27, when it was re-designed to be compatible with the Google Chrome browser. In general, internet browsers versions are continually updated, therefore they have short life cycle. Moreover, each new version of the internet browser required an increased memory (RAM) capacity. It resulted in slow performance of classroom computers, consequently, teaching process was often interrupted. The same situation repeated itself when the "more popular" Firefox and Google Chrome were used later on the Windows 7 operating system. In this case, older versions of internet browsers had to be used because the higher versions were not supported.

Such obstacles complicated the computer support of teaching when students generated html-files, e.g. within writing the collaborative semester work, or production of study material for chemistry to eliminate their lower knowledge. Other specific problems occurred, e.g. because the html-outputs generated by WPad were not always identical in every browser, including problems with the diacritic of natural language (some computers had only English settings).

It must be mentioned as well that the described research was performed by the teacher at the faculty's detached workplace which has no IT administrators.

2.2 Changes of Hardware and Software

One should be aware that computers and hardware have shorter lifetime than the one that is required by any long-term teaching and learning. The teacher had to move all his documents, pictures and other files from old computers to new computers (commonly tens to hundred thousand files). It was therefore very time-wasting, to change data for

10 or 15 classroom and home computers. But it was even more time-wasting to install previously used software into new computers, including its activation. In addition, certain 32-bit programs, e.g. for speech recognition, worked only on 32-bit computers. Thus, the teacher had to upgrade to 64-bit versions of the software.

In comparison with this, the transfer of study content created by the educational software BIKE(E)/WPad was not problematic because this database application functions well under any Windows operating system. However, because of newer Windows versions and software, certain specific items of the user menu had to be modified in order to adapt the existing programming codes to the newer operating system. It affected mainly the menu items design related to file management in offline mode. It must be also noted that these obstacles are rarely mentioned in literature. In all related scientific papers, it is mostly automatically supposed that technology works without any problems. In the case of IT support of teaching, computers worked reliably commonly for three to five years. After that period they were slower and slower.

For example, it took the teacher’s work computer - with Windows XP (2005) on it - ten minutes to launch and it was regularly frozen during the antivirus background scan. Nevertheless, this was eliminated via remote control, i.e. by switching from computer to the teacher’s virtual space on the faculty’s cloud. After the switching, the teacher worked without problems using Windows 7 and WPad (client-server).

In real life, a teacher must simultaneously use computers with different operating systems, usually combining his work computers with home client computers and notebooks. In our case:

- Home computer’s lifecycle with Windows XP on it was about 15 years before its break-down (currently, Windows 10 notebook).
- Work computer with Windows XP (2005) on it was discarded (2017) when the teacher changed work position.

To summarize, the lifecycle of hardware and software was less than five years. The educational software BIKE(E)/WPad was working without any problems for 12–15 years. This meant that it was not problematic to copy the educational content, only certain menu items, related to the compatibility with Windows functions or general software, were rewritten.

2.3 Servers and Networks

Nowadays, digital technology is more or less embedded in every kind of teaching, so to use terms like “traditional face-to-face teaching”, blended learning, WEB-based learning, instructional design, as well as, e-Learning, technology-enhanced learning, educational technology, seems to be problematic. Namely, if a teacher wants to solve the IT support of his teaching directly or indirectly, he must combine several kinds of methods and technological tools (software, hardware).

Despite the fact that authors published their research continually under the umbrella of technology-enhanced learning, they consider their research basically as face-to-face teaching, which is supported by any off-line and online digital technology tools to automate teaching processes. During teaching hour, an IT tool must be used which enables the teacher to deliver teaching content in the quickest way and within limited time. Namely, one must be aware that teaching hour has only 45 min and the teacher must cover multiple topics within this short period.

In this context, after implementing the technological off-line and online infrastructure with teaching content (educational knowledge repository), it was needed to solve the feedback and knowledge flow transmission. Therefore, in this stage, the teacher wrote PHP/MySQL internet applications which functioned as communication channels between the classroom and faculty's server learning space. For example, when teaching C++ programming, this was the quickest way to transmit programming codes of students sitting in classrooms, i.e. downloading and uploading files generated by WPad in combination with the Bloodshed Dev-C++ software (Note: it is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language - http://bloodshed.net/ - http://orwelldevcpp.blogspot.sk/2015/). In this case, students wrote C++ codes to WPad-IDE, and using a specific user menu item, the codes were sent and directly opened in Dev-C++-IDE. Finally, students launched it simply by pressing F9. In this case, WPad functioned as a repository of solved codes. Then new bachelors student could use the base of programming codes written by students in previous semesters.

Such support of the programming languages teaching seems to be the ideal pedagogical solution. Namely, if a group of students had a task to write a programming code and only one student in the classroom wrote the correct code, then the teacher only asked the student to upload his code to the virtual learning space and all students could see it on their computer screen (and eventually download it and practice on their computers).

Now, one is able to understand, that several years ago, when the faculty's Windows server switched to UNIX where texts were case sensitive, the amount of work the teacher had with rewriting hundreds of learning (browsing) paths within existing PHP-codes. This illustrates that a teacher must overcome also these specific barriers, especially a teacher who teaches, designs and researches his teaching using digital technology for a longer time. Despite this fact, it was not the most crucial moment. Situation got much more complicated when after ten years of existence of the faculty's support learning portal, the life of the internal server technically ended (2017). In other words, the study content and support tools on the faculty's virtual learning space are not usable for teaching anymore. On the other hand, a new server is being just launched, so a part of content will be transferred to it. This example illustrates the fact that the life cycle of hardware, networks and servers is shorter than the lifetime which is required for the long life-learning of individuals.

2.4 Students and Human Factor

Authors' approach to computer support of bachelors teaching was continually developed according to the reactions of students to pedagogical/didactic and informatics issues. At the beginning, the in-house software BIKE(E)/WPad was programmed to produce teaching and study material for some courses of study (Background of environmental protection, Occupation health and safety, Chemistry, Industrial Management). The generated html files were uploaded via teacher's FTP access to the faculty's server.

Despite the fact that a good pedagogical quality material, which was suitable for self-study, was prepared, students were not enough motivated to use it. In order to improve the feedback, a specific internet PHP/MySQL application was written that functioned as communication channel. So each course of study was supported by this communication channel to simulate popular internet chats.

However, there were two challenges. Firstly, students surprisingly were not able to guess what to write into the communication channels, although the channels were also dedicated to information exchange between students. Thus, the teacher had to tell them to, e.g. write there the titles of their semester works. After several years of using the communication channels, one can say, that they were used by students without teacher instructions in less than 5% cases (it was mostly communication between distance students). This means that the channels were rather used for controlled communication, instructions of teacher, embedding of study material, even for testing suitability for semester exams.

Secondly, if one writes WEB-applications using PHP-forms, there is still the problem of text formatting. So if the student or teacher is to write any text into a text-box form they must be aware that the form is not a text or html editor. In our case, this was solved by pre-formatted text templates as illustrated on Fig. 1. To be honest, one can say that students did not fully understand it. On the teacher's side, when he prepared study or instructional materials for the communication channels by using BIKE(E)/WPad, this challenge was solved by writing a code for user menu, which converted natural language text to html-text. This text was then copied to text-boxes of the communication channels.

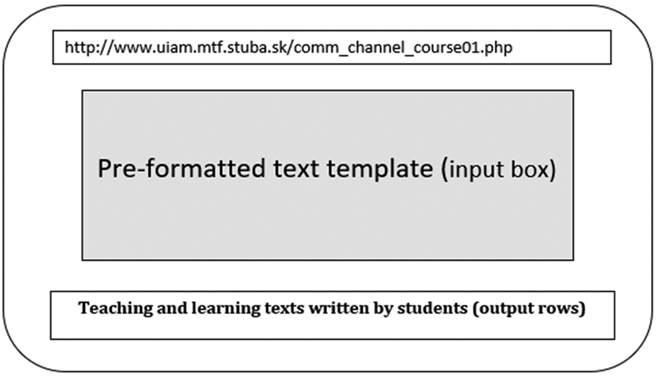

Fig. 1. Example of the text area of communication channels.

After long-term development of the personalized computer support teaching for several courses of study, one should be aware that there are hundreds of links and paths at disposal in the virtual learning environment, open WEB-domain or on off-line classroom computers. To support browsing teaching and study materials, a navigation html file was prepared for each course of study. Such navigation file contained thematic blocks with set of links (e.g., Study material, Communication channels, Calculation area, Tests, Results).

Surprisingly, there was the same problem at the beginning of every teaching hour. Namely, students were not capable to open the links and start the work on time. It took sometimes 10 min off the teaching hour. Although the links were put into Favorites of each browsers, the problems continued. It also occurred after the decision to use only Internet Explorer as default browser (e.g., some students installed Firefox or Google Chrome on their classroom computers again), or after sending them an e-mail with the links. There was only one possibility to technically solve this problem. However, it was time consuming for the teacher who personally had to set the links on each classroom computer. One could wonder where the so called "digital natives" were?

Here must be noted that this situation is connected to the general problem when using computers, i.e. how to switch within and between off-line and online interfaces (see, how we all click, click and click on links). From a research point of view, it logically led to an idea that the navigation paths must be dynamically joined with knowledge flow. It enabled to work out additional BIKE(E)/WPad codes and test the codes for this problem. In other words, the virtual unit should contain both content and active elements (off-line paths, online-links). This was successfully tested within teaching the programming language course which has commonly 4–7 students. This resulted in new authors' visions for the future research that in principle the pedagogical/didactic, informatics and application algorithms should be solved and designed synchronically (compare it with the next text as regards the categories of algorithms).

3 Categories of Algorithms

A typical problem when solving the computer support of teaching is the mechanical use of existing general software. It is suitable mostly for content storing and processing, not for feedback and dynamic teaching activities which are connected to continual knowledge flow. However, when anyone writes programs, e.g. for accounting, he must know accounting rules and algorithms (a sequence of accounting steps). Thus, if anyone should write programs for automation of teaching processes, as well, these must be primarily defined and described by the teaching algorithms (i.e. as sequences of pedagogical/didactic steps).

According to the authors findings, from a programming of view, with one item of the teaching content tens of activities (events) can be potentially connected. Such items can be represented by information, content knowledge, schema, which should be used (processed) within a teaching hour in various ways and up to the teacher needs. Similarly, other activities can be connected with the items like reading texts, constructing teaching material, internet retrieving, writing a paper, assessment, translating, repeating, drilling etc. If one had a set of defined pedagogical/didactic algorithms and was able to write programming codes or applications in general, he would find that this is additionally connected with many activities on his computer, e.g. file management, conversions to various formats (texts, visualization, audio/video). For example, when writing a paper, he must continually switch between many off-line and online windows, folders and use many paths and links. In other words, he must be aware which repeated algorithms he uses when working on his client computer, notebook or network.

During their research, authors indicated that the above mentioned approach “how to computerize and automate teaching and associated educational activities” requires one to solve three categories of algorithms:

- Pedagogical/didactic algorithms needed for teaching (within lectures, exercises, self-study).
- Algorithms for solving the adaptation and compatibility related to Windows, servers and networks.
- Application algorithms for supported teaching activities (to produce outcomes by processing knowledge flow).

A very important finding from modeling the Computer supported collaborative learning was that the identification of pedagogical/didactic algorithms is crucial for this kind of IT support. It is also the key issue when solving shared collaborative activities of group of teachers and researchers. From a layman point of view, if a sequence of teaching steps is not known and defined it is not possible to write any programming codes for collaborative and shared activities.

4 Off-Line and Online Data Transmission

Teaching process is about knowledge transmission between teacher and students. Therefore, the issue of data transmission is very important. In view of this, one should be aware that a teacher or students must use and manage hundreds of online and off-line sources to find and select a suitable information and knowledge. Technology enables them to perform information or knowledge management basically as the so called file management. Nowadays, in real life, a teacher uses commonly multiple home or work computers, i.e. client computers and notebooks, including virtual spaces of clouds and networks. Figure 2 illustrates such personal infrastructure which was developed within the long-term research on TEL to overcome the technological barriers mentioned above.

The issue of educational data transmission (content transfer) is less described in scientific literature than content processing, despite the fact that it is very time consuming. It does not matter whether a teacher teaches ten to hundred bachelor students or he solves automation of personal processes. To overcome this time barrier, it is important to identify which activities he must often repeat, in order to write appropriate programming codes. As regards the computer-computer file transmission when using the ICT infrastructure in Fig. 2, one has various alternatives, i.e. using USB, file manager, Windows Explorer or WIFI to copy/move or download/upload the files. However, if he often needs to transfer files between folder C:\AA and D:\BB he can automate it e.g. by the command written into the bat file:

XCOPY C:\AA\file.* D:\BB\file.*.
In our case, the teacher must transmit files between home computers, IBM BOX cloud and faculty’s cloud. The quickest way is by using IBM BOX cloud, which enables synchronization, so, it is possible to write the following bat-file command:
XCOPY C:\AA\file.* C:\USERS\…\BOX SYNC\file.*

(i.e. to copy the file into BOX Sync-folder on teacher's computer). In the case as in Fig. 2 that WPad is installed both on the faculty's cloud and client computers/notebook (actually Windows 10) the bat-file command could be:
XCOPY C:\AA\file.* \\TSCLIENT\D\BB\file.*
As regards the transmission of BIKE(E)/WPad-files (database tables), it is possible to save them directly to the BOX Sync-folder. As was presented within the FTC 2017 conference, the data transmission is performed according to the Utility model registered on the Patent office.

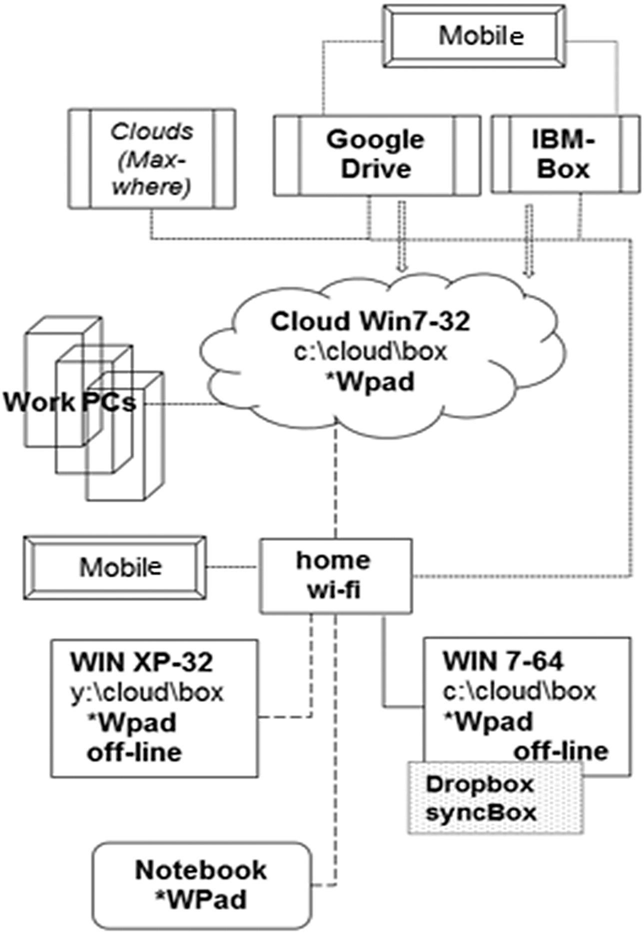

Fig. 2. Example of the personal teacher off-line/online ICT infrastructure.

5 Conclusions

In contrast with actual scientific literature, which automatically supposes that digital technology always works in the education setting without problems, this contribution discussed some challenges regarding the IT sustainability (suitability) for long-term Engineering teaching. Based on around twelve years of experiences with developing a personalized computer support of teaching, it was demonstrated on examples (from the teaching practice and academic research) that life-cycle of browsers, software, hardware, servers and networks is shorter than a real personal computer support of teachers and students requires. Some categories of barriers were mentioned. Basically, the life-cycle of technological tools is too short (3–5 years), while the teacher needs the technology for much longer periods. The authors' research has shown that the best solution to this problem is to develop both their own educational software and personalized ICT infrastructure for educational data transmission (see Fig. 2). In other words, the sustainability of the computer support of long-term teaching was achieved by using the all-in-one software application BIKE(E)/WPad (it has been in use for more than ten years). The application was continually developed as an all-in-one software while teaching around two thousand bachelor students under umbrella of the TEL. In principle, it is based on design of a virtual knowledge data structure that enabled a single user to handle educational content in many ways up to his personal needs. This provides teachers, students or researchers with a user friendly transmission of personalized data via the virtual knowledge structure through off-line and online environments. In this context, the software application BIKE(E)/WPad is additionally suitable for modelling the collaborative activities of students or researchers running on faculty's servers or global clouds. The most effective and low-budget solution for teaching and collaborative activities was the case when WPad was installed both on cloud or virtual computer with Windows 7 and personal computers of teacher and researchers.

Technically, the research limitations are that this software is Windows dependent and that all content must be converted to plain text format. From the pedagogical point of view, the teachers must be willing to use the software and be able to think of ways how to insert knowledge content into the virtual knowledge mentioned above (e.g. content of the course of study). The future research on collaborative activities requires solving a design of their pedagogical-didactic algorithms as a background for writing the programming codes and applications. Authors also presented a new vision of their future research proposing that the pedagogical-didactic, informatics and application algorithms should be designed synchronically.

References

1. Laurillard, D.: Digital Technologies and Their Role in Achieving Our Ambitions for Education. https://www.researchgate.net/publication/320194879_Digital_technologies_and_their_role_in_achieving_our_ambitions_for_education. Accessed 16 Feb 2018
2. Kostadinov, Z.: Sharing personal knowledge over the Semantic Web. In: Proceedings of the International Workshop: Networks for Lifelong Competence Development, Sofia (2006). http://dspace.ou.nl/bitstream/1820/720/1/Paper09.pdf

3. Bieliková, M., et al.: Personalized Conveying of information and knowledge. Studies in informatics and information technology. In: Research Project Workshop Smolenice, pp. 53–86. SUT Press, Bratislava (2012)
4. Matusu, R., Vojtesek, J., Dulik, T.: Technology-enhanced learning tools in European higher education. In: Proceedings of the 8th WSEAS International Conference on Distance Learning and Web Engineering, Santander, Cantabria, Spain (2008)
5. Alfano, M., Cuscino, N., Lenzitti, B.: Structuring didactic materials on the WEB (STRUCT). Commun. Cognit. **41**(1 & 2), 53–66 (2008)
6. Programme Specification for M.Sc. Education (Learning and Technology). Department of Education, University of Oxford. http://www.education.ox.ac.uk
7. Goodman, P.S., et al.: Laurence Erlbaum Associates: Technology Enhanced Learning: Opportunities for Change (2002)
8. Walker, R., Voce, J., Swift, E., Ahmed, J., Jenkins, M., Vincent, P.: 2016 Survey of Technology Enhanced Learning for higher education in the UK. UCISA TEL Survey report (2016)
9. Martens, A.: Software engineering and modeling in TEL. In: Huang, R., Kinshuk, Chen, N.-S. (eds.) The New Development of Technology Enhanced Learning Concept, Research and Best Practices, pp. 27–40. Springer, Heidelberg (2014)
10. Tolgyessy, M., Hubinský, P.: The kinect sensor in robotics education. In: Proceedings of RiE 2011, 2nd International Conference on Robotics in Education, Vienna, Austria, pp. 143–146 (2011)
11. Haidegger, T.: Developing and maintaining sub-domain ontologies. In: Proceedings of Standardized Knowledge Representation and Ontologies for Robotics and Automation. Workshop at IEEE/RSJ IROS, Chicago, IL (2014)
12. Mikulowski, D., Pilski, M.: Ontological support for teaching the blind students spatial orientation using virtual sound reality. In: Advances in Intelligent Systems and Computing book series (AISC) Interactive Mobile Communication Technologies and Learning, Proceedings of the 11th IMCL Conference, vol. 725, pp. 309–316 Springer (2018)
13. Shyshkina, M.: The general model of the cloud-based learning and research environment of educational personnel training. In: Auer, M., Guralnick, D., Simonics, I. (eds.) Teaching and Learning in a Digital World. ICL 2017. Advances in Intelligent Systems and Computing, vol. 715. Springer, Cham
14. Volná, E., Kotyrba, M.: A comparative study to evolutionary algorithms. In: Proceedings 28th European Conference on Modelling and Simulation, ECMS 2014, Brescia, Italy, pp. 340–345 (2014)
15. Svetsky, S., Moravcik, O.: The implementation of digital technology for automation of teaching processes. In: Proceedings of the Future Technologies Conference, San Francisco, USA, pp. 340–348. IEEE (2016)
16. Svetsky, S., Moravcik, O.: The empirical research on human knowledge processing in natural language within engineering education. In: WEEF & GEDC 2016: The World Engineering Education Forum & The Global Engineering Deans Council, Seoul, Korea, pp. 10–12 (2016)

Author Query Form

Book ID :	**465335_1_En**
Chapter No.:	**71**

Springer
the language of science

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author
During the process of typesetting your chapter, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query Refs.	Details Required	Author's Response
AQ1	Please confirm if the corresponding author is correctly identified. Amend if necessary.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	— under matter to be changed	
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	