ULTT – A UNIVERSAL LEARNING AND TEACHING TOOL FOR COOPERATIVE OPEN LEARNING

Max Kesselbacher, Corinna Mößlacher, Stefan Pasterk, Barbara Sabitzer

Alpen Adria Universität Klagenfurt
mkesselb@edu.aau.at, comoessl@edu.aau.at, stefan.pasterk@aau.at,
barbara.sabitzer@aau.at

Abstract

During the last years smartphones and the internet had a deep impact on our daily life as well as on learning and teaching in schools. There are a lot of tools for e-learning and webpages with the possibility to create learning material and to organize the process of teaching and learning. The number of learning and teaching apps for phones or tablet PCs is also rising. After using and comparing some relevant tools our conclusion was, that all of them were missing different core aspects of a good useable learning tool. These core aspects were collected from interviews with teachers and students from schools and university as well. They form the requirements for our software project ULTT, a multi-platform teaching and learning tool developed by using the game creation system Unity and the server framework node.js. "Multi-platform" means that the tool will be available as webpage for the use on a PC and as app for smartphones or tablet PCs running Android operating system or iOS. This feature is one of the core aspects because it allows teachers to comfortably create material on a PC and the students can use their own devices to fulfill tasks, read texts and ask questions in the class-forum. That means two different user-types, teacher and students, with different requirements and use-cases are necessary for the tool. Each user can take the role of a teacher and create a class, develop own tasks like puzzles, quizzes or reading exercises, publish them to make it accessible for the other users of the class. Cooperative tasks are a special type of tasks which allow students to work together simultaneously on the same task. A teacher can provide different additional materials like texts or videos which are not included in the tasks. Furthermore, he/she gets a statistical overview on the achievement of the students. This paper reports on the development process including the analysis of similar products. After an overview of the pedagogical background of ULTT it describes the requirements and the functionality of the application. Finally, we present an outlook on the qualitative evaluation, which is still running.

Keywords: e-learning, multi-platform tool, software project…

1 INTRODUCTION

In today's schools the usage of technology for teaching and learning is an important component of education and the term e-learning has become very popular. Different tools for this purpose offer different functionalities and follow different objectives. Teachers can create content online or offline on their own and pass it to their students or they can make use of existing exercises from other sources. This usually happens on a PC, a Notebook or Mac because of well known working environments. Depending on the tool the students have to fulfill given tasks with the help of supported technology. In some cases these are again PCs, Notebooks or Macs. But some tools offer the possibility to use mobile devices like tablet computers or smartphones to work on exercises. These mobile apps have, because of screen size or other restrictions, in most cases a limited functionality and different usability compared to the standard software or websites.

During a discussion with some teachers who talked about e-learning tools and their experiences and problems with these tools, we became aware of some shortcomings of the available tools. One problem was that the number of offered tools rose in the last years to a very high amount, resulting in a not very clear clutter. A further point was that there are only a few tools available in German language. This can be a reason for not using a good tool. Other tools were lacking an integrated feedback and statistics system for teachers or the possibility for students to work cooperatively on exercises. Based on these experiences, we conducted informal interviews with teachers and teacher students, which confirmed the previous assumptions.

Because of these identified problems, we decided to approach the development of an e-learning tool which includes appropriate solutions to the problems. One of the main intention of this project was to

develop a tool with which teachers as well as students can work on any platform they prefer to, but with the same functionality and usability. So it should be possible that a teacher creates tasks on a Notebook, because it is more comfortable, and the students work on this tasks with their own smartphones, because they are more flexible. Another main goal was to integrate possibilities for cooperative work on the exercises.

This paper describes the first steps of the development process for a new e-learning tool called "Universal Learning and Teaching Tool" or short "ULTT". Section two discusses the two basic learning concepts e-learning and "COoperative Open Learning", short "COOL". They build the pedagogical background for the project. In section three some more or less well known e-learning tools will be described and evaluated. Section four presents the general concept for the e-learning tool ULTT and specifies differences to existing tools. The application itself will be presented in section five. It includes some technological details as well as information about the design. The last section gives a conclusion and some perspectives on future work.

2 LEARNING CONCEPTS

2.1 E-learning

In literature there exist a lot of definitions concerning the exact meaning of e-learning. So it is hard to find one that meets all requirements. E-learning is often used as a collective term for different forms of technology-supported learning [1]. It mostly references to the usage of computer and network technologies in educational aspects [2]. This paper will rely on the following definition for e-learning:

"E-learning is the use of electronic technologies to create learning experiences." [2]

With the initiation of Web 2.0 the functionality of the Internet as a teaching and learning resource changed from an information and material source to a modifiable environment. From this point on students could create their own content, work cooperatively on the same contents and publish material online. For example, the usage of Weblogs, Wikis, Websites, Podcasts, Media sharing, or Social Networks [1] does not require prior knowledge or extensive preparation, but can improve cooperation between students during lessons at school and beyond.

There exist different forms of e-learning and the varieties will even grow in future. Examples for e-learning scenarios would be standalone courses, learning games and simulations, mobile learning, social learning, and virtual-classroom courses [2]. So for example mobile learning is a specialization of e-learning but with its own requirements, possibilities and potentials, and focuses on the use of mobile devices for the learning process. Students can learn with their own mobile devices, which are mainly smartphones or tablet computers today, everywhere and at any time they like to [1]. Thereby, new possibilities and challenges for teachers occur. Mobile learning can appear in different learning-scenarios like

- Miniature but portable e-learning,
- Connected classroom learning,
- Informal, personalized, situated mobile learning,
- Mobile training/performance support [9].

2.2 COperative Open Learning – COOL

An Austrian initiative called "COOL", what is a shortcut for "COoperative Open Learning" [3] was started with the goals to handle strong heterogeneity in classrooms and to foster soft-skills. It started in 1996 in a vocational school in Steyr, Austria and uses concepts from progressive education, especially from the Dalton Plan from Helen Parkhurst [4]. Selected elements, like for example the three basic principles from the Dalton Plan "freedom, cooperation and budgeting time" [3, 4], were adapted and included in the "COOL" approach [3]. The title "COoperative Open Learning" implies the two different methods cooperative learning and open learning.

2.2.1 Cooperative Learning

Cooperative learning includes methods which imply working in groups of different sizes. With the cooperation of the students, their social and self-reliant learning skills should be promoted and enhanced [5]. Some of the basic elements of cooperative learning would be [6]:

- Positive interdependence: With the positive interdependence the dependency of all members in a group is described. The students have to recognize that they can reach their goals only if they work together.
- Individual responsibility: Each student of a group has to provide individual and recognizable performance although the group shares the goals.
- Supportive interaction: Within the groups the students should be encouraged to support, motivate and help each other.
- Reflection on the group process: After the group work is finished the members of a group should think and discuss about the whole process.
- Cooperative skills: Students have to learn and practice to work in groups. For cooperative learning the students have to improve their communicative skills, their skills for group management, and their skills to deal with controversies [6].

2.2.2 Open Learning

In literature the topic open learning is interpreted in different ways. Because it combines principles from different progressive pedagogic approaches like Freinet and Montessori, it is difficult to define the term open learning exactly [7]. One important information about an open learning scenario would be the dimension of the openness, which can be as follows [8].

- Thematic openness: the students have free choice which content they want to deal with and how much time and effort to spend on it.
- Methodic openness: the students have free choice of material and method they want to use during the work on a topic. This means that the self-reliant management of time and work is included.
- Institutional openness: the students have free choice of, for example, their working-location i.e. in which room they want to work at school. Further this kind of openness can be structured into organizational, social and personal openness. Organizational openness refers to the choice of location and time, social openness to joint decisions about e.g. class management or rule structure, and personal openness to the relationship between involved persons [7].

2.2.3 COOL features

Connecting these two education methods to “COOL” can foster the self-reliance, the personal responsibility and the cooperation for students in secondary education (from 5th school grade onwards) [3]. COOL can be integrated into traditional schools without changing the whole institutional structure for all classes. COOL classes have some basic features:

- Freedom of choice for students: Students get written assignments with tasks they should prepare until a given date. During COOL-lessons they have the choice of when, where, how and with whom they do their exercises.
- Portfolio: Additional to traditional assessment methods, students have to submit a portfolio which contains all of their achievements. Students can choose the form and type of their portfolios on their own.
- Evaluation and reflection: Students should frequently give feedback on their working and learning processes. Thereby they should analyze their own behavior.
- A new teacher’s role: Teachers adopt the roles of moderators and companions. That means they have more time to help solving out students’ difficulties.
- Teacher cooperation: COOL-teachers have to act as a team. This requires a closer cooperation and frequent team meetings.
- Class council: This council should be a regular meeting for the students of one class to discuss problems and to practice discussion rules, logging information and moderation techniques.
- COOL parliament: Teachers and students should work out the rules for a structured use of COOL together. These have to be recorded in so called contracts.

- Involvement of parents: Regular parent's evenings should lead to a closer involvement of the parents [3].

The use of computer and mobile devices is also promoted in the COOL approach because these technologies make cooperation and communication easier and independent from room and time. "eCOOL" describes the connection of COOL with e-learning and is an additional feature of COOL. Main characteristics of eCOOL would be [3]:

- Usage of learning platforms
- Assignments including e-learning-elements (eCOOL)
- Larger, individual feedback in digital form
- Less online-time, but more direct communication
- Usage of ePortfolios

The ePortfolios used in eCOOL can have different forms, like Weblogs, Wikis, Websites, Podcasts, Media sharing, or Social Networks, including the content collected and produced by students themselves.

3 E-LEARNING TOOLS

3.1 Description

One of the first steps of the project was a comparison and an evaluation of different free and online learning and teaching tools for PC, tablet-PC or smartphone, which are well known and frequently used in the author's surroundings. In the context of this paper, learning and teaching tools are defined as tools that provide teachers the possibility to create material for the students and make it online available for them. And further it means that students can use the tool to access their teacher's contents and to fulfill different exercises. Following the definition of [2] for e-learning, the tools described above can be handled as e-learning tools. These tools can have different forms as mentioned in chapter 2.1 and can be distinguished by their functionality. For example, the online platform "eLearningAtlas" [10] lists ten product types:

- Authoring Tool (233)
- Consultant (151)
- Content Library (429)
- Custom Content Creation (423)
- Learning Content Management System (127)
- Learning Management System (568)
- Learning Record Store (0)
- Mobile Apps (0)
- Online School (12)
- Other (85)
- Web Apps (0)

The numbers in the brackets were retrieved from [10] on the 19th of January 2015 and show how many products are available from each type on this website. So most of the e-learning tools are learning management systems (568), followed by content libraries (429), custom content creation (423) and authoring tools (233). It is interesting that for the type "mobile apps" and "web apps" no products can be found. These are no official numbers and for sure do not display the definite situation but they can give an impression of which products are needed.

3.2 Differences and Evaluation

The comparison of the existing e-learning tools was part of the project and should provide some information about the current situation. For this task, popular tools like "Khan Academy" [11] or

“Udemy” [12] and tools from the surrounding of the authors like “Moodle” [13] or “LearningApps” [14] were chosen. All results were collected and can be seen in Table 1.

Table 1: Results of the comparison of different e-learning tools

Tool	Authoring Tool	Content Library	LMS	Mobile App	Mobile function	Statistics	German language	Free
Moodle [13]	-	-	yes	-	-	-	yes	yes
LearningApps.org [14]	yes	partial	partial	-	-	-	yes	yes
EducaPlay [15]	yes	partial	partial	yes	read	yes	-	partial
Edmodo [16]	partial	partial	partial	yes	read	partial	partial	partial
Quizlet [17]	yes	-	partial	yes	create	partial	-	partial
Quia [18]	yes	-	partial	-	-	yes	-	trial
Hot Potatoes [19]	yes	-	-	-	-	-	-	yes
Zondle [20]	yes	partial	-	yes	read	yes	-	partial
Khan Academy [11]	-	yes	partial	partial	read	yes	-	yes
Udemy [12]	yes	yes	-	yes	read	-	partial	partial

It is visible that none of these e-learning tools provides all of the functions which the teachers mentioned in the interviews. These were

- fully usable in German language (possibly extendable for other languages),
- providing feedback and statistics for the teachers,
- a useful mobile version, and
- for free.

Based on this evaluation, the concept of our application was developed. The following section will describe the concept in detail.

4 GENERAL CONCEPT

4.1 Implications from tool evaluation

The evaluation of the learning tools came to the result that some of the design ideas are worth to be adopted. Nevertheless, some shortcomings have been detected that should be compensated by our concept.

Concerning the organization of learning content, the user should be able to structure his tasks in classes. Whereas some web tools support the structuring of tasks in classes, this is not a common feature for mobile learning applications. In our concept, the term “class” describes a structure where one user takes the role of a teacher and provides learning material to the other members of the class, who take the role of students. In short, this structure describes a traditional teacher-student relationship. Yet in our concept, the role of each user can change depending on the class he / she acts in.

The fundamental implications of the described concept of classes is that all users can create new learning content in the form of new tasks. When the user wants to make this content available to other users, he or she can create a class, add the tasks to it and invite users. While doing this he is in the role of a teacher. The invited users act as students of this teacher. Users can join classes by using a unique class registration code, which has to be distributed by the respective teacher. The registration has to be confirmed by the teacher. This ensures that the teacher has control about who can see the content of his class.

When creating learning content, the user can choose how his content is going to be presented by choosing among different task types. Many tools are restricted to a small amount of task types. Especially the tools that are usable on smartphones only have limited features. The tools that support more features are often only available as web applications. Since they are designed for the usage on a PC, the usage on a smaller device, like a smartphone, is often not comfortable. In our application, we want to provide a variety of diverse task types, which are suitable for usage on mobile as well as desktop environments.

The basic task types we want to include are: assignment (word-to-word assignments), category (word-to-category assignments) and quiz. The application will provide these and will extend the task types to support cooperative learning.

The tasks are created independently from the classes. Therefor a created task can be assigned to several classes. When assigning a task to a class, the teacher can decide whether it should be an exam or an exercise. The teacher gets feedback on the achievements of the students in exams. Many tools do not record the results of the users when executing a task. The result of the task is only temporarily visible. In some tools that do not have a class structure, the result is visible to the user.

In our application, we want to provide feedback for students and teachers. When assigning a task to a class, the teacher can decide whether the task shall be linked as an exam or as an exercise. The teacher only gets feedback on exams, whereas the students get feedback on all their executed tasks. So it is possible for teachers to provide learning tasks as well as tasks to test the knowledge of the students.

4.2 Incorporation of COOL

To support cooperative learning with the developed application, we thought about possibilities to include cooperative learning exercises with the basic task types described before. The first cooperative learning possibility we want to support is the peer-reviewing of exercise answers, regardless of the task type. For this, two students work together on the solution of an exercise. One student works on the first part of the exercise. After submitting, the results are not immediately checked but passed to the other student. When he / she is active the next time, a prompt asks to review the solution of the first student, make improvements to the solution and submit it again. This solution is regarded final and is checked for correctness.

Additionally the COOL concept is supported by the concept of our application. The users are not strictly bound to a role as teacher or student. Cooperative learning is possible via peer-generated learning content. Each user can create tasks and share them with other users. Another part of the COOL concept is open learning. Primarily the institutional openness can be supported by our application, since the application is designed for mobile use.

4.3 Multi-platform Tool

The application is designed for mobile usage like on smart phones and tablets as well as for the usage on desktop devices. To make the application useable for a broad group of people, Android operating system and iOS is supported. All features are fully usable on all platforms. The support of mobile devices makes it possible for users to easily consume learning content wherever they want to. The creation of content may be more comfortable on a PC.

5 APPLICATION DESCRIPTION

5.1 Used technology

The application is developed as part of a university software laboratory which focuses on developing real applications in a small developer team. After collecting the requirements for our application ULTT,

the next step was to decide which technology the application will be developed in. When sketching the desired application functionality, it was obvious that a division of the application into client and server part was needed. The following sub-sections summarize the considerations made to fully decide on the used technology.

5.1.1 Client Framework Unity

A key factor that influenced the decision of used technology at the client was the requirement that the application should be available as web application as well as on mobile devices, including Android operating system and iOS. While the former could suggests developing a standard approach of HTML5 and JavaScript, we were concerned about the latter. Then we heard about Unity, a runtime and development environment [21]. It was primarily designed to be used in 3D game development, yet in more recent releases also the support of 2D graphical user interface development is increasing. What makes the Unity engine especially appealing for this application is that it offers the option to build executables for desktop environments, different mobile device platforms as well as for the web (powered through the Unity web player), all from the same code base. Finally, this lead to the decision to use the Unity engine as client framework.

5.1.2 Server Framework node.js

After settling the client framework, a server technology had to be found. Because of the encapsulation provided by the Unity runtime environment for all possible user devices, a slim server technology was chosen. node.js, based on the JavaScript runtime environment V8, enables the development of a slim, fast and event-driven server in JavaScript [22].

5.2 Application design

5.2.1 General Structure

The starting point of the application is the login and registration screen. Registration is designed as single opt-in, by entering name, email address, user name, password and optionally the school name. After login, every user has the same basic rights: class creation, task creation and registration in the class of a different user. Registration in classes is handled by entering the unique class code string, obtained from the teacher of the respective class.

First, the basic use cases for a user performing the teacher role are described. Teachers can create classes for different subjects, and organize the classes with topics. They can also create tasks. Each task is assigned to one of the available archetypes. Driven by the archetype, the content of the task can then be entered. Last, the teacher links the tasks to the topics in a class, making them available for the students of this class. The right screen view of Fig.1 illustrates an exemplary class overview screen as seen by the teacher. The student's view of the class is similar, but the administrative control options (like linking tasks or adding topics) are not visible.

Basic steps for users performing the student role are to first receive the class code string from the teacher of the respective class. Then, they can register to the class with this class code. Then they have to wait until the teacher of the class admits the students, granting them class access. Lastly, the students can access the class and execute the linked tasks.

The application can be used from desktop devices as well as from mobile devices. Because the application was built from the same code base, the look and feel is very similar on all target platforms. For example, teachers might use the web player version on a desktop device for task creation and class administration to benefit from the wider screen and keyboard. In contrast, students can use their mobile devices to practice the content whenever they want to.

To use the application in the web player version, only the Unity web player plugin has to be installed. For mobile devices, an installation package is distributed which requires no additional software.

5.2.2 Task Types

After assessing the available tools, we decided on basic task types that shall be supported by our application. This choice was influenced by other tools, specifically including these task types that are most useful to cover a wide variety of different exercises and excluding redundant ones. Ultimately, we decided on 3 basic task types: assignment, category and quiz. Every learning task a teacher designs

is assigned to one of those task types. The task type determines how the information can be entered by the teacher and how the task can be executed by students.

The assignment task type captures the exercise archetype of connecting two words or phrases together. Learning information is entered as pairs of assignments. Assignment tasks can be presented either as flashcard exercise (one of the two phrases is visible, the other one has to be written down), or as pure assignment exercise where two columns of phrases have to be connected.

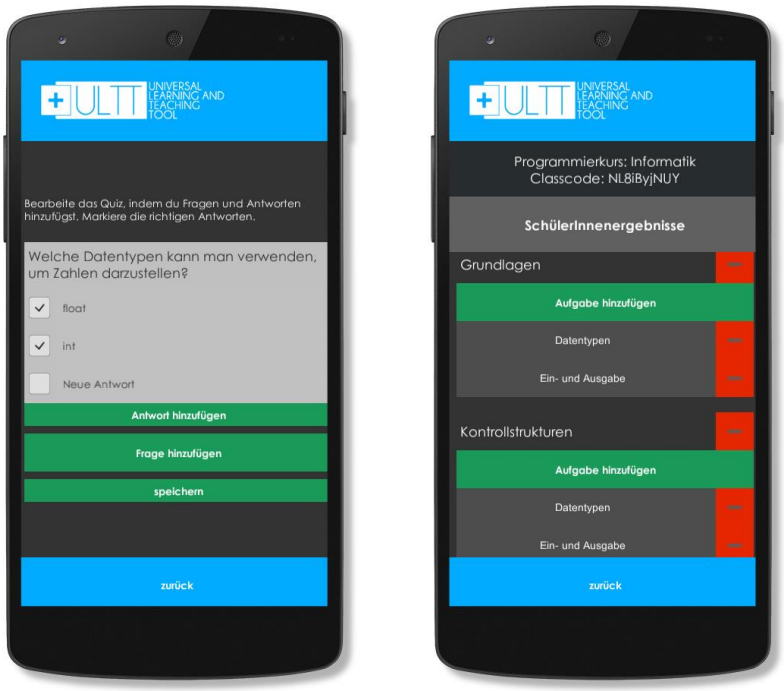

Fig.1: Sample screens of ULTT. Left: Quiz task creation. Right: Class overview of a teacher.

The category task type covers the connection of phrases to category names. For each category name, a number of matching phrases can be entered. Such tasks can be presented in two ways. First, by iterating through all the matching phrases and connecting them to the respective categories, one by one. Secondly, by presenting the student all matching phrases which have to be separated into the respective categories.

The quiz task type enables the creation of quiz exercises. A number of questions can be entered, and for each question different answers can be specified. Furthermore, each answer can be marked as a “correct” or “false” answer. This makes it possible to create questions with none, one or multiple correct answers. Exercise presentation is typical for a quiz: the questions are presented one by one. The student can select the answers deemed correct, which is then checked. The left screen view of Fig. 1 shows the creation form of an exemplary quiz task.

As can be seen, the supported task types already make it possible to cover a variety of exercises, which is further improved by the possibility to present different exercises from a single defined task type.

When linking tasks to classes in order to be executed by students, one of two different modes can be chosen. The first mode is linking as an exercise, which means that the achieved scores are not reported to the teacher. This allows the students to practice the contents without fearing the reporting of poor results. The second mode, in contrast, is the linking as an exam. Scores for exams are saved and reported for the teacher of the class. For all linked tasks, the number of attempts can be specified. This described linking is decoupled from the task itself, therefore making it possible to use the same task as an exercise for some classes and as an exam for others.

6 CONCLUSION AND OUTLOOK

The work on this project was challenging because of several criteria. At first it was hard to find the most important requirements for the tool, which no other tool provided. For this purpose the analysis of some existing tools was performed and resulted in four basic requirements for the project. These could be confirmed by informal interviews of some teachers and students. Compared to the development of the first alpha version, the process of requirements engineering took nearly the same amount of time. A second hard decision was the selection of development frameworks, which was already limited by requirements of a multi-platform tool. Because of prior knowledge and design decisions, Unity was chosen to establish the client functionality and node.js should be used as server-framework. Some time was necessary to deal with the two frameworks.

After finishing a functional continuous alpha version of the application, it is in our intent to first arrange alpha tests with a small, selected group which should provide us valuable feedback.

Also, it is a long-term goal to use our application in a case-study to evaluate a technology-enriched approach of COOL in real classroom environments. For this purpose different test-settings will be defined. Both user-types, teachers and students, will be tested independently. Test persons will be students as well as teachers from schools and universities, with all of them using the teacher role as well as the student role in the application. With this setting, the effects of the relationship between teacher-role and student-role in the application can be observed for both student-to-student and teacher-to-student real life relationships. Further both versions, the desktop and the mobile version, will have to pass test-cases to ensure their functionality and their usability. Some additional tests for the mobile version like data transmission and power consumption will likely lead to an improvement of the tool.

We also want to further explore the possibilities of cooperative learning within our application. Some of the thoughts described in our concept are already implemented. Besides, another form of cooperative learning we want to support in the future is the possibility to share the tasks between different users, making it possible to link generated content from other students or teachers to one's own classes. Effectively, it should be possible to browse tasks that were published by other users.

Another possibility of cooperative learning we plan to extend the application with are class forums where students and teachers can create forum topics. These forum topics can be used to publish additional learning material, or can serve as question and answer board where the teacher of the class and selected students can work together to solve questions about the class topic.

REFERENCES

[1] Ebner, M.,Schön, S. (2011). Lehrbuch für Lernen und Lehren mit Technologien (L3T). Bims e.V., Bad Reichenhall.

[2] Horton, W. (2012). E-Learning by Design. Pfeiffer, 2nd edition.

[3] Hölbling, R., Wittwer, H., and Neuhauser, G. (2011). COOL Cooperatives Offenes Lernen. Impulszentrum für Cooperatives Offenes Lernen. Available from: http://www.cooltrainers.at/fileadmin/impulszentrum/pdf/Cool_Booklet_120x180_lay1.pdf.

[4] Parkhurst, H. (1921). The Dalton Plan IV. The Times Educational Supplement. Retrieved May 24, 2013 from: Dalton Research and Developement: http://www.daltonplan.nl/uploads/media/Artikel_IV_Helen_Parkhurst_01.pdf.

[5] Greimel-Fuhrmann, B. (2006/07). Was ist und was kann Cooperatives Offenes Lernen (COOL)? In: wissenplus(3), Manz.

[6] Borsch, F. (2010). Kooperatives Lehren und Lernen im schulischen Unterricht. Kohlhammer, Stuttgart.

[7] Peschel, F. (2006). Offener Unterricht-Teil 1: Allgemeindidaktische Überlegungen. 4. Auflage, Schneider Verlag Hohengehren.

[8] Ramseger, J. (1977). Offener Unterricht in der Erprobung.Juventa, München.

[9] Kukulska-Hulme, A., Traxler, J. (2007). Design for Mobile and Wireless Technologies. In Rethinking Pedagogy for the Digital Age, H. Beetham, R. Sharp (2007), Ed. Routledge, London.

[10] http://elearningatlas.com/

[11] https://www.khanacademy.org/
[12] https://www.udemy.com/
[13] https://moodle.org/
[14] http://learningapps.org/
[15] http://en.educaplay.com/
[16] https://www.edmodo.com/
[17] http://quizlet.com/
[18] http://www.quia.com/
[19] https://hotpot.uvic.ca/
[20] https://www.zondle.com/
[21] http://unity3d.com/
[22] http://nodejs.org/