Enabling Synchronous Collaboration in Web Mobile Learning Applications

Ivica Boticki[1], *Nino Uzelac*[2] *and Martina Holenko Dlab*[3]

Abstract

This paper presents the Author system for designing and delivering interactive and multimedia-based learning lessons. The system is designed as a generic tool for building learning contents and as a multi-platform content delivery tool. Every web application is considered to be a widget that can be imported into the system and then made collaborative. By specifying collaboration attributes such widgets are empowered with synchronous collaborative support to be used in and out of the classroom, depending on the devices available to students and the teachers. The paper presents the system design, two collaborative widgets for synchronous math learning in pairs and triplets and the initial experiences in deploying and using the system in early primary school settings.

Keywords: mobile learning, web applications, collaborative learning, frameworks

1. Introduction

Creating applications for mobile learning nowadays comes with a variety of concerns and challenges such as sustainability, multi-platform deployment, usability. This puts additional burden on researchers and developers and potentially increases both the development and maintenance time and the price of the developed learning applications. This in turn potentially leads to the lack of creative and novel designs that could possibly push the boundaries in the field.

This paper presents a framework for mobile collaborative synchronous learning (Ivica Boticki, Wong, & Looi, 2013) that is based on novel and relatively mature technologies that could be applied to turn existing web mobile learning applications into synchronous collaborative mobile learning applications (Caballero et al., 2014; Chen & Chen, 2014; Sun, Looi, & Xie, 2014) . As part of the SCOLLAm mobile learning project (Jagust, Mekterovic, & Boticki, 2015), the Author system was designed and equipped with such extensions, therefore allowing applications developers to utilize synchronous mobile learning designs (Boticki, Baksa, Seow, & Looi, 2015).

In addition to demonstrating the design underpinnings of the system and synchronous collaborative learning extensions, a case of using two such applications is given and demonstrated within early primary school learner context. Such an approach is designed to demonstrate that existing designs can be easily converted into synchronous mobile learning designs and used in mobilized classrooms.

[1] University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, Email: ivica.boticki@fer.hr
[2] University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia, Email: nino.uzelac@fer.hr
[3] University of Rijeka, Department of Informatics, Croatia, Email: mholenko@inf.uniri.hr

2. The Author System

The Author system resides on the SCOLLAm architecture which is built to support development access to multiplatform web-based mobile learning applications and content. Designers can use the Author system to design and store web mobile learning lessons into the system database via the custom designed web services. Once the content is created it can be accessed both by the authors to be reviewed or reused and subsequently by the end consumers (students) to be used in educational scenarios (Figure 1).

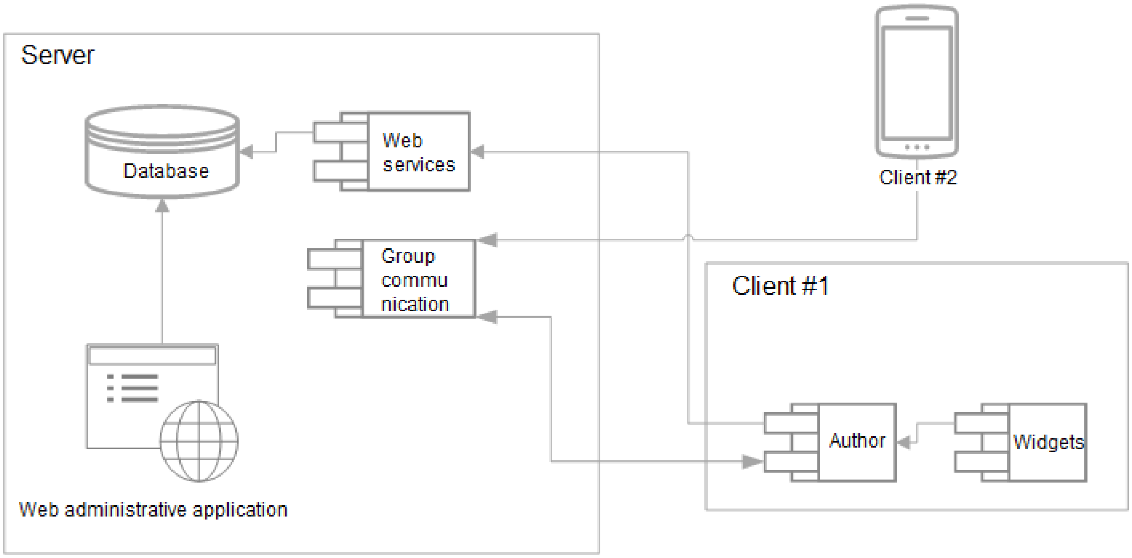


Figure 1. SCOLLAm architecture as the base for the Author system operation

2.1. Designing Multimedia Learning Content

One of the Author system functions is to serve as a tool for designing and reproducing interactive lessons. Each lesson is a package that may contain multimedia elements such as text, image, shapes, widgets and others. There is also an option of defining interactive rules that trigger an action based on some interaction. Some examples of user-defined triggers can be object move, object touch, object drag and drop that can make some other object change position, style or size. Using these simple concepts, a variety of interactive lessons can be created which would allow for easier usability.

Layout of the Author system is depicted in Figure 2. The layout of the system graphical user interface is divided into three distinct parts: the left column part showing all the lesson visual subelements (also to referred to as slides), the central part showing the currently edited visual element and the right column part showing the parameters of the element focused upon in the central area. The central part allows for the addition of multimedia visual elements via the top toolbar buttons.

A lesson defined in such a way can be played via the Author system by using the Play toolbar button. This means the lesson preview will start and that all multimedia and interactive elements will be shown to the end user. Similarly, such preview can be started from any web-enabled device outside of the Author system, which will be in detail presented further in the paper.

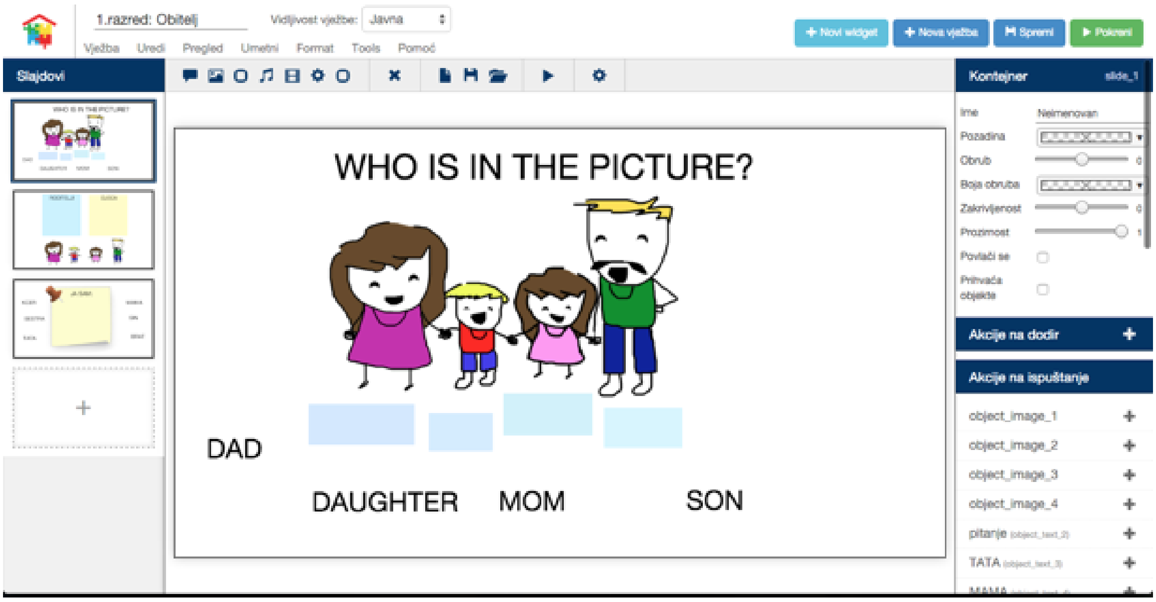


Figure 2. Author learning system - editor component

2.2. Widgets - Bundling Interactive Learning Content

Building multimedia lesson as presented in the previous paragraphs is a very straightforward way of building web and mobile learning applications, but does come with a limiting set of functions. Since users can only choose from a limited set of multimedia elements, there is no way to implement more interactive user experiences. This is not only a limitation as seen by the system power users, since nowadays even young learners expect more interactive and engaging elements for any application. The Author system opens up the possibility for delivering more interactive and engaging learning contents by introducing modular pluggable learning application support, known in the system as *widgets*. Widgets are taken as already created modules and inserted into digital lessons that are being designed via the Author system, as illustrated in Figure 1.

The Author system will accept as widget any already developed web-based content that conforms to the three design requirements:

1. The application **must** be developed using any web-technology that can be used in today's web browsers (such as Chrome, Internet Explorer etc.).
2. The application **must** be described by a simple metadata file indicating its main properties.
3. The application **may** communicate with the Author system to exchange some data (i.e. send usage log to the Author system, collect input parameters from the Author system or achieve group communication via Socket.IO etc.).

In order to better illustrate the use of widgets in the Author system, an example of a simple web application is presented in Figure 3. A simple standalone web application for adding two numbers is converted into a widget by simply adding widget metadata, which are shown to lesson designers as configurable parameters (marked with red in Figure 4).

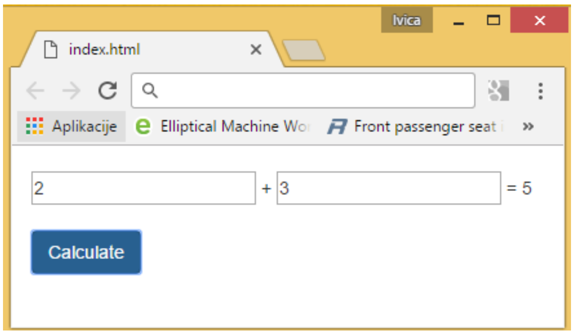


Figure 3. A simple interactive web page that is to be converted into a widget

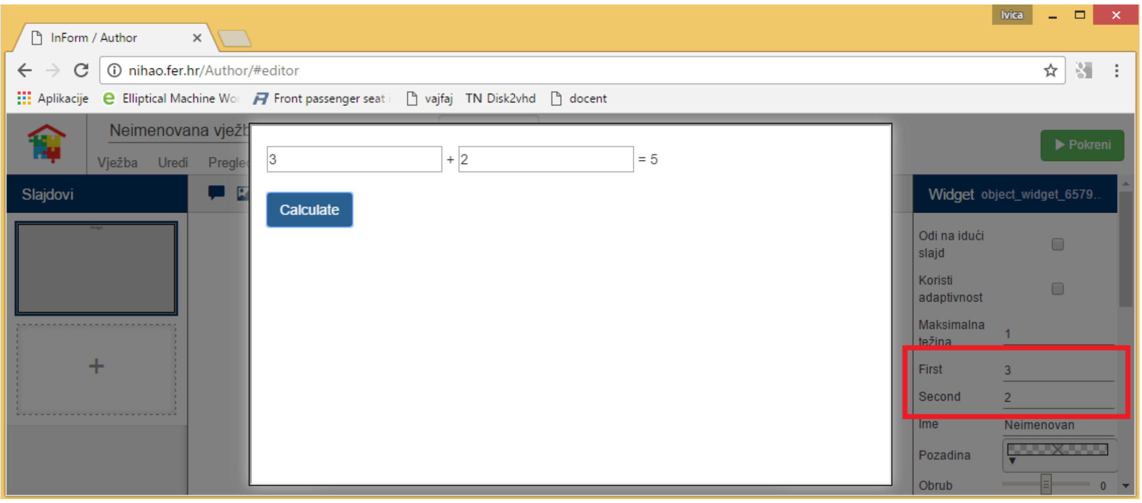


Figure 4. Anatomy of an Author compliant widget

Figure 5 illustrates in detail the anatomy of an Author compliant widget where the left hand side of the figure shows code used to realize the widget appearance and behaviour, the middle part of the picture widget metadata code used to parametrize the widget via the Author system, and the right hand side of the picture the interfaces used to communicate with the Author system and to achieve group communication.

It is to be noted that any web-based code or framework can be used to create a widget, and that widgets interfaces already exist and are ready to be used in programming by the widget creator.

Figure 5. Anatomy of an Author compliant widget

2.2. Synchronous Collaboration with Widgets

As a technological base for implementing synchronous communication in the SCOLLAm system, a Socket.IO was utilized. It allows for two way communication in real time and consist of a server and client components, which are both web-based. Socket.IO encapsulates and deals with numerous communication and coordination issues such as reliability, automatic reconnection via long-polling, detection of communication interruptions, various web browsers. It also allows for an easy setup of communication groups for sharing messages to one or more users on one or more devices.

Table 1. Synchronous communication methods in SCOLLAm to be used by widgets implementing collaborative learning scenarios

Method name and parameters	*Method description*
postGroupMessage(object message, object options)	Sending a message to all group members
postPersistentGroupMessage(object message)	Sending a message to all group members and saving the message on the server
onGroupMessage(function callback)	Listening for an incoming group message and reacting to her
updateStateOnBackend(object message)	Saving custom (any data) state on the server
requestState()	Requesting state from the server
requestMessages()	Requesting saved messages from server
onStateFromBackend (function callback, boolean executeOnlyOnce)	Listening to a state change from the server. The executeOnlyOnce flag will remove the listener if set to true
onMessagesFromBackend(function callback)	Listening to received saved messages from server
onUserInfo(function callback)	Listening to information on user roles from server
getState(function callback)	Eqivalent for onStateFromBackend(callback, true) +

	requestState for easy state fetch from the server
ready()	Notifying the Author system that the widget registered all listeners and is ready for message receival

As depicted in Figure 5, widgets creators will use already prepared communication interfaces in their widget designs to in order to utilize the SCOLLAm synchronous infrastructure for collaborative learning scenarios. The list of available methods which comprise the group communication widget interface is listed in Table 1.

Figure 6 illustrates which messages get exchanged when the Author system initializes a widget. Widget issues three initial method calls where the third lets the Author system know the widget is ready to begin its operation. The Author system then utilizes the server-side Socket.IO component to request any state pending on the server. This communication is especially important if widget designers want to save data as the user switches between different applications on her device.

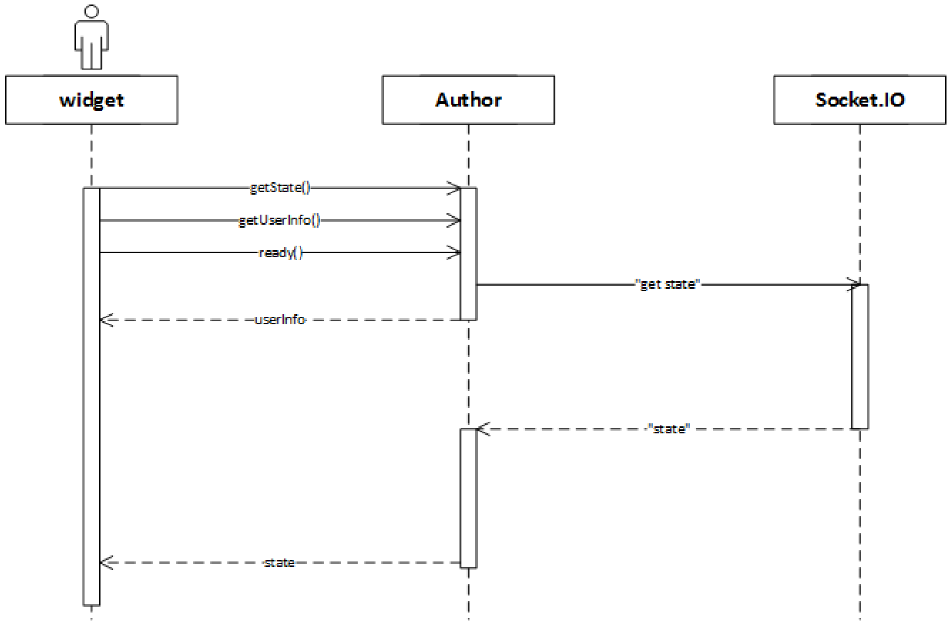
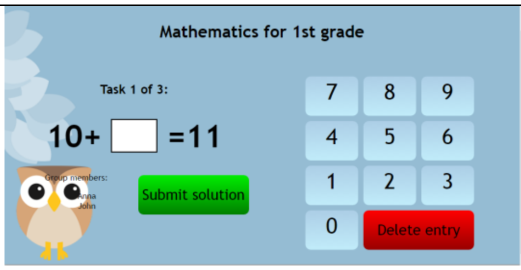
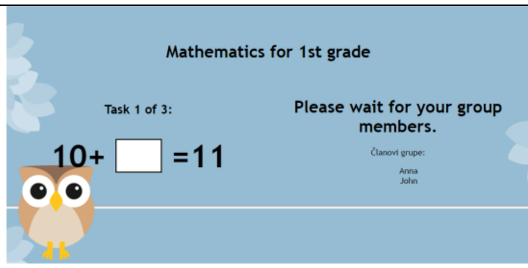
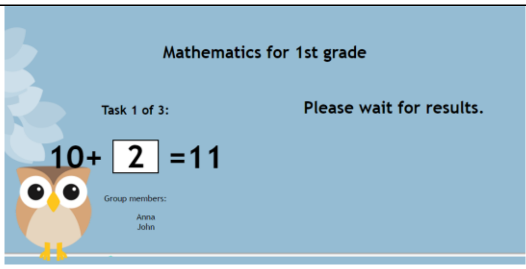
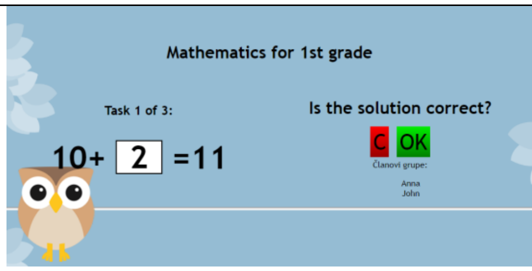
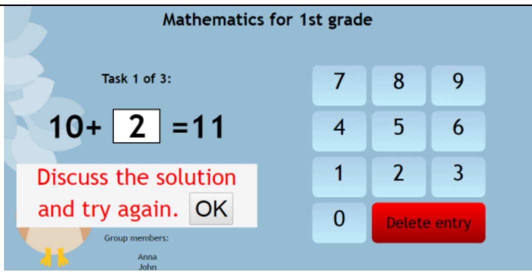
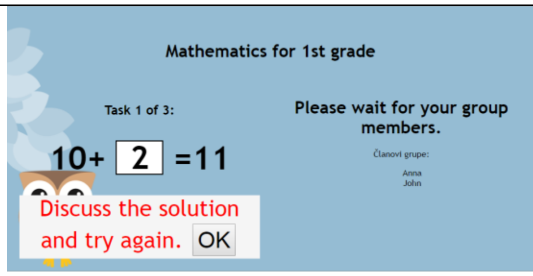
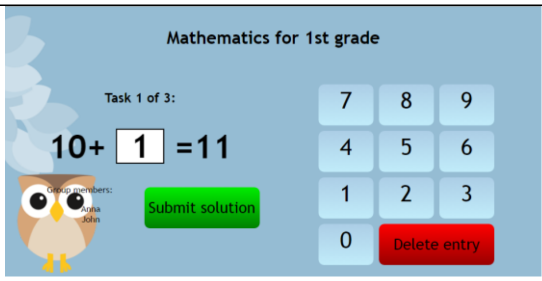
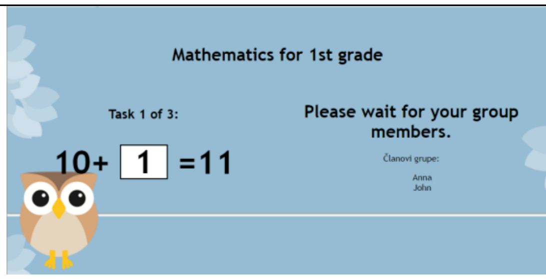
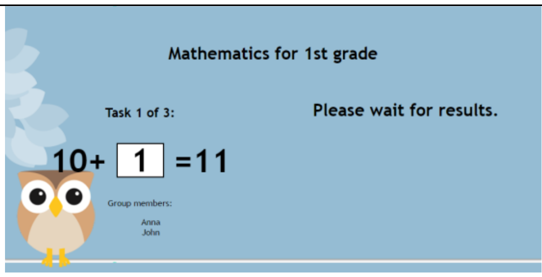
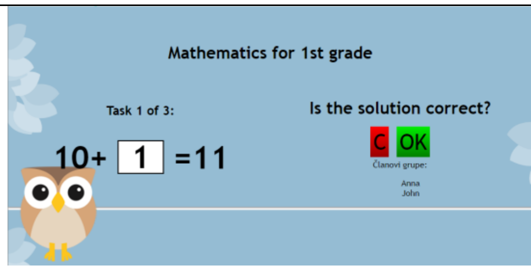
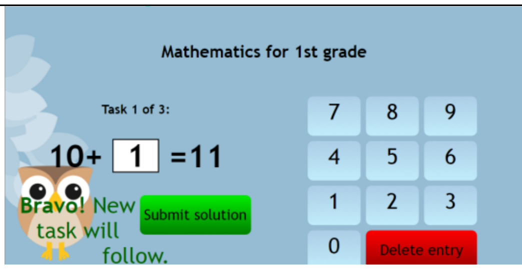
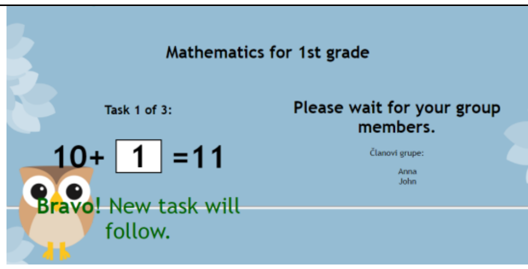


Figure 6. Messages exchanged upon widget initialization – requesting previously stored state from the server

3. A Case Study: Learning Early Primary School Mathematics with Pairs and Triplets Collaborative Learning Widgets

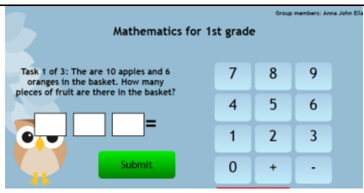
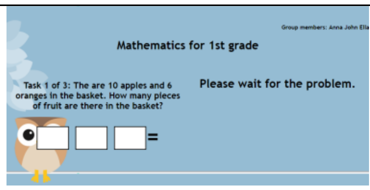
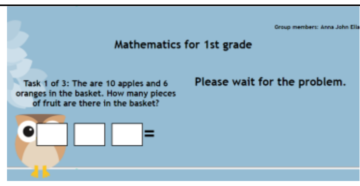
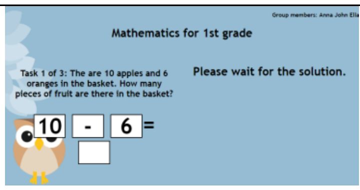
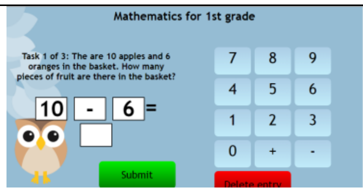
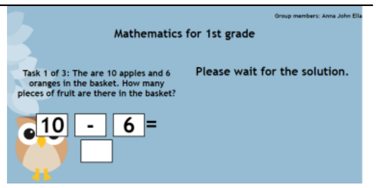
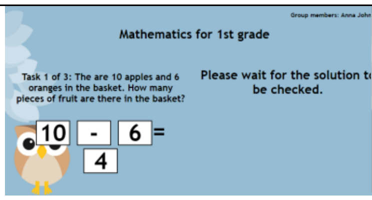
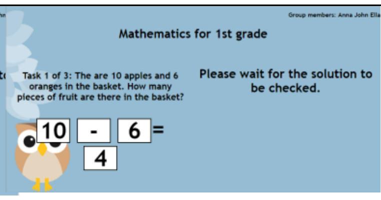
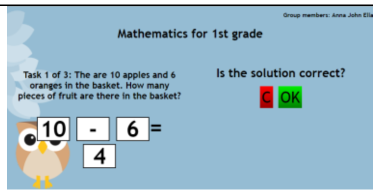
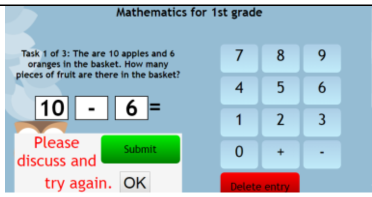
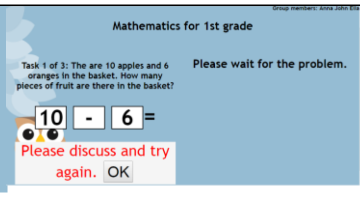
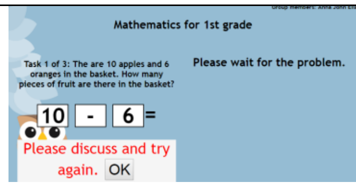
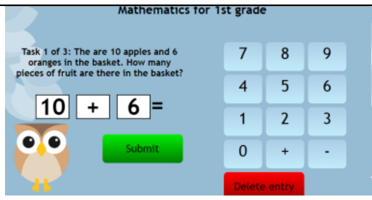
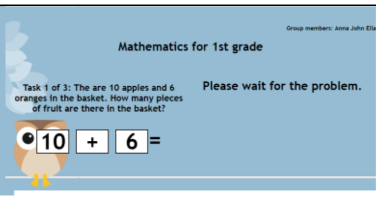
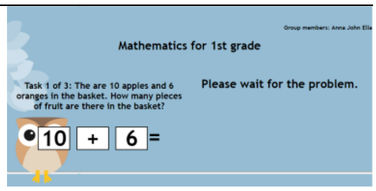
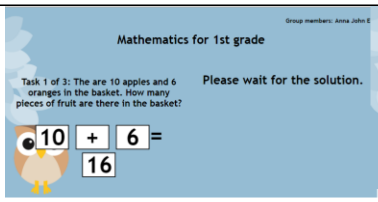
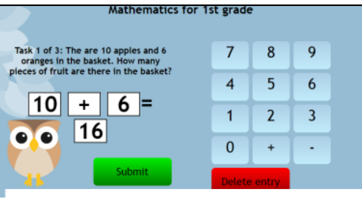
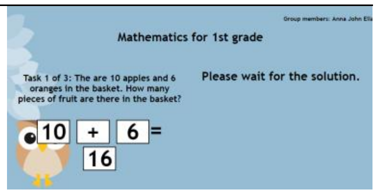
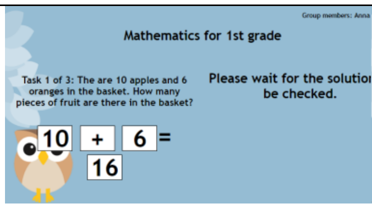
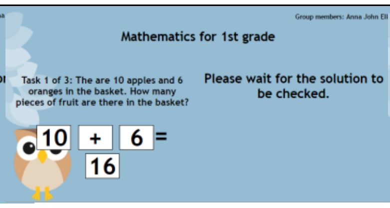
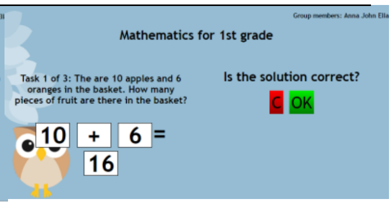
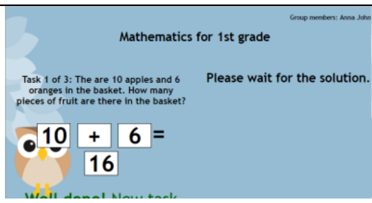
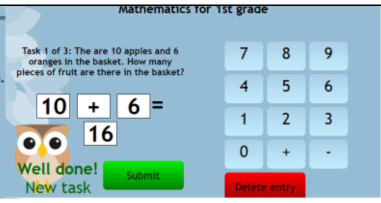
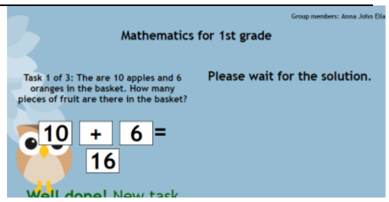
In order to utilize and to pilot the SCOLLAm project infrastructure and the Author system, two widgets for learning Math were created: one for learning early grade Mathematics in pairs and the other for learning in triplets. Since widgets differ significantly in problem presentation, the process students are to exhibit when participating in the learning activities with widgets and the roles students are assigned when participating in the learning activity.

Table 2. The process of solving a mathematical problem via the pairs synchronous collaboration widget

Step	*The Editor role*	*The Checker role*
1		
2		
3		
4		
5		
6		

* Steps 2 and 3 appear only in the case of wrong solution offered

Table 3. The process of solving mathematical problems via the triplets synchronous collaboration widget

Step	*The Author role*	*The Editor role*	*The Checker role*
1			
2			
3			
4			
5			
6			
7			
8			

* Steps 2, 3 and 4 appear only in the case of wrong solution offered

Table 2 illustrates the process of solving a mathematical problems in different roles where two students get assigned the roles of Editors and Checkers and participate in problem solving task. Similarly, Table 3 displays the triplets widget in which three students get assigned the roles of Author, Editor and Checker to participate in a problem solving task.

4. Discussion

The case study demonstrated two educational widgets for computer supported collaborative learning used in early primary school mathematics education. The first widget implements working in pairs in two roles: editors and checkers, which means students participate in task assuming various roles. On the other hand, the second widget is for working in triplets where the process is more complicated since a new role (author) is introduced. Additionally, tasks for working in triples are more high level, where the author student role had to translate the textual assignment into a mathematical expression. The two widgets differ in terms of graphical user interface and the implemented process, which is all done by the widget creator. When it comes to communication with the server side communication feature of the SCOLLAm system, the differences are minor since the widget creator is making a few interface calls more in order to support one role more.

The experience in using the widgets and the SCOLLAm system was gathered via a series of experiments taking place in years 2016 and 2017 in the 1st and 2nd grade in a primary school in Croatia. The widgets were co-designed with the primary school teachers and the task contents were provided by the teachers themselves so that they are aligned with the curriculum and the pace teachers take in delivering their lessons. Preliminary results show that the increased complexity of working in triplets on more difficult and abstract tasks affects the ways young learners use the system leading to more engaged and on task behaviour. This confirms that the special care should be paid to the alignment of the digital lesson tasks with the curriculum implementation flow.

Implementing synchronous collaborative learning solutions does come with a need of ensuring available stable connection throughout system usage and possible contingency when system operation becomes less stable. As part of the case studies, mobile 4G routers were used with 8 children sharing one mobile 4G router when participating in the activity. In most of the performed tasks this turned out to be sufficient, but on certain occasions, seemingly non-deterministically, this proved to be a challenge because mobile network signal dropped or the service level became unsatisfactory. This leads to two lessons learned: contingency in terms of additional non-connected mobile learning content needs to be provided to bridge the idle time of young learners and the learning systems need to be designed so they are able to continue operation from the exact point when the lesson was interrupted.

5. Conclusions

Synchronous collaborative mobile learning activities can be designed in a way they are added to current content via the central server infrastructure. By providing an interface consisting of two method calls, widgets for working in pairs and triplets were implemented and the following major experiences gathered: collaborative learning contents should be aligned with the curriculum, there should be contingency and recovery planned in the case of connection failures, and the complexity of different roles should be aligned to ensure adequate digital learning activity dynamics.

Acknowledgements: This work has been fully supported by the Croatian Science Foundation under the project UIP-2013-11-7908. The authors would like to thank the staff of Primary School Trnjanska - Ivančica Tajsl Dragičević, Kristina Vlah, Luci Plenković Omerso, Jasna Haraminčić, Petra Crnjević Trstenjak and Nada Šimić - for their enthusiastic partnership in the realization of the study presented in this paper.

References

Boticki, I., Baksa, J., Seow, P., & Looi, C. (2015). Usage of a mobile social learning platform with virtual badges in a primary school. *Computers & Education*, *86*, 120–136. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360131515000688

Boticki, I., Wong, L. H., & Looi, C.-K. (2013). Designing Technology for Content-Independent Collaborative Mobile Learning. *IEEE Transactions on Learning Technologies*, *6*(1), 14–24. https://doi.org/http://doi.ieeecomputersociety.org/10.1109/TLT.2012.8

Caballero, D., van Riesen, S. A. N., Álvarez, S., Nussbaum, M., de Jong, T., & Alario-Hoyos, C. (2014). The effects of whole-class interactive instruction with Single Display Groupware for Triangles. *Computers & Education*, *70*(0), 203–211. https://doi.org/http://dx.doi.org/10.1016/j.compedu.2013.08.004

Chen, C.-M., & Chen, F.-Y. (2014). Enhancing digital reading performance with a collaborative reading annotation system. *Computers & Education*, *77*(0), 67–81. https://doi.org/http://dx.doi.org/10.1016/j.compedu.2014.04.010

Jagust, T., Mekterovic, I., & Boticki, I. (2015). The experiences of setting up, developing and implementing a mobile learning project in Croatia: The SCOLLAm project. In *Proceedings - Frontiers in Education Conference, FIE* (Vol. 2014). https://doi.org/10.1109/FIE.2015.7344160

Sun, D., Looi, C.-K., & Xie, W. (2014). Collaborative Inquiry with a Web-Based Science Learning Environment: When Teachers Enact It Differently. *EDUCATIONAL TECHNOLOGY & SOCIETY*, *17*(4), 390–403.