Mobile Learning System for Enabling Collaborative and Adaptive Pedagogies with Modular Digital Learning Contents

Tomislav Jagušt, Ivica Botički

University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia

I. INTRODUCTION

The use of mobile technologies can be beneficial towards learning, in terms of improved student cognitive results, engagement, interest and motivation. A plethora of mobile technologies have been used to support the learning process, from simple note taking devices and classroom response systems to the virtual reality or augmented reality applications for tablet computers engaging students in cognitive and psychomotor domain, encouraging them to move about the environment and to communicate with the surroundings and fellow students to solve a given task.

Novel pedagogies advocate for the use of personalization and adaptation to individual students and collaborative learning with the final aim of enhanced learning and gained skills (Alvarez, Alarcon, & Nussbaum, 2011; Brusilovsky & Peylo, 2007). Such approaches, without any doubt, can be implemented without the use of technology, but the presence of technology allows for a more efficient and convenient delivery and implementation of such efforts. Adaptive learning can therefore appear in situ and utilize the most relevant and recent student data, while collaborative groups get formed instantaneously upon teachers' request. This has led to the development of recommender systems, adaptive hypermedia, and personalization of the learning experience and creation of personal learning environments - PLEs. Collaborative and cooperative learning are no exception, with researchers demonstrating a variety of strategies for group tasks, dividing student into groups, exploring dynamic groups creation and different models of scripting group work leading to the improved learning (Järvelä et al., 2014; Looi et al., 2016; Stahl, Koschmann, & Suthers, 2006).

The majority of studies in the field of adaptation and collaboration or cooperation in learning are built using open or publicly available technologies or are set up as a custom solution tailored for the specific purpose of supporting a research intervention or digitalize a specific part of curriculum to be used by the teachers (for example, a mathematics lessons on learning addition and subtraction designed as a standalone mobile application). Such approaches lead to limited system reusability, and their further development is hindered by the specifics of the target platform and the fact that the digital learning content is interwoven with the learning application. Such solutions typically require a substantial amount of work and resources to be extended with the use of novel pedagogies such as the adaptive, collaborative and cooperative learning in case they do not originally focus on directly supporting them. This is mainly due to the fact that introducing support for novel pedagogies into existing technology enhanced learning solutions usually means making significant changes on the existing internal systems' databases containing learner-oriented data.

In this study, we propose system design addressing the following challenges: (1) the definition of modular data organization and common interfaces that allow for packaging new and existing digital learning contents to support new pedagogies (2) the provision of system-embedded synchronous real-time message exchange mechanisms allowing for learning through collaboration and cooperation between students across multiple platforms and devices, and (3) the mechanisms allowing system components and learning contents adaptation to students, utilizing both real time and long term data.

A systematic solution addressing the three stated challenges is proposed, implemented and trialed as part of a research project on seamless mobile learning in early primary schools in Croatia. To demonstrate its applicability to extending the existing digital learning contents with novel pedagogies support, the existing digital lessons in the area of mathematics are examined as part of three case studies: the first study using a basic mathematics digital learning lesson, the second one extending the basic lesson using adaptation to learners, and the third extending the basic lesson with the use of collaborative mathematics learning.

II. Theoretical Background

A. *Systems for Technology Enhanced and Mobile Learning*

Mobile learning aims for student participation in learning processes at any time and location (O'Malley et al., 2005) and seamless switching between learning contexts (Sharples et al., 2015; Wong & Looi, 2011). Mobile learning systems offer high levels of interactivity, constant connectivity and easy communication with other learners, can utilize sensor data and provide content depending on the context in which learning takes place (Wald, Li, & Draffan, 2014).

Existing systems or frameworks for mobile learning are built with the aim of fulfilling a particular goal and are scarce in features and functionalities targeted at all possible learning scenarios (Martin et al., 2011). In spite of that, they share some common architecture and design characteristics: the client-server architecture model (Alvarez et al., 2011; Su & Cheng, 2015; Wen & Zhang, 2015), designed separate applications for students and teachers (Su & Cheng, 2015; Wen & Zhang, 2015), and a lesson data model typically including multimedia. Some authors put focus on content presentation on a variety of devices (Zhao & Okamoto, 2008), the design process of a learning system (Holenko Dlab, Hoic-Bozic, & Boticki, 2017) or on supporting a single prevailing pedagogy (i.e. student collaboration) (Alvarez et al., 2011). The others give special attention on learning analytics including data logs, the accompanying analysis and insights (Mutahi, Bent, Kinai, Weldemariam, & Sengupta, 2015), systematic use of gamification (Su & Cheng, 2015) or augmented reality (Chiang, Yang, & Hwang, 2014).Such a variety and fragmentation is noted in a review study by Martin et al. highlighting the need of a systematic approach to designing a learning system, middleware or framework that would take care of device and platform specific implementation challenges, allowing lesson designers to focus on lesson content and learning outcomes implementation (Martin et al., 2011).

A recent review of existing research literature in the field, conducted by authors (Jagušt, 2015), revealed 47 studies that address the topic of designing and developing mobile learning systems. The review has shown strong growth of the number of Android and iOS based mobile learning solutions, with Java and Objective-C being the most popular programming languages. Even though they account for only about 10% of the analyzed studies, web technologies show strong growth, especially in combination with hybrid application development frameworks like Apache Cordova or Xamarin. The prevailing architecture model is client-server (43% of all studies), with over 72% of systems using mobile or wi-fi network for real-time communication and data transfer.

Having the results of this review study in mind, we designed and proposed a mobile learning system that follows the trends in the field, and at the same time offers modularity, reusability and exposes a set of common functionalities that can be used across different learning lessons, in different scenarios and environments.

B. *Adaptivity*

Adaptive learning or learning that adapts to the needs and abilities of a learner is one of techniques that provide personalized learning experience, with the goal of optimizing learning efficiency, increasing student engagement and addressing specific needs, interests or preferences of each learner (Hsiao, Sosnovsky, & Brusilovsky, 2010). Adaptivity can be achieved in different forms, by adapting visual representation or appearance of the learning material, adapting the content and its difficulty or adapting the navigation and learning pathway (order or sequence of learning contents) to best fit the needs of each individual learner (Brusilovsky & Peylo, 2007). Many existing solutions focus on adaptation of learning pathways or the difficulty levels of given lessons and problem sheets, with only some systems entailing more complex adaptation models.

Adapting learning processes or contents to the needs and abilities of specific students remains one of the most discussed topics in the field of technology enhanced learning and the source of debate until today (Kirschner, 2017). It is advocated to help balancing different abilities, learning speed, skills, and to improve student engagement, interest and knowledge (Shute & D, 2012). A number of research studies explored diverse aspects of adaptivity and personalization in mobile learning, from systems that adapt to student (El Guabassi, Bousalem, Al Achhab, JELLOULI, & EL Mohajir, 2018; Madhubala, R., & Akila, 2017; Zhao, Anma, Ninomiya, & Okamoto, 2008; Zhao & Okamoto, 2008) to the systems that adapt to the available technology, learning environment and context (Luo, Yang, & Wei, 2017; Syvänen, Beale, Sharples, Ahonen, & Lonsdale, 2005).

In this study the adaptive learning techniques were implemented using two types of adaptivity: the pre-lesson adaptivity based on students' previous results in the same subject (Mathematics) and in-lesson adaptivity based on students' immediate responses. These features were implemented in the digital lesson through the Adaptivity and Aggregate Data Subsystem described in chapter IV E.

C. Collaborative Learning using Mobile Devices

The use of connected mobile devices during computer supported collaborative learning (CSCL) activities lets students to move around the learning environment and interact while completing a joint task (Zurita & Nussbaum, 2007). Asynchronous CSCL utilizes tools such as the online forums and message boards that can be accessed at any location and time of convenience. On the other hand, synchronous CSCL happens when learners engage in learning in a specific time-frame, but can, depending on the technology, be positioned at various locations (Bower, Dalgarno, Kennedy, Lee, & Kenney, 2015).

Researchers have been exploring modes of structuring interaction appearing in using asynchronous and synchronous CSCL indicating that interactions in asynchronous CSCL environments bring complexity to learning processes and more guidance to students is needed to promote better conceptual understanding (Fita, Monserrat, Moltó, Mestre, & Rodriguez-Burruezo, 2016; Sun, Looi, & Xie, 2014). Therefore the blend of asynchronous and synchronous CSCL (Perveen, 2016) or a shift towards the synchronous CSCL is advocated for (Fita et al., 2016).

Synchronous mobile computer supported collaborative learning (mCSCL) activity design employed in this study includes identification and specification of the mechanisms for structuring synchronous interaction pattern via roles management (Strijbos & Weinberger, 2010), group size (Veerman & Veldhuis-Diermanse, 2001) and composition that will consequently determine the activity interaction patterns employed by the students.

III. Challenges In Modularizing And Extending Digital Learning Contents For The Use In New Pedagogies

A. First Challenge: Modularizing Multimedia and Interactive Digital Lessons

Packaging multimedia learning contents should take into account structuring a variety of media content elements (i.e. pictures, videos, sounds), their grouping into modular elements (i.e. slides or pages that tell a story connecting the media elements), sequencing of modular elements to support lesson flow and the interactive elements that are embedded as part of the modular elements. The modules should be stand alone and in the same time have all the necessary interfaces to be connected with the other modules to form lessons to be further used by the students and teachers.

A structured approach to organizing digital content elements and encapsulating the existing educational contents to be used in a modular way brings potential value to the educational contents that would otherwise be neglected in terms of maintenance and technology change. Furthermore, having well-defined modular contents design allows for extension of the content elements via interfaces that support new pedagogies. New interfaces could add the support for new pedagogies on top of the existing modules or collection of modules in a systematic fashion, relieving content designers of difficulties and idiosyncrasies these extensions presuppose.

In this paper we advocate for an approach where system architecture becomes an enabling key element in extending digital learning lessons. It would allow for utilizing the shared system logic and APIs to enrich the existing digital contents with new pedagogies.

B. Second Challenge: Enabling Synchronous Collaboration across Platforms

Developing digital learning lessons which include synchronous collaborative and cooperative learning pedagogies usually entails great complexity due to the fact that technology for consuming such content needs to be designed in a performant way and scale across multiple platforms. That way learners can engage with the digital contents by performing actions and adding or modifying content in real time, thus completing a collaborative or cooperative task in an interdependent way.

Designing activity utilizing synchronous collaborative or cooperative learning with technology is dependent on background information on the participating students', desired grouping algorithms and the information on collaborative and cooperative learning task and the way it is scripted to allow all group members to participate equally and meaningfully. The system supporting such activities needs to be able to generate collaborative or cooperative groups on demand, to maintain the state of collaborative activities across multiple groups, restore state in case of activity interruptions and maintain fast synchronicity at all times. It is likely that longer delays or the need for the learners to reload or refresh the digital learning contents due to the technical issues would contribute to lower levels of students' involvement and raise disinterest in such activities (Holenko Dlab, Boticki, Hoic-Bozic, & Looi, 2017).

As part of this paper a system for supporting synchronous collaborative and cooperative pedagogies in digital learning contents is presented, with the functionalities for automatized group creation, synchronous group messages creation and delivery and group activity state restoration.

C. Third Challenge: Adapting to Students in Real Time and Long Term

Implementing adaptivity includes the design of two main components, a component that collects and processes information on adaptivity (the data analysis component), and the component that implements the adaptivity (the lesson adaptation component). Often, these two components are bound into a single adaptation engine, reducing flexibility of such a system and locking it to a single adaptivity implementation.

In adaptive systems multiple adaptation models could be enabled ranging from basic (e.g. adapting difficulty level in the beginning of the lesson) to complex adaptation techniques within and between lessons. Adaptation within a lesson is to be provided through real-time student progress monitoring using an analysis engine that estimates student performance and compares the individual student score with the score of other students, giving feedback in regard to the lesson elements, changing lesson elements' difficulty levels, adapting existing contents by adding help or guidance etc. For long term adaptation, a system should analyze usage logs and extract long term data describing student learning behavior in a longer period of time.

In this paper we propose an adaptive system with decoupled components, centralizing the data analysis engine as the core part of a learning system and leaving the adaptivity part to be utilized by the digital learning lesson via the designed interfaces.

IV. SYSTEM ARCHITECTURE FOR MODULAR AND EXTENSIBLE DIGITAL LEARNING LESSONS

A. System Architecture Overview

The system architecture proposed in this study consists of three parts: the Central server and the two client applications for use by students and teachers (schematic representation of the system is given in Fig. 1). In addition to the Communication subsystem, the Central server includes a database, a multimedia content repository, an event log subsystem, the Adaptivity and aggregate data calculation subsystem supporting the adaptivity pedagogy and a Groupwork and group communication subsystem supporting the collaborative and cooperative group work pedagogy.

Mobile application used by the students (left hand side of Fig. 1) is designed as a multiplatform hybrid multilayer application (Rahul Raj & Tolety, 2012; Xanthopoulos & Xinogalos, 2013). It consists of the Communication subsystem implementing communication with the server, the Subsystem for digital learning lessons delivery ensuring that a digital learning lesson is delivered to a student, the Groupwork delivery subsystem for the delivery of collaborative and cooperative activities and the Progress monitoring subsystem for collecting log data to be used by the Central server for a variety of purposes, including adaptation algorithms operation.

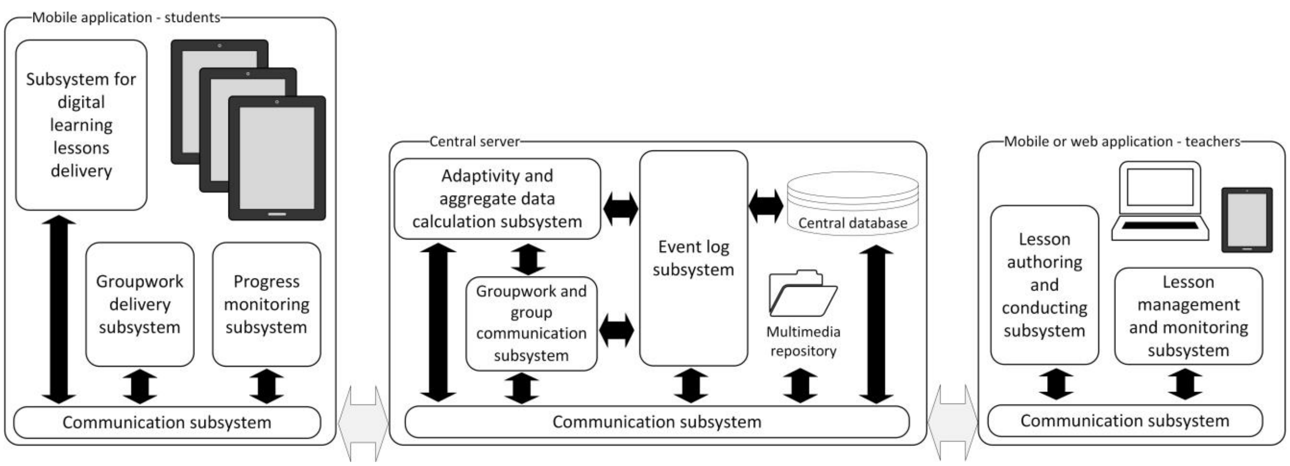


Fig. 1. Schematic representation of the mobile learning system architecture

The Lesson authoring and conducting subsystem (as a part of the web application for teachers) is logically composed of multiple components depicted in Fig. 2, with the basic information on each of the component given in Table 1. Components are grouped into several component groups according to their purpose.

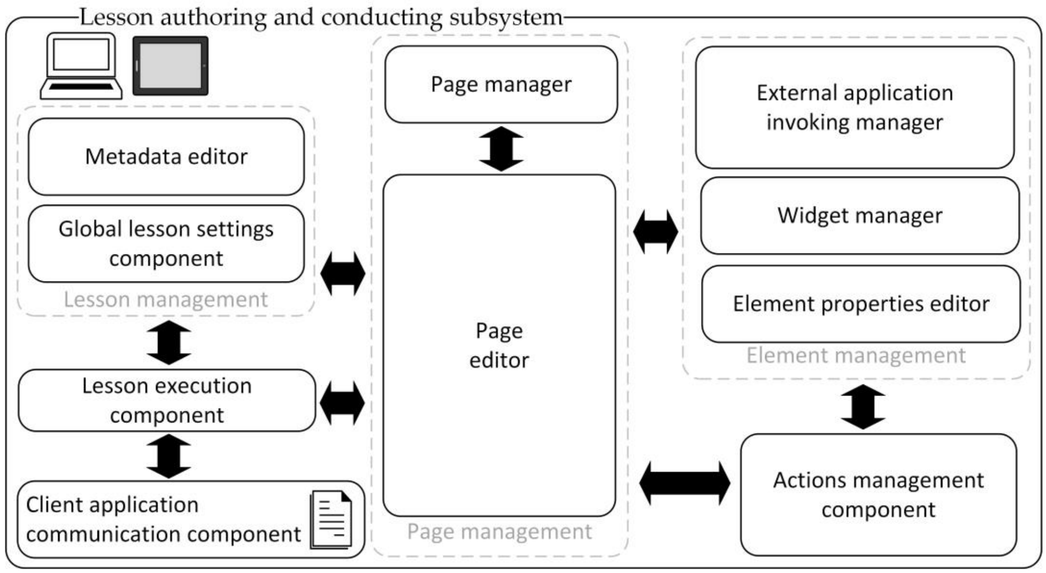


Fig. 2. The internal components comprising the Lesson authoring and conducting subsystem

The Lesson management component group deals with configuring and describing a digital learning lesson, Page management group ensures proper rendering of multimedia digital lesson contents to a student, while the Element management component group deals with configuration and customization of concrete page elements (such as setting parameters of the text, image or programmed widget components).

Subsystem for digital lessons delivery is a digital module (lesson) player invoked typically by native mobile student applications on tablet computers. The component loads an educational lesson and all its parts, retrieves configuration parameters for starting the lesson including current user data, runs it and controls the course of the educational lesson etc. During the execution of the module (lesson), the Subsystem for digital lessons delivery executes user actions (i.e. “navigation to the next slide”, “start widget”), monitors global variables and liaises with the Progress monitoring subsystem to keep track of the lesson state and to send the detailed usage log to the Central server Event log subsystem via the Communication subsystem.

TABLE I

COMPONENTS OF THE LESSON AUTHORING AND CONDUCTING SUBSYSTEM ORGANIZED INTO COMPONENTS GROUP AND THEIR RESPONSIBILITIES

Component Group	**Component Name**	**Component Responsibility**
Page management	Page manager	Navigation through lesson modules/pages, overview of the lesson. Entry point for adding, deleting or repositioning of modules/pages.
	Page editor	Visual page editor for manipulating module/page elements.
Element management	External application invoking manager	Definition of external application links and parameters used to run external applications installed on students’ devices. Depending on the platform type Universal Links (iOS) or App Links (Android) are used as supporting technologies.
	Widget manager	Insertion and configuration of programmed elements (widgets).
	Element properties editor	Context-based menus appearing upon selection of a module/page or a single module/page element. Used to edit visual properties (color, font, transparency etc.) or to set programmed element (widget) start parameters. Separate parameters can be set for each widget instance across digital lessons or across modules/pages in a single digital lesson.

Lesson management	Global lesson settings component	Configuration of digital lesson type, title, visibility and definition and initialization of global variables.
	Metadata editor	Setting digital lesson metadata in Dublin Core[1] or IEEE Learning Object Metadata (IEEE LOM)[2] formats.
Lesson execution component		Starts and controls lesson execution.
Actions management component		Contextual menus that allow for definition of actions to be executed upon typical events. For example, jumping to a digital lesson page after an "element is dropped onto another" action is detected.
Client application communication component		Communication message exchange between components and applications.

B. Modules and Elements as Key Digital Lesson Building Components

In the proposed model, digital lessons are designed recursively where each lesson module is a container that consists of a series of lesson sub-modules (referred to in this paper as pages). Fig. 3 illustrates basic relationship between lesson modules, sub-modules (pages), and lesson elements.

A lesson sub-module contains basic elements or a group of basic elements. Each basic element can be reused multiple times in the same slide, lesson, or in a different lesson. Such reuse requires the design where each element instance occurrence contains its own specific set of properties such as the size, position and input parameters. There are three kinds of basic elements:

- passive elements (e.g. picture or text)
- interactive elements (button, drawing canvas, text input field etc.)
- widgets - programmed interactive elements (e.g. an animation or a game).

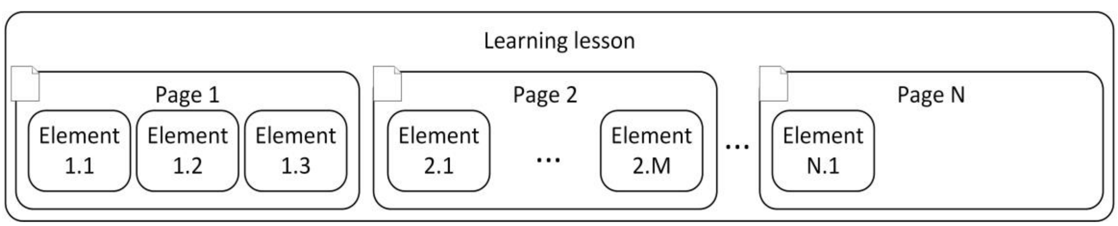


Fig. 3. Relationship between lesson modules, lesson sub-modules (pages) and lesson elements

The Lesson Authoring and Conducting Subsystem is a visual digital module (lesson) editor, based on the What You See is What You Get (WYSIWYG) principle. It allows for editing and arrangement of digital lesson sub-modules (pages) and elements. Elements are positioned in relative relation to each other on a single page, allowing for size scaling for different rendering on different delivery platforms, from desktops to tablets or smartphones. Element dimensions are stored as relative values, and real dimensions are calculated at runtime in relation to the size of the container within which they are rendered.

[1] http://www.dublincore.org/
[2] http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1032843

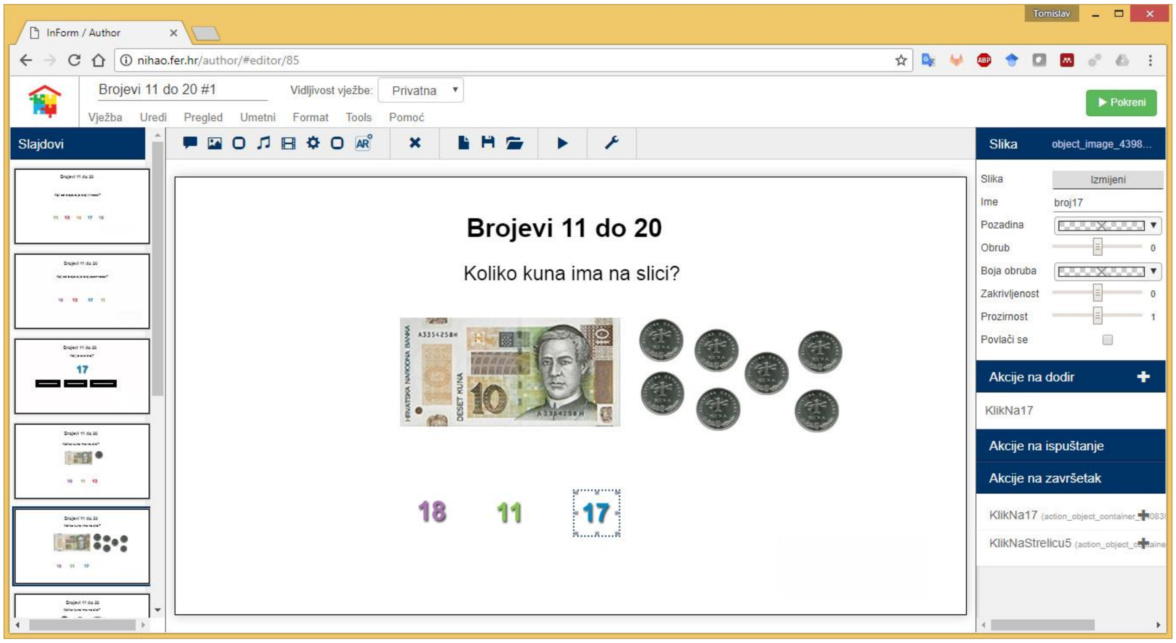


Fig. 4. Visual digital module (lesson) editor as an implementation of the Lesson Authoring and Conducting Subsystem

On the left-hand side of the visual digital module editor is a compact page carousel, allowing for quick overview and reorder of digital lesson sub-modules (pages), while the right-hand side of the sub-module editor consists of context-dependent menus, presented in form of windows, allowing users to define and edit properties of the currently selected sub-module or individually selected elements. Standard navigation through sub-modules involves switching to the next or previous sub-module and jumping to an arbitrary sub-module within the module.

The Actions Management Component allows for definition of actions that affect standard module behavior. An action can be defined on a sub-module or element level affecting module execution flow depending on user interaction. The available actions include the dynamic (real-time) change of element properties (shape, color or size), element move, and change of a lesson sub-module or modification of a global variable. For example, if an action defining a lesson completion condition is met, user is allowed to move to the last page, alternatively if the completion condition is not met the user gets positioned to the lesson start (such an exemplar scenario is depicted in the schematic diagram shown in figures Fig. 5 and Fig. 6).

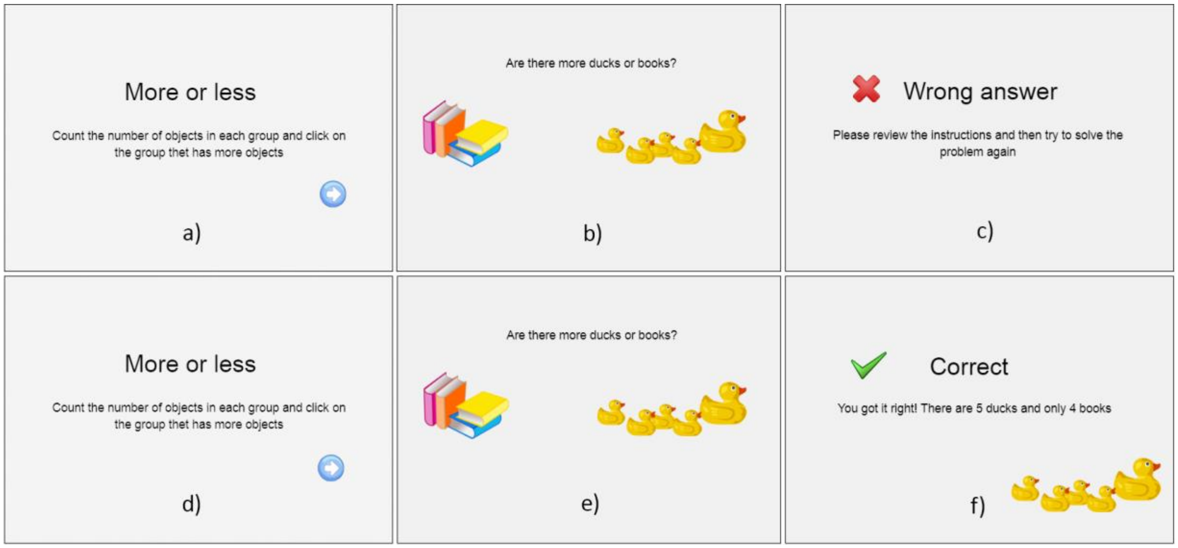


Fig. 5. An example of actions – a sub-module with a description of a task (a) is followed by a question submodule (b), where a user chooses a wrong answer and is sent to the "wrong answer" page (c) and then is returned to the first sub-module in a lesson (d).

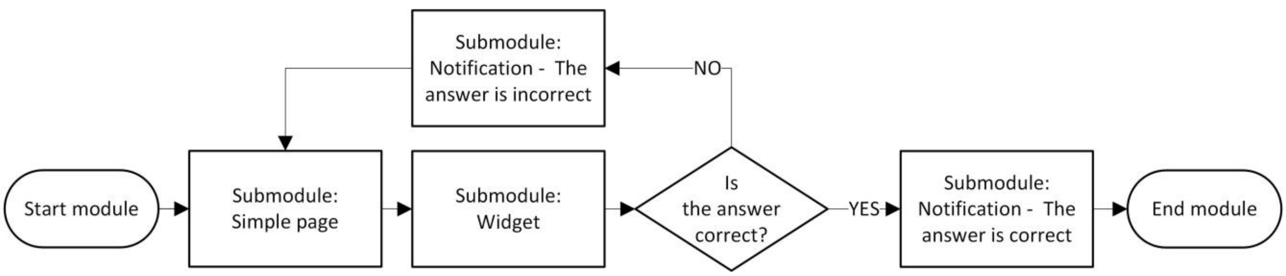


Fig. 6. An exemplar scenario of using actions as part of the Actions Management Component – flow of the module (lesson) is modified depending on a user interaction input

C. Lesson Authoring and Conducting Subsystem - Encapsulating and Reusing Existing Applications with the Widget manager component

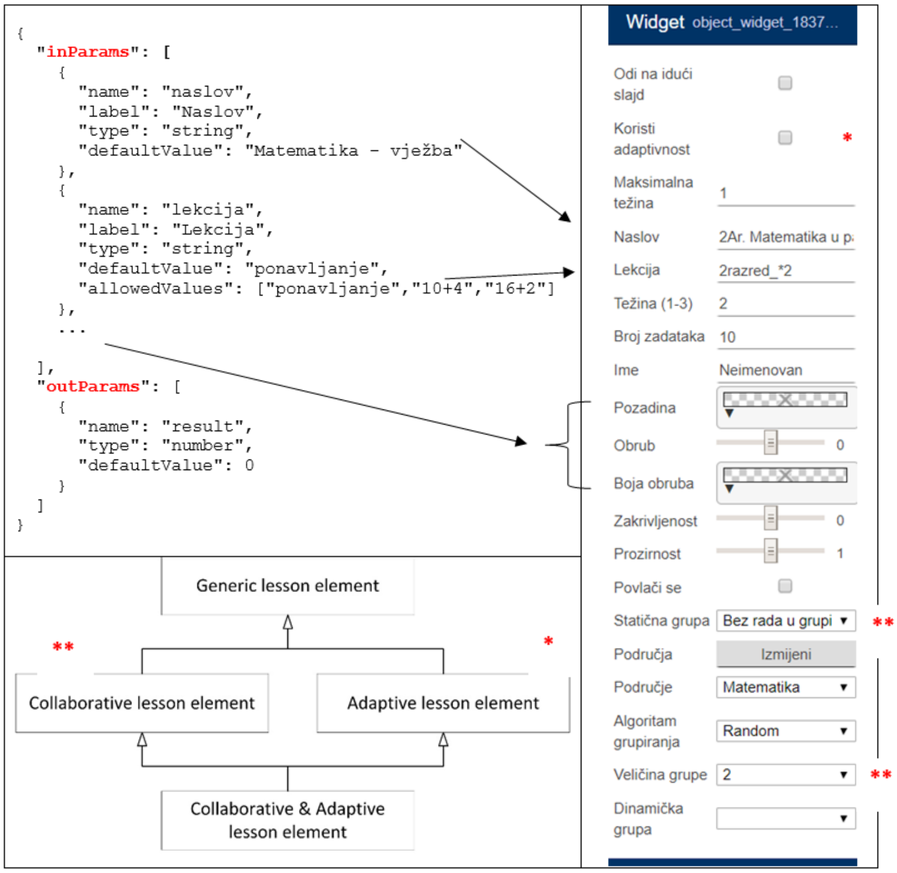
Widgets are programmed digital lesson elements that can be used as part of the digital lesson sub-modules (pages) like the regular simple elements (i.e. text or image). Widgets provide an addition to educational lessons enriching them with interactive contents. As with other basic elements, widgets have a set of properties (such as size and position on the page) and their own configuration parameters which are passed to the widget by the lesson player in runtime.

The process of using a widget typically includes using an existing educational web application which is converted into a widget by following two simple rules: producing a widget parameters file and creating a single archive (i.e. zip) file with the parameters file and all other web application specific files. Implementation-wise, the authors of a widget have to make sure that the generated content is functional and can be rendered in a web browser. Such a file package is then uploaded into the system using the Lesson authoring and conducting subsystem and positioned onto the rectangular frame, which is a placeholder representing a lesson sub-module (page) where the widget will be loaded at the runtime, once the sub-module gets shown to a student using the digital lesson.

Widget parameters and their type are defined at the widget level in a specially formatted parameters params.json file by the widget authors. The Lesson authoring and conducting subsystem parses the parameters file and dynamically renders visual representation of the widget input parameters. Widget params.json file depicted in Table 2 includes two main sections: the inParams and the outParams sections, where the former can contain as many properties as needed while the latter contains only one property. Each parameter can specify name, label, type, defaultValue and allowedValues properties that are rendered as part of widget instance configuration in the Lesson Authoring and Conducting Subsystem.

TABLE II

WIDGET PARAMETERS AND CONFIGURATION



An example JSON params.json file describing widget parameters (upper left-hand side) and its rendering in the Lesson Authoring and Conducting Subsystem (upper section on right-hand side). Lower left-hand side: polymorphic nature of digital lesson elements – base element, adaptive and/or collaborative elements, lower right-hand side: widget element configuration properties for activating adaptivity and collaboration features.

Each lesson module is designed as a polymorphic structure, where the generic lesson element serves as a base for adaptive and/or collaborative elements, depending on the pedagogies the module designer wishes to implement. The concrete type is determined by adjusting module properties and is automatically applied at run time (indicators * and ** in Table 2). For example, selecting "Koristi adaptivnost" (Use adaptivity) will activate system adaptivity features for the selected module. Similarly, selecting a value from the "Staticna grupa" (Static group) list or from the "Algoritam grupiranja" (Grouping algorithm) list will activate system collaborative features for the selected module."

D. Groupwork and Group Communication Subsystem - Implementing Collaborative Learning Support

Groupwork and Group Communication Subsystem is typically used by widget creators to include designs with collaborative and cooperative learning into digital lessons. Widgets authors use already defined and implemented communication interfaces as part of the Groupwork and Group Communication Subsystem in their widget code to utilize the synchronous infrastructure for group learning. Such an approach of encapsulating synchronous communication via the well-defined interfaces allows for more efficient and less error prone communication due to the reusability of the system communication infrastructure. The communication interface methods to be reused for implementing synchronous communication are presented in Table 3.

Using the visual digital module (lesson) editor, a module element can be configured to use the Groupwork and Group

Communication Subsystem functionalities and APIs therefore enabling group work. Fig. 7 depicts the following parameters: “Staticna grupa” (Static group) used to specify if an already existing predefined group is to be used in a group work activity, “Podrucje” (Content area) used to specify the content area for the group activity, “Algoritam grupiranja” (Grouping algorithm) to specify if the groups will be created using a random, heterogeneous or homogeneous group creation algorithm (according to knowledge levels of students in the specified content area), “Velicina grupe” (Group size) the size of groups to be generated and “Dinamicka grupa” (Dynamic group) the input list of students that are organized into groups.

TABLE III

SYNCHRONOUS COMMUNICATION METHODS AS PART OF THE GROUPWORK AND GROUP COMMUNICATION SUBSYSTEM USED BY WIDGETS IMPLEMENTING GROUP LEARNING DIGITAL LESSONS

Group communication Sub-Module	**Method name and parameters**	**Method description**
Activity Management	onUserInfo	Listens to detailed information on user from the server
Activity Management	ready	Notifies the system that widget registered all listeners and is ready for message receival
Message Forwarding	postGroupMessage	Sends a message to all group members
Message Forwarding	onGroupMessage	Listens for an incoming group message and acts upon
Message Forwarding	onMessagesFromBackend	Listens for received but already saved messages from the server
Message Forwarding and History Storage	postPersistentGroupMessage	Sends a message to all group members and saves the message on the server
History Storage	updateStateOnBackend	Saves custom (any data) state on the server
History Storage	requestState	Requests state from the server
History Storage	requestMessages	Requests already saved messages from server
History Storage	onStateFromBackend	Listens for a state change from the server
History Storage	getState	Combination of onStateFromBackend + requestState for easy state fetch from the server

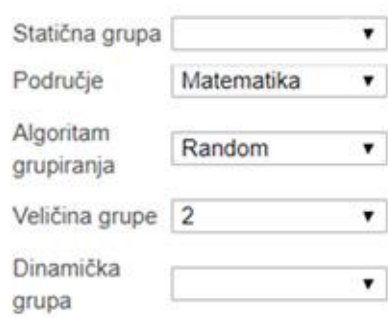


Fig. 7. Specifying digital lesson element parameters to be utilized by the Groupwork and Group Communication Subsystem to provide group work functions

Groupwork and Group Communication Subsystem is part of the Central server that takes care of synchronous message exchange in real-time, forwards the messages to clients and stores and manages information on collaborative and cooperative work of students. The subsystem is composed of three main components as illustrated in Fig. 8: 1) the Activity Management component which deals with recording user participation and stores users and groups that are currently active in the group communication activities, 2) the History Storage component storing the ordered sequence of all communication messages and group states (i.e. message parameters) from the activity onset and 3) the Message Forwarding component for distributing messages to other active participants.

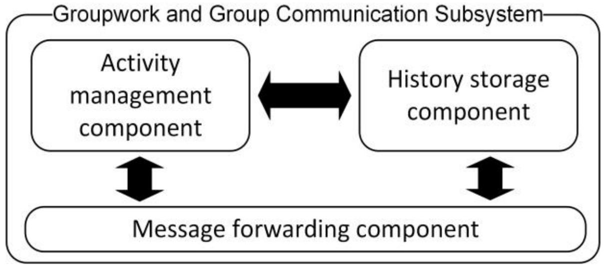


Fig. 8. The three main Groupwork and Group Communication Subsystem components (Activity Management, History Storage and Message Forwarding components)

E. Adaptivity and Aggregate Data Subsystem - Implementing Adaptive Learning Support

The Adaptivity and Aggregate Data Subsystem is a Central server component that collects students' usage data and performs calculations and aggregations in order to calculate input values to be used in adaptation strategies implemented by lesson elements. It liaises with the Event log subsystem to extract information for performing adaptation on the client side. The subsystem API methods are used to fetch metadata and performance data on students and the current learning lesson, e.g. user score in relation to other users that participated in the same learning lesson or the average score of all users participating in one lesson. This data can be used to implement adaptation strategies, including adjusting the complexity of tasks, adjusting the available time to complete a task, adjusting visual representation or a lesson type to students etc. A list of available Adaptivity and Aggregate Data Subsystem API methods is shown in Table 4, while the underlying database model for supporting the Adaptivity and Aggregate Data Subsystem logic is shown in Fig. 9.

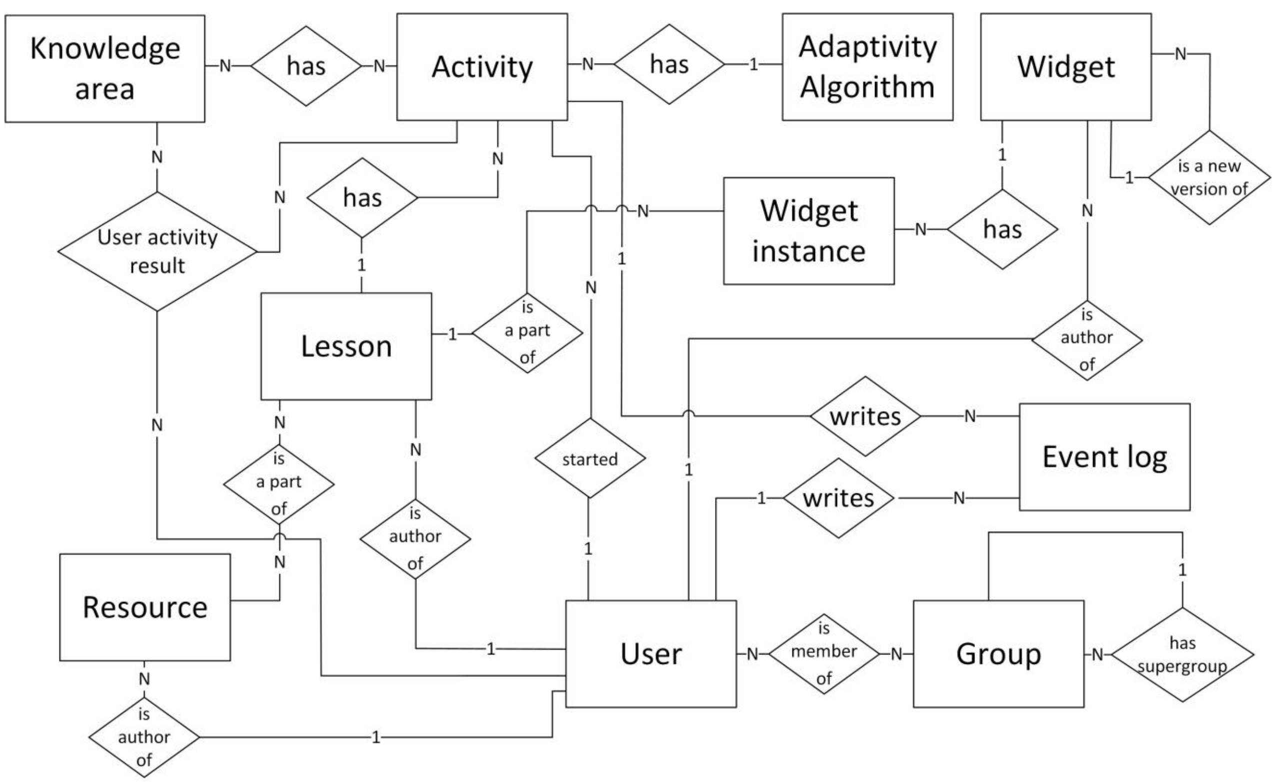


Fig. 9. System data model with the Adaptivity, Collaboration and other entities

TABLE IV

ADAPTIVITY AND AGGREGATE DATA SUBSYSTEM API METHODS

method	Description
GetAdaptivityKnowledgeAreas	Returns a list of knowledge areas (e.g. “Mathematics” or “Mathematics – multiplication with number 2”) according to which a module can adapt
GetUserLevel	Returns a combined knowledge level of a given user, calculated with an algorithm specified as a parameter (for the list of preselected knowledge areas)
GetGroupLevel	Similar to GetUserLevel but returns a combined knowledge level of a group of students
GetUserTaskTime	Returns mean time a user spent on solving a single task in a list of specified activities
GetUserPerformanceTrend	Returns trend of user performance during a period of time for selected activities. The trend is calculated as a relationship of a number of tasks solved in consecutive discrete time intervals during each of the specified activities.

Using the visual digital module (lesson) editor, a module element can be configured to use the Adaptivity and Aggregate Data Subsystem functionalities and APIs enabling adaptation. Fig. 10 depicts the following parameters: “Koristi adaptivnost” (Use adaptivity) used to specify if the adaptivity functions should be provided to the selected digital lesson element, “Podrucja” (Content areas) used to specify the content areas according to which adaptivity values (i.e. knowledge levels) are calculated by the system. Content areas are defined via separate feature allowing for the selection of multiple areas (“Podrucje”), retroactive days taken into account for an area (“Broj dana za promatranje”) and weights for an area-retroactive days pair (“Tezina”). Such modelling allows the Adaptivity and Aggregate Data Subsystem to provide calculations specifically targeted at specific element usage within a digital lesson implementing adaptive learning features.

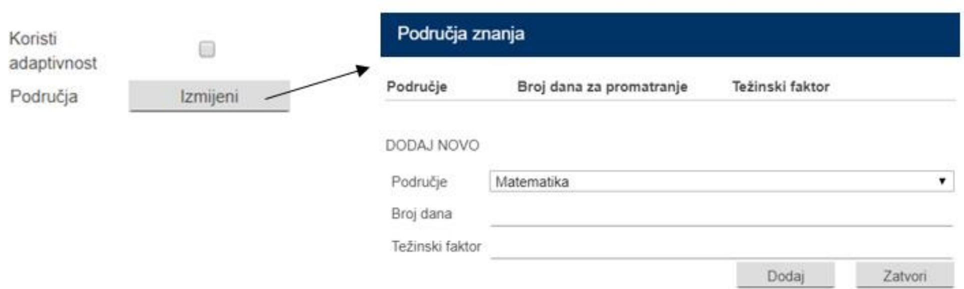


Fig. 10. Digital lesson element parameters to be utilized by the Adaptivity and Aggregate Data Subsystem to provide adaptivity functions

V. CASE STUDIES

A. *Case Study 1: The Math Widget*

As part of the study presented in this paper, a simple widget called “The Math widget” for learning mathematics in early primary school is created. It is used to simulate a typical in-class scenario of completing mathematics tasks, which students usually perform on paper or in front of the class, at the blackboard, one by one.

Upon widget initialization within a module (digital lesson), widget instance parameters including concrete formulae (please refer to the allowedValues list in the second property of the inParams element in Table 2) are send as input to the widget, which then displays these tasks to the learner. After a learner provides a response, the widget logic evaluates it and displays a feedback message to the learner. One typical series of steps in the Math widget use is shown in Fig. 11.

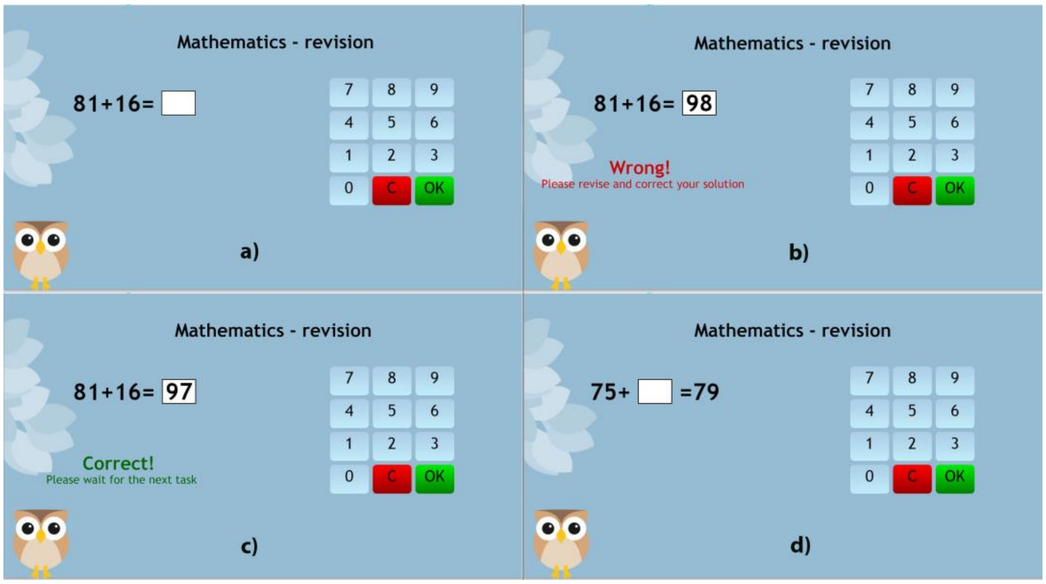


Fig. 11. Usage pattern of the Math widget: a) a task specified in the widget parameters is displayed to the student, b) student submits a solution which is then evaluated as incorrect, c) student resubmits the correct solution and d) a new task is displayed to the student

After a student attempts to solve a task, the widget logs the submitted solution using the Progress monitoring subsystem and in case of a correct solution transitions to a new task. In case of an incorrect task solution provided by the student, widget displays an error message prompting the student on another task solution attempt. Fig. 12 shows students using the Math widget for learning mathematics in classroom whereby each student is assigned with a tablet device that is used individually, and the individual results being displayed on the central canvas as part of the Web application for teachers.

Fig. 12. Left hand side of the picture shows a student completing a mathematics task using the Math widget, while the right-hand side of the picture shows a class where individual results are displayed to all students

B. Case Study 2: Adding Collaborative Features to the Math Widget

In the second study the Math widget is supplemented with collaborative features utilizing the Group work and group communication subsystem to enable synchronous learner collaboration in real time. Students still continue to use the Math widget features which are now rendered in an alternative way so that students get to work in groups to solve a single task. Group work is organized using a role-based approach where students participate in three main roles: the author, the editor and the checker roles.

Fig. 13 depicts a step in a collaborative activity implemented using the Math widget where the student in the author role processed the assignment text into a formula, the student in the editor role provided a solution to the formula, while the student in the checker role is about to evaluate the correctness of both the author's and editor's solutions.

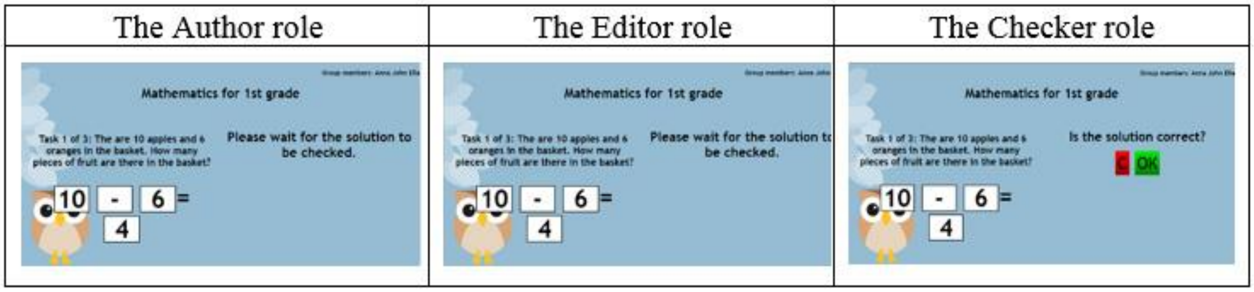


Fig. 13. Math widget used in a collaborative operation mode. Three students participate in a collaborative learning activity by completing the same mathematics task, where each student got assigned a different role and needs to complete a portion of the task

After all students provide their contributions to the joint task, the activity formulae are transferred to other students' mobile devices in real time using the Group work and group communication subsystem API methods. The discussion throughout which students come up with a joint solution by completing their own portions of the task is illustrated in Fig. 14.

Fig. 14. Students discussing their jointly proposed solution while completing a synchronous collaborative learning task

Table 5 illustrates two processes of completing a mathematics task using 1) the basic Math widget where a student attempts to complete a task in multiple calculation attempts and 2) with the use of the Math widget with collaboration features where students come up with a solution via mutual collaborative agreement on the individually proposed sub-task elements.

TABLE V

STUDENT ACTION SEQUENCE WHEN USING THE BASIC AND COLLABORATIVE MATH WIDGETS IN COMPLETING A TASK

Task: Jenny has 12 crayons. Paul has 5 crayons more that Jenny. How many crayons do they have all together?"	
Basic Math widget usage example	Collaborative Math widget usage example
Student: 12+5 * Translate the problem to mathematical equation	Author: 12+5 * Translate the problem to mathematical equation
Student: 17 * Solve the equation	Editor: 17 * Solve the equation
Student receives the feedback on the incorrect task and needs to figure out the error and attempt again	Checker: Incorrect (C) * Check the entire solution
Student: 12+5 * Translate the problem to mathematical equation	Author and editor were first convinced that they solved their subtasks correctly, but editor, after thinking about the task more thoroughly, advised author to read the problem more carefully.
Student: 17 * Solve the equation	Author: 12+5+12 * Translate the problem to mathematical equation
After repeated incorrect attempt, student considers an alternative approach in completing the task.	Editor: 29 * Solve the equation
Student: 12+5+12 * Translate the problem to mathematical equation	Checker: Correct (OK) * Check the entire solution
Student: 29 * Solve the equation	-

The widget operation modes impact the task completion, especially when students reach an impasse in completing a task. In the Basic widget usage students mostly revert to the trial-and-error strategy of solving a difficult task throughout numerus completion attempts. On the other hand, when working in groups with roles, the task distribution allows student to leverage group resources to jointly come to a correct solution.

C. Case Study 3: Adding Adaptive Features to the Math Widget

The Math widget is supplemented with adaptive features in the third case study relying on the system's Adaptivity and Aggregate Data Subsystem for calculating the initial student knowledge levels. By utilizing that adaptivity information the widget further provides specific in-lesson adaptivity to the learners (Fig. 15).

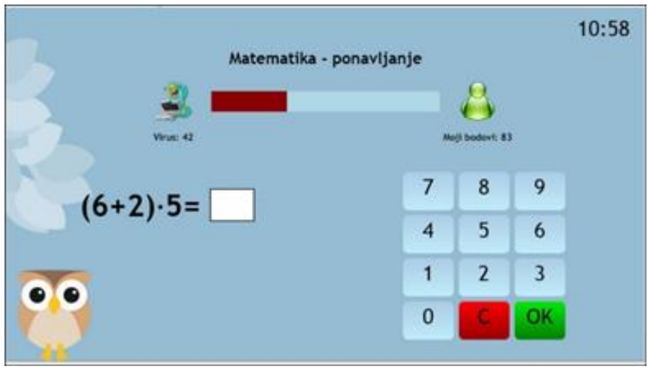


Fig. 15. The Math widget used in the adaptive operation mode. A student is presented with a mathematics task and is given certain amount of time (indicated with the progress bar element). Left over time is adapted to the collected points

This adaptive features in the Math widget are used with the final goal of encouraging motivation, interest and engagement in students and are developed as a game, in which a student competes against a computer opponent (presented to the student as a narrative of a "computer virus" that infiltrated their tablet computers). By completing as many tasks correctly and as fast as possible a student is awarded points (three points for the correct answer), while incorrect solution results in deducting one point from the overall student's number of points, and failing to answer correctly in the given time increases the overall virus points (3 points go to the virus). The goal of the game for the student is to have more points than the virus at the end of the digital learning lesson (Fig. 16).

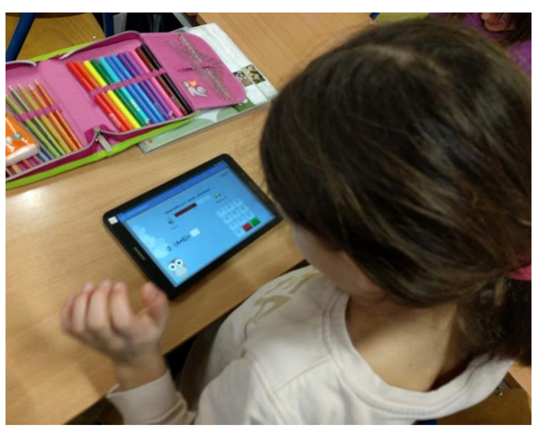


Fig. 16. Students engaging in an adaptive math learning task of competing with a "virus"

The adaptive Math widget receives average time a student spent on solving a single task in related past lessons as an input parameter from the Central server Adaptivity and Aggregate Data Subsystem to initialize progress bar value. The adaptive widget adapts to students both in real-time and in long-term using history data on student performance. Real time data collected from the students is used to adapt and maintain student motivation during the digital lesson execution.

The widget was used as part of the mathematics subject in early primary school with students aged 7 to 9 years (2nd and 3rd grade). A total of 51 students participated in the study where their detailed usage was logged with the help of the Progress monitoring subsystem and used to determine the students' performance improvement regarding basic Math widget. The total lesson time (15 minutes) for each lesson was analyzed by observing student performance change in 3-minute intervals. The performance change between the intervals was modelled via linear regression, with steepness (slope) of the regression line indicating the improvement or deterioration of performance throughout the whole activity. Table 6 shows aggregate data collected in experiments with the basic and adaptive Math widgets, while Fig. 17 illustrates the linear regression models of both widget uses.

Table 6 presents total and average number of tasks students managed to solve within each consecutive three-minute interval. On average, a student managed to solve around 18 math problems in the first 3 minutes of the lesson. In the Basic math widget this average decreases to around 15, while in the Adaptive Math lesson, the average number of solved tasks increased to around 21 tasks per student. When using the Adaptive widget students increased their performance during the lesson, while with the Basic lesson widget their performance decreased as the lesson progressed.

TABLE VI

STUDENT PERFORMANCE IN BASIC AND ADAPTIVE MATH WIDGET USAGE. 15-MINUTE DIGITAL LESSONS ARE SPLIT INTO INTERVALS OF 3 MINUTES TO ALLOW FOR PERFORMANCE CHANGE OBSERVATION

Experiment using Widget / Time interval	0 - 3 min	3 - 6 min	6 - 9 min	9 - 12 min	12 - 15 min
SUM					
Basic Math widget	937	870	840	866	792
Adaptive Math widget	910	1009	1014	1037	1103
AVERAGE					
Basic Math widget	18	17.1	16.5	17	15.5
Adaptive Math widget	18	19.8	19.9	20.3	21.6

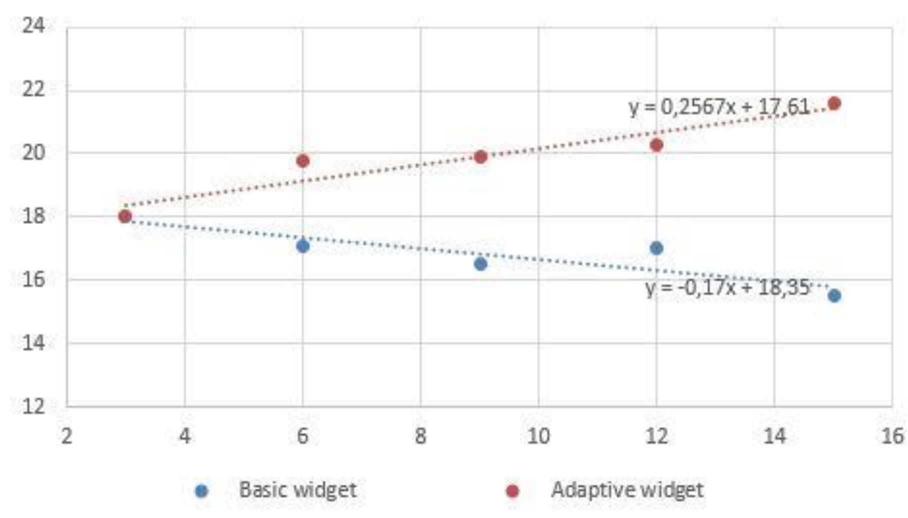


Fig. 17. Linear regression models for the two Math operations: the basic and the adaptive Math widgets

The comparison of student performance using the basic and the adaptive Math widgets indicates student performance in the adaptive learning scenario brought better performance levels, whereby students completed more tasks in general and had more correctly solved tasks.

VI. Discussion

This paper proposes a model and architecture of a mobile learning system which uses collaboration and adaptivity as part of modular digital learning lessons. The overarching objective of this study was the design and implementation of a complete mobile learning solution that supports new pedagogies and allows for the reuse and enrichment of the existing learning designs. The proposed system uses modular organization of learning materials into reusable blocks, thereby allowing for easy composition and linkage of related learning elements.

The proposed architecture was implemented and trialed in a primary school environment through the three case studies designed to examine the key functions of the proposed system in more detail. These case studies covered the main aspects of the system: its modularity and the potential for reuse and combining exiting learning materials, the support for real time synchronous collaboration between students across different platforms, and the adaptivity using adaptive learning algorithms.

The first case study examined the use of a simple widget which replaces the standard pen-and-paper in-class practice of mathematics. Compared to the pen-and-paper approach, the use of the system during 45-minute Mathematics practice sessions resulted in students solving 50 to 55 mathematics tasks. The use of the mobile learning system made solving mathematics tasks simpler since students used tablet computers at their desks and could complete as many tasks as they wanted to, without spending time on mundane tasks such as copying solutions into their notebooks. With the use of the system, the number of solved tasks per student increased and students immediately received feedback (on whether their solution was correct or wrong) which was usually not the case with the traditional pen-and-paper sessions. Nevertheless, similarly to traditional sessions, students' interest would gradually deteriorate as time progressed (Fig. 17).

The second case study showed that in the context of collaborative learning tasks for solving lower primary mathematics problems (Alvarez et al., 2011) , the use of roles (Marcos-García, Martínez-Monés, & Dimitriadis, 2015) relates to the task difficulty, whereby roles are more suitable difficult tasks. For less demanding tasks, where students work individually and discuss their solutions when misconceptions occur, roles may not be needed. In terms of technology design more coordination and synchronization mechanisms were needed to support the collaborative activity flow.

The third case study used the built-in system adaptivity for providing learning experiences tailored to abilities of each student. The combination of adaptivity and gamification of the lessons resulted in increased motivation, interest and engagement of most students, who retained or increased their performance during the lesson (Fig. 17), which is was indicated by similar studies in the field (Garcia-Cabot, De-Marcos, & Garcia-Lopez, 2015; Soflano, Connolly, & Hainey, 2015). On the other hand, there were students who did not like the competitive style, felt pressure and quit before the end of the activity. Results of our research indicate the future versions of similar adaptive modules should detect such situations and further adapt to the students and keep in order to keep them in their comfort zone.

VII. Conclusion

In this paper a system architecture for supporting new pedagogies is presented and discussed in three mutually interconnected studies. The presented architecture is built to allow for the organization of existing digital interactive web-based learning contents via the concept of modules, sub-modules and basic elements, amongst other including widgets. Widgets wrap the existing interactive digital educational solutions already created by the teachers, content designers and other parties and provide interfaces that allow for their systematic use and pedagogical enrichment as part of the proposed system.

The system architecture contains architecture elements implementing the support for adaptive and collaborative learning pedagogies that can be added onto the modules, sub-modules and widgets following their import into the system. By setting up parameters for adaptive and collaborative learning, any standard web-based programming component packed into a widget can utilize the system adaptivity logic or build its own logic based on the data provided by the central system adaptivity infrastructure. Widget designers choose amongst the available adaptivity interface methods and utilize the adaptivity information provided by the system to enrich their educational digital contents and logic. Similarly, the architecture elements for collaborative learning add the features of automatized grouping of students that can be generated and used in real time upon the collaborative activity onset. In such way, multiple collaborative activities per a single digital lesson are possible, each with own contents, parameters, participants and groupings.

The experiences gained in the study are presented throughout the three case studies where a digital lesson for learning basic mathematics operation for early primary school children presents the base widget that is packed into a module and then further enriched with the adaptive and collaborative features. The case study utilizing adaptivity features depicts the use of adaptation increasing student performance while solving mathematics tasks, compared to the use of the basic widget with no adaptation support. In the case study using collaborative learning system infrastructure and APIs the student problem solving processes are changed, as opposed to the individual task completion process.

Apart from this study, which focuses on the design and implementation of the mobile learning system, the same system was used in a series of experiments that explored the effects of game elements and gamification on student motivation and class performance (Jagušt, Botički, & So, 2018). In order to achieve that, experiments were designed, and data was collected, analyzed and interpreted. The study used specially designed widgets and software extensions implementing game scenarios, which were used as new modules in the presented mobile learning system. The study showed, in line with conclusions presented in this paper, that the use of novel pedagogies, collaboration, adaptivity and game elements improves student engagement, keeps them interested and focused for a longer period of time, resulting in improved student performance, potentially leading to improved student knowledge and test results.

Future work on the system will include development of a new case study utilizing both the adaptive and collaborative learning architecture features in the same time to allow for the exploration of joint effects of adaptive collaborative learning. Additionally, methods for comparison of student performance when working in groups in adaptive and non-adaptive modes will be devised and examined.

References

Alvarez, C., Alarcon, R., & Nussbaum, M. (2011). Implementing collaborative learning activities in the classroom supported by one-to-one mobile computing: A design-based process. *Journal of Systems and Software*, *84*(11), 1961–1976. https://doi.org/10.1016/j.jss.2011.07.011

Bower, M., Dalgarno, B., Kennedy, G. E., Lee, M. J. W., & Kenney, J. (2015). Design and implementation factors in blended synchronous learning environments: Outcomes from a cross-case analysis. *Computers and Education*, *86*, 1–17. https://doi.org/10.1016/j.compedu.2015.03.006

Brusilovsky, P., & Peylo, C. (2007). Adaptive and Intelligent Web-based Educational Systems.

Chiang, T. H. C., Yang, S. J. H., & Hwang, G. (2014). An Augmented Reality-based Mobile Learning System to Improve Students' Learning Achievements and Motivations in Natural Science Inquiry Activities.pdf, *17*(April), 352–365. https://doi.org/10.1109/ISMAR.2006.297800

El Guabassi, I., Bousalem, Z., Al Achhab, M., JELLOULI, I., & EL Mohajir, B. E. (2018). Personalized adaptive content system for context-aware ubiquitous learning. *Procedia Computer Science*, *127*(8), 444–453. https://doi.org/10.1016/j.procs.2018.01.142

Fita, A., Monserrat, J. F., Moltó, G., Mestre, E. M., & Rodriguez-Burruezo, A. (2016). Use of synchronous e-learning at university degrees. *Computer Applications in Engineering Education*, *24*(6), 982–993. https://doi.org/10.1002/cae.21773

Garcia-Cabot, A., De-Marcos, L., & Garcia-Lopez, E. (2015). An empirical study on m-learning adaptation: Learning performance and learning contexts. *Computers & Education*, *82*, 450–459. https://doi.org/10.1016/j.compedu.2014.12.007

Holenko Dlab, M., Boticki, I., Hoic-Bozic, N., & Looi, C. (2017). Adaptivity in Synchronous Mobile Collaborative Learning. In L. Gómez Chova & I. López Martínez, A. Candel Torres (Eds.), *Proceedings of the International Conference on Education and New Learning Technologies - EDULEARN17* (pp.

454–460). Barcelona: IATED Academy.

Holenko Dlab, M., Hoic-Bozic, N., & Boticki, I. (2017). A Design-Based Approach to Developing a Mobile Learning System. *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, *11*(10), 2320–2325.

Hsiao, I. H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: Adaptive navigation support in an E-Learning system for Java programming. *Journal of Computer Assisted Learning*, *26*(4), 270–283. https://doi.org/10.1111/j.1365-2729.2010.00365.x

Jagušt, T. (2015). Mobile learning platforms : A literature review.

Jagušt, T., Botički, I., & So, H.-J. (2018). Examining competitive, collaborative and adaptive gamification in young learners' math learning. *Computers & Education*, *125*(June), 444–457. https://doi.org/10.1016/j.compedu.2018.06.022

Järvelä, S., Kirschner, P. A., Panadero, E., Malmberg, J., Phielix, C., Jaspers, J., … Järvenoja, H. (2014). Enhancing socially shared regulation in collaborative learning groups: designing for CSCL regulation tools. *Educational Technology Research and Development*, *63*(1), 125–142. https://doi.org/10.1007/s11423-014-9358-1

Kirschner, P. A. (2017). Stop propagating the learning styles myth. *Computers and Education*, *106*, 166–171. https://doi.org/10.1016/j.compedu.2016.12.006

Looi, C., Lim, K. F., Pang, J., Lay, A., Koh, H., Seow, P., … Soloway, E. (2016). Bridging formal and informal learning with the use of mobile technology. In *Future learning in primary schools* (pp. 79–96). https://doi.org/10.1007/978-981-287-579-2

Luo, L., Yang, Y., & Wei, Y. (2017). Research on Adaptive Mobile Collaborative Learning System. *ACM Computing Surveys*, *50*(3), 413–423.

Madhubala, R., & Akila, A. (2017). Context Aware and Adaptive Mobile Learning : A Survey. *Advances in Computational Sciences and Technology*, *10*(5), 1355–1370.

Marcos-García, J. A., Martínez-Monés, A., & Dimitriadis, Y. (2015). DESPRO: A method based on roles to provide collaboration analysis support adapted to the participants in CSCL situations. *Computers and Education*, *82*, 335–353. https://doi.org/10.1016/j.compedu.2014.10.027

Martin, S., Diaz, G., Plaza, I., Ruiz, E., Castro, M., & Peire, J. (2011). State of the art of frameworks and middleware for facilitating mobile and ubiquitous learning development. *Journal of Systems and Software*, *84*(11), 1883–1891. https://doi.org/10.1016/j.jss.2011.06.042

Mutahi, J., Bent, O., Kinai, A., Weldemariam, K., & Sengupta, B. (2015). Capturing Learner's Activity Events from a Mobile Learning System Using Adaptive Event Framework. *Proceedings - 2nd ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2015*, 109–112. https://doi.org/10.1109/MobileSoft.2015.23

O'Malley, C., Vavoula, G., Glew, J. P., Taylor, J., Sharples, M., Lefrere, P., … Waycott, J. (2005). Guidelines for learning / teaching / tutoring in a mobile environment. *Public Deliverable from the MOBILearn Project (D.4.1)*.

Perveen, A. (2016). Synchronous and Asynchronous E-Language Learning : A Case Study of Virtual University of Pakistan. *Open Praxis*, *8*(1), 21–39. https://doi.org/10.5944/openpraxis.8.1.212

Rahul Raj, C. P., & Tolety, S. B. (2012). A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach. *2012 Annual IEEE India Conference, INDICON 2012*, 625–629. https://doi.org/10.1109/INDCON.2012.6420693

Sharples, M., Adams, A., Alozie, N., Ferguson, R., Fitzgerald, E., Gaved, M., … Yarnall, L. (2015). *Innovating Pedagogy makers*.

Shute, V. J., & D, Z.-R. (2012). Adaptive educational systems. *Adaptive Technologies for Training and Education*, *7*(27), 7–27.

Soflano, M., Connolly, T. M., & Hainey, T. (2015). An application of adaptive games-based learning based on learning style to teach SQL. *Computers and Education*, *86*, 192–211. https://doi.org/10.1016/j.compedu.2015.03.015

Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences* (pp. 409–426). Cambridge University Press. https://doi.org/10.1145/1124772.1124855

Strijbos, J.-W., & Weinberger, A. (2010). Emerging and scripted roles in computer-supported collaborative learning. *Computers in Human Behavior*, *26*(4), 491–494. https://doi.org/10.1016/J.CHB.2009.08.006

Su, C. H., & Cheng, C. H. (2015). A mobile gamification learning system for improving the learning motivation and achievements. *Journal of Computer Assisted Learning*, *31*(3), 268–286. https://doi.org/10.1111/jcal.12088

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.

Syvänen, A., Beale, R., Sharples, M., Ahonen, M., & Lonsdale, P. (2005). Supporting pervasive learning environments: Adaptability and context awareness in mobile learning. *Proceedings - IEEE International Workshop on Wireless and Mobile Technologies in Education, WMTE 2005*, *2005*, 251–253. https://doi.org/10.1109/WMTE.2005.62

Veerman, A. L., & Veldhuis-Diermanse, A. E. (2001). Collaborative learning through computer-mediated communication in academic education. In *Proceedings European Perspectives on Computer Supported Collaborative Learning : Euro-CSCL* (pp. 625–632). Maastricht : Maastricht McLuhan Institute.

Wald, M., Li, Y., & Draffan, E. a. E. (2014). Synote: Collaborative Mobile Learning for All. *Procedia Computer Science*, *27*(Dsai 2013), 240–250. https://doi.org/10.1016/j.procs.2014.02.027

Wen, C., & Zhang, J. (2015). Design of a Microlecture Mobile Learning System Based on Smartphone and Web Platforms. *IEEE Transactions on Education*, *58*(3), 203–207. https://doi.org/10.1109/TE.2014.2363627

Wong, L. H., & Looi, C. K. (2011). What seams do we remove in mobile-assisted seamless learning? A critical review of the literature. *Computers and Education*, *57*(4), 2364–2381. https://doi.org/10.1016/j.compedu.2011.06.007

Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications. *Proceedings of the 6th Balkan Conference in Informatics on - BCI '13*, 213. https://doi.org/10.1145/2490257.2490292

Zhao, X., Anma, F., Ninomiya, T., & Okamoto, T. (2008). Personalized Adaptive Content System for Context-Aware Mobile Learning. *IJCSNS International Journal of Computer Science and Network Security*, *8*, 153–161.

Zhao, X., & Okamoto, T. (2008). A Device-Independent System Architecture for Adaptive Mobile Learning. *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, 23–25. https://doi.org/10.1109/ICALT.2008.21

Zurita, G., & Nussbaum, M. (2007). A conceptual framework based on Activity Theory for mobile CSCL. *British Journal of Educational Technology*, *38*(2), 211–235. https://doi.org/10.1111/j.1467-8535.2006.00580.x