

# On Data Sharing in Educational Processes

<http://dx.doi.org/10.3991/ijet.v11i04.5298>

Yousef Ibrahim Daradkeh<sup>1</sup>, Mujahed ALdhaifallah<sup>2</sup> and Dmitry Namiot<sup>3</sup>

<sup>1,2</sup>Prince Sattam bin Abdulaziz University, Wadi Aldawaser, KSA

<sup>3</sup>Lomonosov Moscow State University, Moscow, Russia

**Abstract**—In this paper, we present one model for data sharing in educational classes. Typically, Learning Management Systems present data stores for keeping educational materials as well as the conversations between teachers and students. In our model, we propose a peer to peer data exchange via smartphones. With the high penetration of smartphones across students, the ability to support one-to-one communication with teachers could be a good add-on for the traditional learning support systems. This ability could be especially useful for on-demand organized classes, where the standard support is very costly.

**Index Terms**—file sharing, second screen, peer to peer model, cloud services.

## I. INTRODUCTION

Learning Management Systems (LMS) are web-based systems that allow instructors and/or students to share materials, submit and return assignments, and communicate online [1]. So, a typical LMS is a software application for the administration, documentation, tracking of a learning process. LMS support reporting and delivery of electronic educational technology (e-learning) courses and training programs.

Some of the authors also describe learning content management system (LCMS). LCMS is a related software technology that provides a multi-user environment where developers, authors, instructional designers, and subject matter experts may create, store, reuse, manage, and deliver digital educational technology content from a central object repository. In order to highlight the difference, we could say that LCMS focuses on the development, management and publishing of the content that will be later delivered via an LMS [2].

Learning Management Systems range from systems for managing training and educational records to software for distributing online or blended/hybrid college courses over the Internet with features for online collaboration. Colleges, universities, school districts, and schools use LMSs to deliver online courses and augment on-campus courses. Corporate training departments can use LMS applications to deliver online training, as well as to automate record-keeping and employee registration.

Of course, all LMS (LCMS) has got the ability to present some data from teachers (instructors) to students. Actually, it is one of their main functions – let instructors share data with students. In the most obvious use case, it is some data prepared by a teacher (teachers) for the distribution and placed in advance on the appropriate channels within LMS.

In this paper, we would like to discuss the possibility to share some data on demand. We will present a way (one approach) for sharing data between instructor and student

(one-to-one) or instructor and students (one-to-many) on demand. For example, some data could be prepared right on the instructor's mobile phone (with the help of his phone) and transferred (demonstrated to the students). For example, draw a chart on the screen and share it, draw a chart on the paper, make a photo with a camera and share an image, and so on.

In this paper, we will discuss on-demand data sharing. Of course, there are many use cases for such applications, but at the first hand, we will be interested in education.

## II. ON DEMAND INFORMATION SHARING

The fundamental role of the network (The Internet in our particular case) is to provide an environment for the data transfer between users. This data can take a variety of forms such as text or binary files, structured information, etc. Of course, numerous methods of data transfer have existed since the time the Internet became available for public use (such as email and FTP), many of these methods require users to first register an account with an application provider. Only this account lets the user send and receive messages [3]. More recently, we can mention downloadable applications which have been created for instant communication between mobile clients. But it is again required for these clients to have the same application preloaded on their devices in order to communicate with other. Of course, the registration requirements make it difficult for use to communicate in an instantaneous, or on-demand, manner.

On-demand communication is especially important in professional fields such as emergency medicine, petroleum engineering, etc., where information must be transferred between specialists working at different locations [3].

One of the first things to come to mind, in this case, is the use of social networks. In fact, they are just intended for the exchange of information. It is possible to organize the exchange of information out there on the one-to-one or one-to-many model. The teacher (instructor) can post own messages (this, of course, is done directly from a mobile phone), and his subscribers (followers - whatever is a name of a social network) will read this information. It is possible also to organize of interaction on the one-to-one model (private messages in Facebook, direct messages in Twitter, etc.). But even in the social networks, there is a rising trend to create on-demand professional groups [4]. The current approach to custom group creation is to pre-categorize groups of users (e.g., friends, follower, etc.) in advance of sharing information. We can mention in this context several papers [5-7]. Groups could be created automatically (e.g., Smart Lists in Facebook), based on a person's social graph or other common features (e.g., inferred closeness, current location, etc). Of course, these

automatically generated groups can then be manually edited for correctness or to capture other preferences [4]. As a fully on-demand information system, we can mention our paper [8]. It describes a request-response system on the top of the Twitter social network.

In terms of the education, think about a lecture on the conference, a consultancy meeting, etc. there are no pre-defined groups of students and very often there are no pre-defined support sites. It would be good to allow the teacher (instructor) to directly get in contact with the listeners (visitors). And vice-versa, it would be good to allow the students (listeners, participants) to contact the teacher without any registration on external portals.

Right now, many educational organizations use (create) special accounts in social networks. In the same time, with regard to the training, we could face an unexpected problem of restricted access. In fact, in Russia, for example, access to social networks in schools could be limited. It has just one simple reason behind: to prevent students from spending all their time in the social networks. So, the use of social networks directly in the classroom (in any form) may be prohibited. Actually, the same is true for instant messages.

In order to deal with the preinstalled applications problem, we should pay attention the mobile browsers. The mobile browser is the application already preinstalled on any smartphone.

There are different ways to organize data exchange with the web (web browser) only. For example, Web Real-Time Communication (WebRTC) is being developed by various organizations with the idea to open data (information) channels right in browsers [9]. WebRTC embeds Peer-to-Peer technology into web browsers, allowing them to create direct links over the internet with the computers of other users, without the need for installing additional software or service applications [10].

Another interesting area is formed by mesh networks. Mesh networking is a type of network topology in which any particular node (e.g., mobile device) transmits its own data as well as serves as a relay for other network nodes (mobile devices). There are two forms of mesh networks, which could be interesting for educational applications. A wireless mesh network is an infrastructure-based network organized in a multi-hop mesh topology of static routers with single-hop connected mobile clients. In other words, there is no direct connectivity between clients. And the second (and probably most interesting) model is presented via Social Mesh Networks. Social Mesh Networks use existing user equipment (mobile terminals) via multi-hop connections outside the cellular infrastructure. In Social Mesh Network, a wireless device would establish a connection with other wireless devices nearby. The communication would be based on a direct link between the devices. Each device might be linked to one or many other devices. In other words, we will talk about the direct connections from a teacher's (an instructor's) device to the students devices (or vice versa). In this model, each device would be open to sending, receiving and relaying messages as a part of the mesh network. The other forms of mobile mesh networks are presented in our paper [11]. The main drawback for this approach is the diversity of standards and incompatibility in existing implementations. In this paper, we will present a client-server based data transfer.

### III. ON DATA EXCHANGE

The proposed model (WebPair [12]) lets virtually connect two (or more) smartphones, so data prepared by one user will be available for all connected parties. The word "virtual" here means that all parties will transparently use the same cloud-based data sharing area.

More precisely, this service virtually links several mobile browsers. It let us link mobile browser to mobile browser, mobile to a desktop browser, desktop to desktop, mobile browser to Smart TV, etc. The key idea is that internet connection should be enough for passing data from one browser to another (from any screen to another screen). There is no need to download and install any additional application. Everything should work right in the browser. For example, internet connected mobile phone should be enough for transferring data to Smart TV [13]. So, in some educational class, an instructor will be able to share data from own mobile on the big screen of Smart TV, for example.

### IV. ON WEBPAIR IMPLEMENTATION

The main idea for linking (pairing) web browsers has been borrowed from pairing devices in Bluetooth networks [14]. How does it work? Both parties (two mobile users, for example) should point own browsers to the same URL: <http://pair.linkstore.ru> (Figure 1).

Of course, this URL could be different for the service on another domain. The opened mobile web page contains some unique 4-digits code. The picture will be the same in the second (third, etc.) browser. Of course, each party will see an own code. This random code will be used for pairing two devices. The schema for pairing is similar to Bluetooth pairing. There are two parties: master and slave. A master browser transfers the data, a slave browser accepts data.



Figure 1. An initial screen for WebPair application.

On the screen (phone) that should display the data (be linked to another phone/show data in slave mode) we have to type the code presented on the second (master) browser. We do not need to type anything on the second (master) phone. It will recognize the connection from the slave device automatically.

The system supports one-to-one connections (master - slave) as well as many-to-one models (one master and many slave devices). In both cases, an instructor should tell his code to participants. His browser will be master, the students will use slave browsers.

As soon as the mobile user submits the code for his master, the slave screen is ready to accept data. And as soon as the code is entered, the second screen (master) will be updated automatically (Figure 2).

A master (teacher, instructor) can type any text and press Push button on the master screen. His pals will see data on the slave screen. It will be provided automatically, without refreshing the screen.

It is a basic example that transfers text data. Technically it is possible to pass any data available in the browser. For example, it is possible to:

- just type any text. All connected slaves phones will see that information;
- type any phone number. It will be presented as a clickable link on the slave screen (screens)
- type any URL. E.g., <http://servletsuite.com> Slave screen will show a clickable link
- type any email address. Slave screen will show a clickable link for *mailto*;
- type the following command `t:some_name_from Twitter`. E.g., `t:t411`. Slave screen (screens) will show the link to that account in Twitter (<http://twitter.com/t411> in our example). So, your pals will be able to read your feed, follow it, etc.
- type the following command `f:some_name_from Facebook`. It is the same functionality as above, but for Facebook. Slave screen (screens) will show the link to that profile in Facebook. So, your pals will be able to read your feed, subscribe, etc.

In the latest version, a teacher can transfer image files (or pictures from the camera) too. E.g., get the file (camera picture) from mobile and show it on Smart TV, etc. A master screen contains a widget for images selection. Actually, in our educational use case, it was a favorite scenario. A teacher prepared some chart on the paper (actually, it is still a much faster that use any online editor), make a camera image and transfer (demonstrate) that image to students. Another popular scenario is show images right on the Smart TV in a class (in other words, show an image from a mobile phone on the big screen). As a next actively used scenario, we can mention paper works checking by the teacher. A teacher checks student's paper work, writes notes, makes a photo, and shows it in the one-to-one mode to a student.

A non-educational use case contains on-demand data sharing (actually, image sharing) for paramedics [3]. In this case, a paramedic used live images for consulting with colleges. Of course, it was an anonymous data exchange, but as an additional security mechanism, such systems can use so-called self-destructed messages [15].

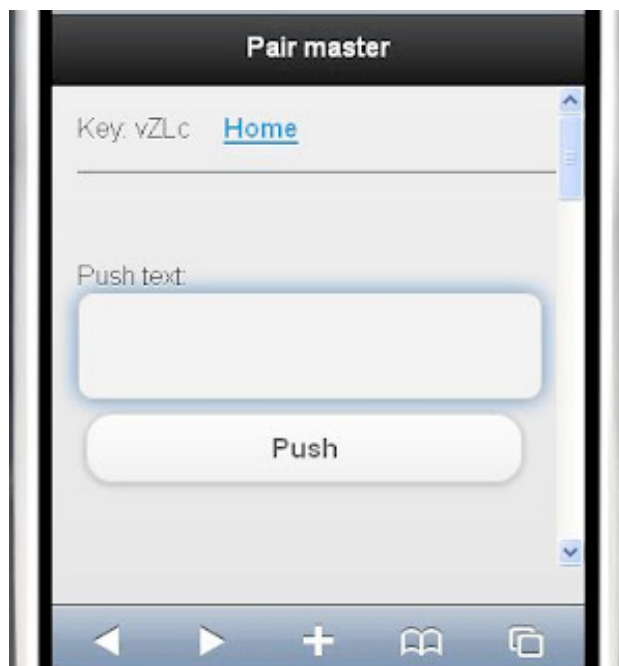


Figure 2. A master screen of WebPair application

The file transfer is based on HTML5 File Read API [16]. It has been selected due to compatibility reasons. An alternative solution is *WebSocket JavaScript* interface, which defines a full-duplex single socket connection over which messages can be sent between client and server. The *WebSocket* standard simplifies much of the complexity around bi-directional web communication and connection management [17]. The problems are the compatibility again and the lack of two-way communication requirements for this particular use case. In general, we think that the future of such kind data transfer is Web RTC [18]. As well as audio and video, WebRTC supports real-time communication for other types of data.

The *RTCDataChannel API* enables a peer-to-peer exchange of arbitrary data, with low latency and high throughput. The biggest problem at this moment is the compatibility and support in mobile browsers. In WebRTC, communication occurs directly between browsers, so *RTCDataChannel* can be much faster than *WebSocket*

## V. CONCLUSION

In this paper, we present a web-based on-demand data exchange application. It lets mobile users share own data with one-to-one and one-to-many models. In educational processes a teacher (an instructor) can share data with students. The whole process does not require any authorization or authentication and it lets share data on-demand, without the registration in social networks or downloading and installing additional applications. In the tested scenarios, a teacher shared image files with the students in the one-to-one model or demonstrated them on the Smart TV in class (one-to-many model). The non-educational use case tested on-demand image sharing for paramedics. The proposed approach (its implementation) showed the good compatibility level (it worked on all tested mobile browsers) as well as an acceptable performance for mobile internet connectivity (both 3G and GPRS).

ACKNOWLEDGMENT

We would like to thank the people from Open Information Technology Lab in Lomonosov Moscow State University for their help in testing our applications.

REFERENCES

- [1] Lonn, S., & Teasley, S. D. (2009). Saving time or innovating practice: Investigating perceptions and uses of Learning Management Systems. *Computers & Education*, 53(3), 686-694. <http://dx.doi.org/10.1016/j.compedu.2009.04.008>
- [2] Chen, Kai, and D. X. Teng. "The Exploration of Network Platform Based on Open Source LCMS." *XuZhou Institute of Technology 4* (2007): 18-20.
- [3] Stephenson, A., & Namiot, D. (2015). On data transfer between mobile web clients. *International Journal of Open Information Technologies*, 3(1), 30-40.
- [4] Amershi, S., Fogarty, J., & Weld, D. (2012, May). Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 21-30). ACM. <http://dx.doi.org/10.1145/2207676.2207680>
- [5] Kelley, P. G., Brewer, R., Mayer, Y., Cranor, L. F., & Sadeh, N. (2011). An investigation into facebook friend grouping. In *Human-Computer Interaction-INTERACT 2011* (pp. 216-233). Springer Berlin Heidelberg. [http://dx.doi.org/10.1007/978-3-642-23765-2\\_15](http://dx.doi.org/10.1007/978-3-642-23765-2_15)
- [6] Pang, J., & Zhang, Y. (2015). A new access control scheme for Facebook-style social networks. *Computers & Security*. <http://dx.doi.org/10.1016/j.cose.2015.04.013>
- [7] Bartel, J. W., & Dewan, P. (2013, October). Evolving friend lists in social networks. In *Proceedings of the 7th ACM conference on Recommender systems* (pp. 435-438). ACM. <http://dx.doi.org/10.1145/2507157.2507194>
- [8] Namiot, D. (2015). Twitter as a Transport Layer Platform. *arXiv preprint arXiv:1509.06874*.
- [9] Johnston, A. B., & Burnett, D. C. (2012). WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web. *Digital Codex LLC*.
- [10] Azevedo, J., Pereira, R. L., & Chainho, P. (2015, April). An API proposal for integrating sensor data into web apps and WebRTC. In *Proceedings of the 1st Workshop on All-Web Real-Time Systems* (p. 8). ACM. <http://dx.doi.org/10.1145/2749215.2749221>
- [11] Namiot, D. (2015). On Mobile Mesh Networks. *International Journal of Open Information Technologies*, 3(4), 38-41.
- [12] Web Pair <http://pair.linkstore.ru> Retrieved: Nov, 2015
- [13] Sutherland, I., Xynos, K., Read, H., Jones, A., & Drange, T. (2014). A forensic overview of the LG Smart TV.
- [14] Roh, J. H., & Jin, S. (2014, February). Device control protocol using mobile phone. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on* (pp. 355-359). IEEE. <http://dx.doi.org/10.1109/icact.2014.6778981>
- [15] Tung, T. Y., Lin, L., & Lee, D. T. (2012, March). Pandora Messaging: An Enhanced Self-Message-Destructing Secure Instant Messaging Architecture for Mobile Devices. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on* (pp. 720-725). IEEE.
- [16] Robbins J. N. *HTML5 Pocket Reference*. – " O'Reilly Media, Inc.", 2013.
- [17] Pimentel, V., & Nickerson, B. G. (2012). Communicating and displaying real-time data with WebSocket. *Internet Computing, IEEE*, 16(4), 45-53. <http://dx.doi.org/10.1109/MIC.2012.64>
- [18] Nurminen, J. K., Meyn, A. J., Jalonon, E., Raivio, Y., & Garcia Marrero, R. (2013, April). P2P media streaming with HTML5 and WebRTC. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on* (pp. 63-64). IEEE.

AUTHORS

**Yousef Daradkeh** is with the Prince Sattam bin Abdulaziz University, College of Engineering at Wadi Aldawaser, 18611, KSA (e-mail: daradkeh@yahoo.ca).

**Mujahed ALdhaifallah** is with the Prince Sattam bin Abdulaziz University, College of Engineering at Wadi Aldawaser, 18611, KSA (m.aldhaifallah@psau.edu.sa:).

**Dmitry Namiot** is now with the Lomonosov Moscow State University, Moscow, Russia (dnamiot@gmail.com).

This paper was supported by the deanship of scientific research at Prince Sattam bin Abdulaziz University under the research project # 2014/1/863 Submitted, 20 Nov 2015. Published as resubmitted by the authors on 21 Dec 2015.