Collaborative Drawing Annotations on Web Videos

István Koren(✉), Petru Nicolaescu, and Ralf Klamma

Advanced Community Information Systems (ACIS) Group, RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
{koren,nicolaescu,klamma}@dbis.rwth-aachen.de
http://dbis.rwth-aachen.de

Abstract. Collaborative Drawing over a computer network, in particular on videos, usually requires some complex client-server architecture. In this paper, we want to demo an approach with the following distinctive features. On the client side we enable peer-to-peer collaborative video drawing in recent Web browsers supporting WebRTC. Developers can therefore embed it in arbitrary Web pages and users do not need to install any additional software. For persistence we use a microservice driven cloud approach which can be set up easily. All components are open source to facilitate wide use and further development. A use case evaluation showed promising results and will be presented in the demo.

1 Introduction

With the increasing mobility needed at the workplace, technologies for efficient collaboration come to the fore of various businesses. This shift is supported by mobile devices such as smartphones or tablets that make it possible to quickly turn to co-workers for ad-hoc help-seeking in remote locations or collecting useful bits and pieces of information for longer reflection. In many cases, capturing such information is realized using multimedia, such as images or videos. Yet, intuitive ways of collaboration for instructing colleagues, like highlighting certain parts on an architectural plan or marking hazardous spots in a video are not adequately attainable on today's information systems, even though technologies such as stylus pens for capacitive touchscreen devices like tablets exist. In particular, we are motivated by informal learning scenarios at construction sites, where new technical equipment and building materials are available frequently and companies struggle to cope with the growing speed of innovation while having decreasing resources for training.

To this end, we propose a Web-based toolkit for collaborative annotation of video data that is both reusable and extendable on the frontend and backend side. The demo consists of a video player that is overlaid by a drawing canvas onto which vector graphics can be collaboratively drawn on the frame level of the video. A master client, automatically designated by a calculated device profile, makes the annotations persistent using a backend service.

P. Cimiano et al. (Eds.): ICWE 2015, LNCS 9114, pp. 671–674, 2015.
DOI: 10.1007/978-3-319-19890-3_54

Our demo[1] is a seminal advancement of the DireWolf 2.0 framework [1] by using the underlying peer-to-peer communication technology with a standalone real-time collaboration framework, refining the user interface widgets as reusable components and adding a microservice-based backend for a scalable distribution.

We believe that other researchers may leverage our permissive open source work to further strengthen the Web as a device-independent platform for collaborative distributed applications.

2 Collaborative Video Drawing

We developed the video drawing prototype (cf. Figure 1) after evaluating our earlier SeViAnno system for semantic video annotation with construction companies [2]. Important requirements were the annotation of videos by drawing straight and free-hand lines, squares, circles and textual content. To support live inquiries, the drawings need to be synchronized in near real-time. As such, further requirements are consistency and automated conflict handling. To keep the mobile bandwidth low, we opted for vector-based instead of pixel graphics. Finally, we required a modular approach both on the front- and backend to be able to use the functionalities in other software projects as well.

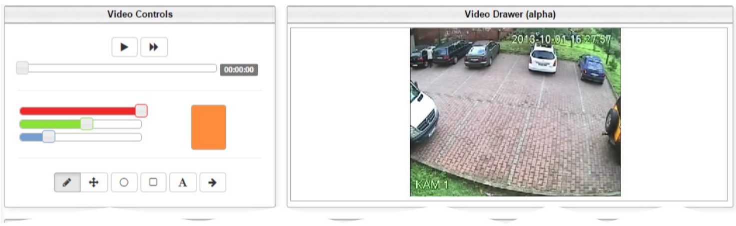

Fig. 1. Screenshot of the Video Drawing Tool Used in the Evaluation

The prototype allows the video selection from a list of videos uploaded by or shared with the user. The video may be started and stopped with a video controls toolbar that includes a player-like timeline. Existing drawings on certain frames are marked and clickable on the timeline for awareness reasons.

The specific drawing annotation types (free-hand, lines, squares, circles) can be selected in another toolbar. A video player displays the video and is overlaid by the drawing canvas. We chose to separate the controls from the video player to allow drawing over the borders of the video if desired so. Additionally, we leverage the widget distribution technologies introduced by DireWolf, i.e. the video canvas may be moved to a different device under the user's control, most usefully a touchscreen tablet.

[1] A demo video is available at http://goo.gl/iLy3gM

While a drawing is completed, it gets synchronized to other users looking at the specific video. I.e. users are seeing a live version of the current drawing. To ensure the persistence of the drawing, it is saved on a backend server.

Technically, the video widgets are implemented as Web Components [3], a recent Web standard that aims to provide reusable user interface components on the Web. Currently, Web Components are available as JavaScript library with so-called poly-fills, whereof there are some projects available. Although we currently use Polymer[2], the library should be easily exchangeable in the future, especially when the standard gets implemented natively in browsers.

The drawings are distributed as vector graphics to avoid having to send bandwidth-intensive pixel graphics on every change. Though remarkable work has been done in this respect, i.e. targeting the DOM layer as synchronized entity [4], we found available SVG based libraries to be too heavy for resource-constrained mobile devices. Therefore we employ the lightweight FabricJS[3] library that works with the JSON data, though it can also output SVG data.

The JSON data structure of the drawings gets automatically distributed to other users of the application watching the same video through our Yjs framework for shared editing of arbitrary data types [5]. Using an optimistic concurrency control algorithm, the framework is composed of a collaboration engine that maps certain data types, and connectors for message propagation. The video drawing prototype currently uses a JSON data type and the WebRTC connector of Yjs. For storing the shared state of the annotation on the server, a dedicated client is selected from the available collaborators. The selection is implemented using a score based on the device profile. Currently, mobile devices get lower scores due to their processing power, but the score can be adjusted to meet other requirements. If two or more devices have the same profile score, a decision is taken randomly.

On the backend, we employ a powerful REST-based microservice architecture with a security layer powered by OpenID Connect. OpenID Connect is a recent single sign-on standard that supports authentication and authorization of end users. The available videos are returned by a scalable cloud storage service based on OpenStack Swift. Another microservice accepts requests to save and retrieve the video annotation data in a MySQL database. Overall, our architecture is highly modular on both the HTML5 UI frontend with Web Components, as well as on the backend implemented as REST-based microservices.

3 Evaluation

We performed both a comparative technical evaluation as well as a user study to prove our concept. For the technical part, we measured which annotation type produced the biggest latency, which proved to be the creation of text objects with an average latency of 238.2 ms. The lowest latency were obtained for movements and rotations, with an average of 46 ms.

[2] see https://www.polymer-project.org/
[3] see http://fabricjs.com/

For the usability test, 15 people in groups of three people in a lab were asked to collaboratively draw certain annotations. Each user had to accomplish different tasks like highlighting certain objects in the video using rectangles and circles, pointing to objects using arrows, adding text annotations on screen, browsing through existing annotations using the video timeline and moving various annotations on the screen. During the test we encouraged users to talk to each other, to see how the speed of actions was perceived; in terms of this, the dragging feature was rated the highest. Furthermore, the tool was rated positive in terms of usability and performance.

4 Conclusion and Future Work

In this demo, we presented our prototype for drawing annotations on Web videos. The vector-based drawings are synchronized across devices using the infrastructural findings of our earlier DireWolf prototype and by employing Yjs, a newly developed Web-based collaborative editing framework.

Future work includes the generalization of the drawing canvas to be applicable to still images, live WebRTC video streams and other underlying content. Our evaluation study furthermore suggested adding more awareness features.

We are currently rolling out our prototype to real construction sites to further evaluate the applicability of our scenario and the technical approach. Finally, we want to use the prototype in a wide variety of Web applications ranging from the learning domain over requirements engineering for annotating screencasts.

Acknowledgements. The work has received funding from the European Commission's FP7 IP Learning Layers under grant agreement no 318209.

References

1. Koren, I., Bavendiek, J., Klamma, R.: DireWolf Goes Pack Hunting: A Peer-to-Peer Approach for Secure Low Latency Widget Distribution Using WebRTC. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) ICWE 2014. LNCS, vol. 8541, pp. 507–510. Springer, Heidelberg (2014)
2. Nicolaescu, P., Klamma, R.: SeViAnno 2.0: Web-Enabled Collaborative Semantic Video Annotation Beyond the Obvious. In: 12th International Workshop on Content-Based Multimedia Indexing 2014, pp. 1–6. IEEE (2014)
3. Glazkov, D., Ito, H.: Introduction to Web Components. Technical report. http://www.w3.org/TR/components-intro/ W3C (2014)
4. Heinrich, M., Lehmann, F., Springer, T., Gaedke, M.: Exploiting Single-User Web Applications for Shared Editing - A Generic Transformation Approach. In: Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S. (eds.) 21st International Conference on World Wide Web 2012, WWW 2012, pp. 1057–1066. ACM, New York (2012)
5. Nicolaescu, P., Jahns, K., Derntl, M., Klamma, R.: Yjs: A Framework for Near Real-Time P2P Shared Editing on Arbitrary Data Types. In: Cimiano, P., Frasincar, F., Houben, G.J., Schwabe, D. (eds.) 15th International Conference on Web Engineering. LNCS, vol. 9114. Springer, Heidelberg (2015)