ColLab Sketch

Case study on collaborative sketching

Kateřina Nováková[1], Vladimír Jakubal[2], Henri Achten[3], Dana Matějovská[4]
[1,3,4]Faculty of architecture CVUT Prague
[1]http://www.katerinanovakova.cz, [2]http://www.bitwise.cz, [3,4]http://www.molab.eu
[1]bruhokat@fa.cvut.cz, [2]vladimir.jakubal@gmail.com, [3]achten@fa.cvut.cz, [4]dana.matejovska@gmail.com

Abstract. *In this paper we present an application we developed for collaborative sketch sharing within a design process. We review the specific application development process and discuss the features of the application itself. The tool has been tested and used in a design studio setting between two universities located in different countries. We observed that it is suitable for architectural communication, and also allows monitoring of the sketch activity during the design process. This paper also describes application architecture and selected technologies. We have furthermore defined multiple groups of application requirements. Our self-developed application was proven to suit specified needs and overcame previously tested commercial tools.*
Keywords. *Programming; sketching; communication; collaborative design.*

INTRODUCTION

Remote communication became a very important part of modern methods of teaching. Activities like online collaboration among universities, as well as remote teaching and learning are increasingly becoming an important and regular part of the workday along with technology development and new software tools for collaboration and cooperation. Within a domain of architecture, remote communication means not only audiovisual connection, but also a sketch exchange.

This paper is devoted to a description of an application developed as a result of needs defined during a remote collaboration among students of architecture who have been using various computer platforms in different locations throughout Europe. The authors were running a collaborative design studio during the summer semester 2011/2012, where students from ETH Zürich collaborated with students of ČVUT Prague (Nováková and Achten, 2012).

We primarily tried to use Adobe Connect, which we could not use on various platforms, because it's sketch transition appeared to be too slow. Based on the defined requirements we developed a program for simple sketching and live sketch sharing called CollabSketch. CollabSketch has been successfully tested on various platforms with both collocated and dislocated collaborators.

REMOTE SKETCHING AS A MEAN OF COMMUNICATION

The importance of sketching during the design process is already well described in various papers of different era (Ullmann et al., 1990; Maher et al., 2005;

Denzer and Gardzelewski, 2011). In 2010 Tang, Lee and Gero proved that when traditional pen-and-paper environment simulated, there is no significant difference in the design process in its initial phases between traditional and digital media. Furthermore sketch collaboration among professionals within computer supported collaborative design was researched by ARIAM-LAREA laboratory at the Superior National School of Architecture of Paris-LaVillette. They conducted experiments with a computer tool called "Studio Digital Collaboratif," which is an interactive digital platform made of desktop, videoconference device and software called "Sketcha." (Rajeb, 2010) When developing our software we shared the idea of separating the sketch tool from the videoconference tool. Furthermore, study on interactive computer supported collaborative design environment by H. Bier (2012) was investigated, where CoLLab sketch fit as a supplementary addition to much more complex programs. Nevertheless Bier mentions problems, that our application seems to target: availability of the software, easiness of use and possibility to connect a number of users without slowing the interactive digital process.

Together with our experience as architects and teachers we can observe that students of architecture enjoy sketching not only in the initial phase, but throughout the whole design process. Furthermore students use sketching as an important mean of communication both during collocated sessions (using old-fashioned means like paper and black-board) and dislocated sessions (using special software tools and hardware equipment) (Nováková et al., 2012).

These observations can be further corroborated with our experiences that came out of a Parametric modeling workshop that we ran in the winter semester of 2012/2013 (Nováková et al., 2013). We have noticed a close connection in the ideation process between hand sketches and parametric modeling. We may conclude that CAD programs do not substitute the role of sketching. More specifically, some of sophisticated parametric modeling tools even support the hand-sketched design (Naya at al., 2008).

APPLICATION REQUIREMENTS

We have identified multiple needs that software tool has to fulfill for a collaborative design project. Some of them are obvious, others were identified based on the user experience during our first trials. We have divided all requirements into four groups:

- platform requirements (including both hardware and software environment)
- performance requirements (for both network exchange and GUI performance)
- feature requirements (what tools shall the application offer)
- licensing requirements (both pricing and multi-user limits)

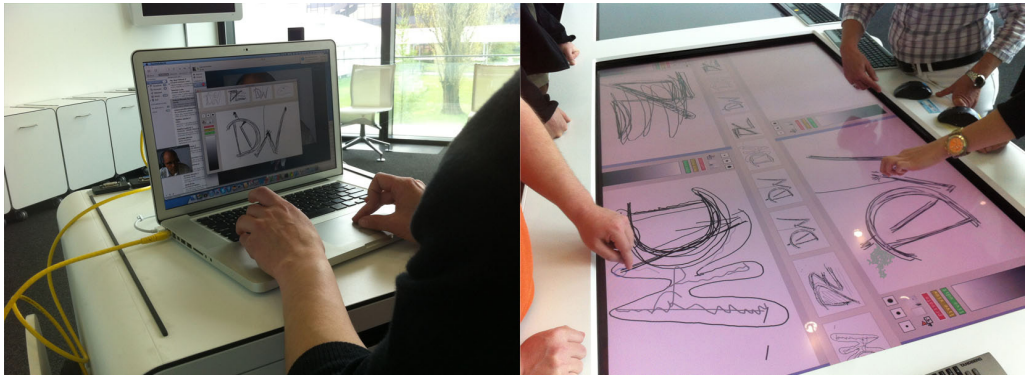
Platform requirements

Our expert group was facing very specific platform requirements - the ValueLAB at ETH Zurich is equipped with PQLabs' 65" touch displays with multi-touch interface, whereas MOLAB at ČVUT in Prague and other participants mostly use common personal computers with mouse as a pointing device. Operating systems varied from different versions of Microsoft Windows (TM) to Apple Mac OSX (Figure 1).

Feature requirements

The project of collaborative sketching application has focused on remote sketching as a mean of communication. The list of features to be implemented includes the ability of sketching, sharing a sketch via common library to all other connected users, downloading a sketch from a common library for editing and uploading back as a new version (branch), ability do watch somebody's process of sketching (what we call "live view") and support for sketching onto the same canvas ("live edit" mode). Library itself also works as a persistent storage after users disconnect and among it's required features belong support of separated sessions (for different lessons and/or groups of users), basic user management and web-based sketch viewing and management (export to common file format, deleting unneeded sketches etc).

Figure 1
Various devices used for testing.



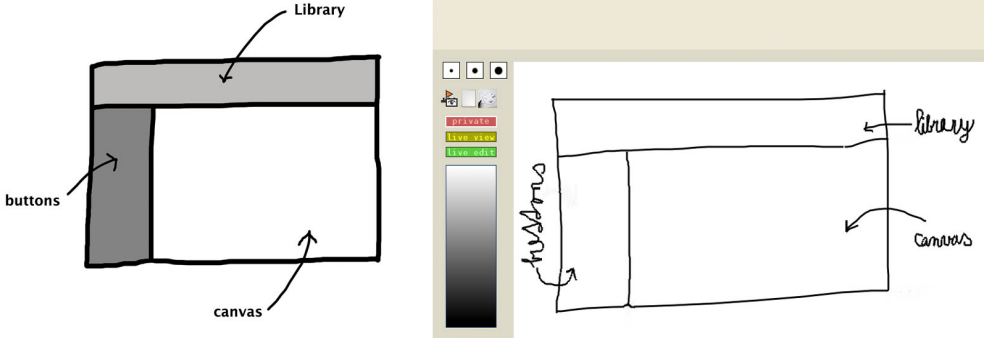
From low priority feature requirements, simple un-do, replaying a process of sketching, using raster image as a background for sketching and basic settings of width and color for painting can be mentioned. It was very surprising outcome from our primary testing, that students do not tend to use different colors during sketching, which may correlate with old-fashioned pencil black-and-white tradition. Thus we decided to keep our user interface as simple as possible (Figure 2).

Performance requirements

The sketching must have a fairly good response time (0,1 - 0,2 sec) on the sketching device, otherwise the user loses a connection between hand action and perceived sketch. There is an understandable requirement on sketching precision, which is device dependent. In a multi-platform environment it means we have to deal with different degrees of precision, especially when using touch-screen interface where users' pointing location is not known until the line is drawn (< 0,3 cm or similar to line width).

Since the sketches are used as a mean of communication, sharing the sketches via network needs to be ensured within a sufficiently short period of time (Table 1). This requirement is emphasized when users want to preview a process of sketching in a live mode, or even when they want to sketch into the same canvas at the same time from various locations.

Figure 2
Left to right: user interface wireframe, working application with features.



Type of operation	Requested time period
Sharing sketch via library	< 5 seconds
Update of live view	< 0,5 second
Update of live edit	< 0,5 second

Table 1
Network performance requirements.

Licensing requirements

One of our goals is to make CollabSketch as widely available as possible. As a process of software development obviously can't be done without any costs, pricing conditions shall correspondent with a frequency and academic area of usage. The application shall be available as SaaS (software as a service) with pay-per-use, pay-per-user or pay-per-course option for a fair price covering development expenses.

SOFTWARE DESIGN

Based on previously mentioned requirements, CollabSketch was designed based on a distributed application architecture, with separated client application for users' sketching using uniform GUI (CollabSketch), server application allowing sketch interchange among connected users as well as storing sketches persistently and independently into database (CollabCentral), and web-based application for viewing saved sketches as well as for simple user and library management (CollabWeb).

Client-server communication model has been prioritized over peer-to-peer model, as there would have been too many network connections to deal with and, due to vast use of network address translation (NAT) and packet filtering, to make peer-to-peer model work would be nearly impossible to achieve. For live-view and multi-access live-edit modes it is preferred to send incremental sketch changes and to use vector implementation of sketched lines.

CHOSEN TECHNOLOGIES AND DESCRIPTION OF COMPONENTS

To make it easier to develop client application for use in heterogeneous environment and to facilitate a communication between client and server, we have decided to use Java programming language with excellent support for multi-platform interoperability for both CollabSketch (users' sketching tool with uniform GUI adapted for different platforms) and CollabCentral (daemon application for connecting, sketch interchange and storing sketches into database) (Figure 3).

CollabSketch is furthermore based on an open-source Java framework MT4j (Multi-touch for Java) [1] which can deal with multi-touch devices [2] in ValueLAB, ETH. MT4j supports multiple protocols and drivers for various multi-touch devices and platforms, providing unified input events and gesture events programmatic access. MT4j also includes popular Processing library with OpenGL support integration.

CollabCentral is a multi-threaded application that maintains separated message queue for every connected CollabSketch client. This component is responsible for redistributing incoming messages, user authentication and persistent database storage via common JDBC driver.

CollabWeb application is a stand-alone tool for web-based viewing and managing stored sketches, users and sessions, connected with the others only

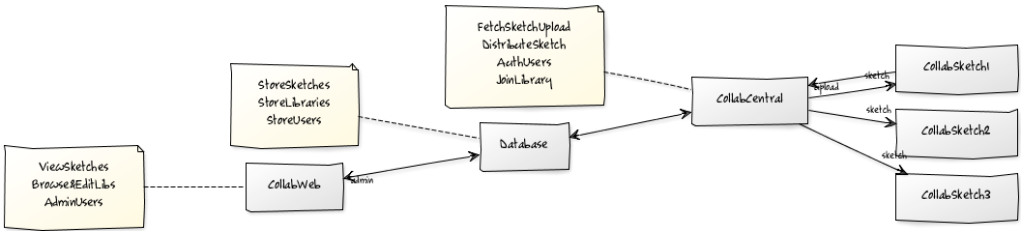


Figure 3
Collab architecture: component diagram.

Table 2
Tested platforms and configurations.

Location	Operating system	Pointing device
ValueLAB	Windows XP	PQ Labs multi-touch (4-user interface)
Molab	Windows 7	Mouse controlled
Students	Mac OSX	Mouse / touchpad

by the persistent database storage, thus it is technologically independent on CollabSketch and CollabCentral. We have decided to develop CollabWeb using a famous PHP programming language using a GD library for creating images programmatically [3]. The database storage itself runs on MySQL [4] database engine. All of these technologies are for free, mostly available as an open-source software.

DEPLOYMENT AND TESTING ON DEVICES

CollabSketch application is available as a portable application packed in single ZIP file as a bundle of CollabSketch Java class files with application resources, supporting Java and native libraries. There is also 32-bit Java runtime environment for Windows with 64-bit JRE bundled, because 32 bit version of Java is required by CollabSketch libraries. Application can be downloaded and extracted to whatever location on a hard drive or even flash disk and started according to a target platform.

There are three variants of CollabSketch at this time: one for ValueLAB multi-touch devices for four users working simultaneously using touch gestures, one for Windows-based computers with mouse or touchpad interface and one for Mac OSX mouse or touchpad interface. We have successfully tested these variants on various devices with different configurations (Table 2).

PERFORMANCE ISSUES AND SYSTEM REQUIREMENTS

The application previously described has been thoroughly tested during it's usage for sketch-based communication between ETH in Zurich and CVUT in Prague. Our team has been facing several issues regarding specific environment configuration and conditions.

Both participating universities are using very strict network security settings like disabling IP connectivity for BYODs ("bring your own devices") in local networks or packet filtering of outgoing connection. Thus we have decided to run the server part of our application (CollabCentral) in our own environment with full control of security settings. We were forced to use generally allowed port used for HTTP protocol instead of originally proposed application-specific one. Later we had to move sketch thumbnails in CollabSketch's library from vector graphics to raster images due to enormous OpenGL rendering workload on computers with software rendering emulation.

Our multi-platform client application has following system requirements:

- Java runtime environment 6.x or higher (windows version uses 32-bit JRE)
- reliable IP connectivity with outgoing connection to port number 80 enabled
- pointing device
- OpenGL support (either software or hardware driven)

CONCLUSION

CollabSketch functions in multi-platform setting. It enables CollabSketch application allows users to share their ideas in a sketched form with a minimal delay for optimal communication speed. Created application also meets specified requirements. Some of performance requirements depend on other aspects like network latency and computer performance.

FUTURE DEVELOPMENT

Our ongoing experiments focus on Android platform and iOS - based devices. We also target on HTML 5 technology, because Java runtime environ-

ment is not available for iOS platform. HTML 5 application can be also more light-weight that Java application based on MT4j library.

ACKNOWLEDGEMENT

We thank to iA ETH in Zurich and Fa CVUT in Prague for technological support and the stuff of both universities for assistance.

REFERENCES

Bier, H (2012) 'Internet-Supported Multi-User Virtual and Physical Prototypes for Architects in Architectural Education and Research' in JL Moore (Ed.), *International Perspectives of Distance Learning in Higher Education,* InTech, pp. 317 – 332.

Denzer, A and Gardzelewski, J, 2011, 'Drawing and Modeling: Analog tools in the Age of BIM' *in AEI*, pp. 44 – 53.

Maher, ML, Bilda, Z, and Marchant D, 2005 'Comparing Collaborative design behaviour in remote sketching and 3D virtual worlds' in Proceedings of International Workshop on Human Behaviour in Designing, pp.3-26.

Naya, F, Contero, M, Alexois, N, Jorge, JA and Company, P. 2008 'Sketch Based Interfaces for Parametric Modeling' in I Pavlidis (ed.), *Human Comuputer Interaction,* In Tech, Rijeka [5].

Nováková, K and Achten, H, 2012, 'For Students: Sketching Apart Together'*, Physical Digitality - Proceedings of the 30th International Conference on Education and research in Computer Aided Architectural Design in Europe*, Fakulta architektury ČVUT Prague, Vol. 2 pp. 77 – 85.

Nováková, K, Achten, H, Treyer L and Schmitt G, 2012, 'Value Lab: Innovation in Teaching Visual Design: There is Nothing to Wait for', *Proceedings of 16th International Conference Information Visualisation,* IEEE Computer Society [online] Montpelier.

Nováková, K and Achten, H, 2013, '150 000 – Parametric Control of PET Bottle Structure', in R Stouffs and S. Sariyildiz (eds), *Computation and Performance – Proceedings of the 31st eCAADe Conference – Volume 2*, eCAADe, Brussels, and Faculty of Architecture, Delft University of Technology, Delft, pp. 555-562.

Rajeb, BS, Lecourtois, C and Guéna, F 2010 'Operations of Conception in Architectural Collaborative Design', *Proceedings of the eCAADe Conference*, Zurich, Switzerland, pp. 687-695.

Tang, HH., Lee, YY and Gero, JS., 2010, 'Comparing col- laborative co-located and distributed design processes in digital and traditional sketching environments:A protocol study using the function-behaviour-structure coding scheme' Elsevier Ltd.

Ullman, DG, Wood, S and Craig, D, 1990, 'The Importance of drawing in the mechanical design process', *in Computers and Graphics*, Elsevier, Corvallis, USA, pp. 263 – 274.

[1] www.mt4j.org
[2] www.pqlabs.com
[3] www.libgd.org
[4] www.processing.org
[5] www.intechopen.com/books/human_computer_interaction/sketch-based_interfaces_for_parametric_modelling