2016 International Conference on Collaboration Technologies and Systems

Tele-Board Prototyper - Distributed 3D Modeling in a Web-Based Real-Time Collaboration System

Matthias Wenzel*, Adrian Klinger†, Christoph Meinel‡
Hasso Plattner Institute Potsdam
Prof. Dr. Helmert Str. 2-3, Potsdam, Germany
Email: *matthias.wenzel@hpi.de, †adrian.klinger@hpi.de, ‡christoph.meinel@hpi.de

***Abstract*—Prototypes help people to externalize their ideas and are a basic element for gathering feedback on an early product design. Prototyping is oftentimes a team-based method traditionally involving physical and analog tools. At the same time, collaboration among geographically dispersed team members becomes more and more standard practice for companies and research teams. Therefore, a growing need arises for collaborative prototyping environments. We present a standards compliant, web browser-based real-time remote 3D modeling system. We utilize cross-platform WebGL rendering API for hardware accelerated visualization of 3D models. Synchronization relies on WebSocket-based message interchange over a centralized Node.js real-time collaboration server. In a first co-located user test, participants were able to rebuild physical prototypes without having prior knowledge of the system. This way, the provided system design and its implementation can serve as a basis for visual real-time collaboration systems available across a multitude of hardware devices.**

***Keywords*—remote collaboration; prototyping; 3D; real-time; JavaScript; WebGl**

I. Introduction

Prototyping is a widely used method that helps people to illustrate their ideas and externalize implicit knowledge [1]. A prototype represents different states of an evolving design [2] and communicates those to other stakeholders [3]. Prototyping may also result in competitive benefits when customers and suppliers are involved in the process [4].

Within innovative organizations, which are tending toward more prototype-driven product specifications [4], the process of generating ideas that become manifest in later prototypes is often a team-based activity with numerous participants. At the same time those teams, especially in corporate environments, are becoming increasingly distributed in time and space. It is a development that is likely to become more commonplace in the future. Thus, we need digital tools to facilitate the development of joint prototypes. Furthermore, the availability of these tools across a wide range of devices has to be ensured. Remote collaboration tools have to be accessible from traditional computer systems, such as desktop computers and laptops, as well as from mobile devices, such as smartphones and tablet computers. Modern web browsers provide the basis for cross-platform software systems as capable and powerful as desktop applications [5].

With Tele-Board, we already developed a web browser-based application that facilitates real-time remote collaboration on the basis of a two dimensional workspace where users can create and manipulate shared artifacts, such as sticky notes, drawings or images. We learned that this setup is especially useful for idea generation and feedback sessions over distances [6]. However, generating ideas oftentimes entails the expression of selected ideas in a tangible way. Although, there is no universal vocabulary for prototyping, prototypes can be classified based on different characteristics, e.g. analog/digital, static/interactive, early stage/later stage, low fidelity/high fidelity etc.

In this paper, we present Tele-Board Prototyper, a standards compliant, web browser-based real-time remote 3D modeling system. With our prototyping support, we target the field of digital 3D low-fidelity prototypes. The intended use case of our proposed 3D modeling application is the collaborative development of early stage three dimensional models that serve as discussion basis and template for a later stage and more detailed and sophisticated development. Our goal is to create a system that facilitates the transition from early, sketching dominated ideation to the point when it comes to the testing of these ideas. Compared to traditional asynchronous collaboration techniques (e.g. email or shared file storages), our approach of a synchronous real-time system offers a direct reference to the shared artifacts and allows collaboration in the context of the actual model. The proposed Prototyper application fully integrates into the mentioned Tele-Board system, following its paradigm of simplicity.

II. Related Work

Collaborative prototyping has been a field of research as well as a domain for providers of respective products. We give an overview of research ideas and describe commercial as well as open source systems.

A. Research Projects

Research projects range from testing of virtual prototypes over integrating frameworks, including manufacturing processes to collaborative design and research regarding haptic interaction techniques. The subject of the system proposed by Lee [7] is the representation of 3D *Computer Aided Design*

978-1-5090-2300-4/16 $31.00 © 2016 IEEE
DOI 10.1109/CTS.2016.82

(CAD) models and their testing in accordance with certain product requirements. The architecture, which is consequently necessary, and data exchange is elaborated. Tay et al. [8] describe a framework for providing manufacturing facilities to geographically separated product development stakeholders. The authors use the term “internet manufacturing” to summarize their approach wherein different proprietary software components are linked together over the internet in order to provide access to prototyping hardware for creating physical prototypes. A synchronous collaboration during modeling is not part of the system. A similar, but much more complex, approach is proposed by Schaefer et al. [9] who take a step further binding together necessary components to support the whole product lifecycle. Li et al. [10] describe a system that allows co-located collaborative 3D model design. Their system provides a web-based visualization component supporting product preview and evaluation of design parts. However, the remote users cannot modify the model. The system relies on client side Java technology and requires respective applets to be executed in a web browser. Another research field focuses on synchronized haptic feedback on physical 3D objects [11], [12]. The actual creation of 3D objects is not the main topic in this area. Instead, the investigation focuses on how to synchronously manipulate distributed physical objects.

The mentioned research projects differ from our approach. The process of collaboratively creating three dimensional prototypes in real-time in a cross-platform web browser environment is not an area of investigation of any of the research projects.

B. Commercial and Open Source Products

SketchUp[1] is a computer program for generating 3D models. It is a proprietary system that is available for desktop computers running either Windows or Mac operating systems. It offers a multitude of functions for creating and manipulating 3D models. Though models can be shared among different systems, it is not possible to work simultaneously on one model. *Clara.io*[2] is a browser-based tool focusing on the creation of detailed, complex 3D models and photorealistic renderings. By virtue of its extensive user interface it aims at more advanced users. The system provides a basic, simultaneous multi-user editing. Another web browser based tool is *Tinkercad*[3]. The system provides a large set of basic shapes and import/export formats in a clear user interface. Generated 3D models can be uploaded to a public web platform and can be reused by others. The Tinkercad system does not allow a synchronized working mode. The open source system *Shapesmith*[4] is also used in a web browser context. The software provides a comprehensible user interface with a limited set of basic shapes and tools making it especially interesting for beginners. At the same time it is very flexible since complex 3D objects can be created out of basic shapes combined to complex structures with the help of logical operations. It also supports import and export of 3D models in STereoLithography (STL) format. A collaborative working mode is missing in Shapesmith.

The described systems either do not provide a synchronous working mode or have an extensive UI focusing on an advanced group of users and use case. Our approach aims to fill this gap allowing real-time remote collaboration on early stage 3D models with an optimized UI.

III. Tele-Board - A Whiteboard Application for Synchronous and Asynchronous Settings

Tele-Board [13]–[15] is a web browser-based real-time remote collaboration system. The application provides a shared workspace in the form of a virtual whiteboard surface. The content data is synchronized automatically by a central server among all connected client applications as shown in Figure 1.

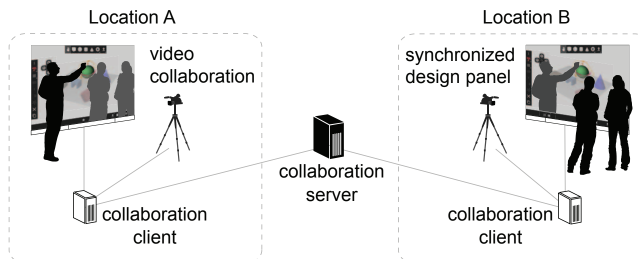

Figure 1. Tele-Board system architecture. Virtual whiteboard data is synchronized among connected clients by the central collaboration server. Remote users can see pointing gestures and facial expressions by system provided video conferencing.

The system consists of three different parts. The *Web Portal* is the entry point of the Tele-Board system. It serves as an administration interface allowing users to manage projects and associated panels in order to organize their work and control access rights. User accounts are assigned to projects which grants access to the projects’ panels. A panel represents a virtual whiteboard including its content and its course over time. It is therefore possible to go back and forth in the history of the whiteboard content [16]. Started from the web portal, the *Whiteboard Client* allows editing of a panel. The application, written in JavaScript, requires no additional browser plugins. It facilitates whiteboard interaction (e.g. writing with different colors, erasing, and the manipulation of sticky notes and images). Whenever a user starts the whiteboard client with a particular panel, the system gets automatically connected to all other clients operating on the same panel. All users can see remote panel actions in real-time and are equally authorized to manipulate any panel artifacts. The *Collaboration Server* component coordinates all communication between the remote partners. Whenever a whiteboard artifact is created or changed a serialized object representation in *JavaScript Object Notation* (JSON) is forwarded by the server to all other connected whiteboards to keep these synchronized. In order to keep track of all whiteboard changes, our system performs

[1] https://www.sketchup.com/
[2] https://clara.io/
[3] https://www.tinkercad.com/
[4] http://shapesmith.net/

synchronization in real-time (i.e. when moving a sticky note on the panel each change in its position is propagated to all connected clients). This way, the remote participants can see the movement of the sticky note not only its final position.

IV. Prototyper - Application Design and Implementation

The primary purpose of Prototyper is the user controlled generation and manipulation of 3D models as a digital counterpart to building these in the "analog" world. Models can be built with a small set of basic 3D objects, that in turn can be combined to more complex objects. This also includes object manipulations such as moving, rotating and scaling. The viewing direction and zoom level can be changed to any value in order to look at specific aspects of the designed model. To provide a visual context when discussing the prototype, a mouse or other digital input device can be used to point at different model parts. All these operations should be synchronized to all other participants in real-time enabling users to better keep track what currently happens on the workspace. In order not to limit Prototyper's application spectrum to a closed digital environment, it was important for us to provide a "bridge" to other systems and physical prototypes. This allows building upon existing 3D models (e.g. digital models created with a CAD tool or physical items digitized with a 3D scanner). Though not widespread at the moment, this scanning technology will become more commonplace when devices get more affordable, and it may even become available on mobile phones [17], [18]. For the other direction, a 3D printer can be used to create a physical object in order to get a haptic feedback on a digital 3D model. For covering these cases, a further objective for Prototyper was to support the import and export of 3D model data.

Supplementing our existing Tele-Board system, Prototyper was integrated with regards to current web standards independent of further browser plugins. Furthermore, it should be possible to support an asynchronous working mode, specifically to keep a server-side "history" of a model in order to follow up the origination of a prototype. Based on these requirements, the following main tasks were solved throughout our implementation of Prototyper: (1) Choosing suitable technology for rendering three dimensional objects in the web browser; (2) Implementing a JavaScript based Single-Page-Application and corresponding server backend for building and interacting with 3D models; (3) Developing a real-time synchronization mechanism allowing a distributed working setup where all users are equally entitled to manipulate a 3D model; (4) Server-side storage of synchronization data to track 3D models' course of development.

A. 3D Modeling in the Web Browser

In the course of HTML5 a multitude of new features became available to application programmers. These encompass sophisticated and powerful 3D rendering capabilities in a web browser. At present, the most widespread technique for web browser-based 3D rendering is *Web Graphics Library* (WebGL). WebGL is a low-level immediate mode JavaScript 3D rendering API providing *Graphics Processing Unit* (GPU) accelerated rendering and is nowadays offered by major web browsers[5]. Another technique that aims to facilitate 3D rendering in a browser is *X3DOM*[6]. X3D is an XML-based file format for declarative 3D computer graphics. X3DOM is a JavaScript library that renders declarative X3D graphics. The declarative representation of a 3D scene is embedded in the HTML markup of a web page. It is a concept that is similar to *Scalable Vector Graphics* (SVG) for two dimensional graphics in a browser. The 3D scene can be modified programatically using DOM operations such as `setAttribute(in DOMString name, in DOMString value)` [19].

Our intention in building 3D models with Prototyper implies real-time interaction and modification. We decided on the immediate mode WebGL API as the basis for our Prototyper implementation. For our system, we did not consider X3DOM and its declarative character to be suitable. It did not prove powerful enough when comparing SVG and HTML Canvas-based 2D rendering for implementing our web browser based Whiteboard Client application [13]. Once we decided on a WebGL based solution we looked for existing web browser-based 3D modeling tools that we could use to build on. As described in section II-B, there are already systems providing the basic functionality required by our application. The closest match to these requirements was the above mentioned open source (MIT license) system *Shapesmith*. Thus, we chose it as the basis for our Prototyper implementation. Shapesmith uses the *Three.js*[7] JavaScript library to access web browser provided low-level WebGL API. Three.js is a high-level 3D web browser library which eases the handling of WebGL. While using a high-level library always means a loss of flexibility, the features provided by Three.js are sufficient for our use case since we do not need fine grained 3D rendering control. It was therefore reasonable for us to keep using Three.js in our Prototyper application. Another positive aspect is that Three.js offers out-of-the-box 3D model import and export. As a first implementation step, we included Shapesmith's web page markup and its corresponding JavaScript and CSS dependencies into the Tele-Board web portal. Based on the existing *Whiteboard Client* architecture, Prototyper is also a single-page-application provided by a view in the web portal's *Model-View-Controller* (MVC) framework. The original Shapesmith components were extensively changed and extended during the integration of Prototyper's data model. We also removed some unused components and libraries and updated to the most recent Three.js version. For our initial version we added features for colorizing and texturing 3D models. A view of the final Prototyper system is shown in Figure 2.

[5]http://caniuse.com/\#feat=webgl
[6]http://www.x3dom.org/
[7]http://threejs.org/

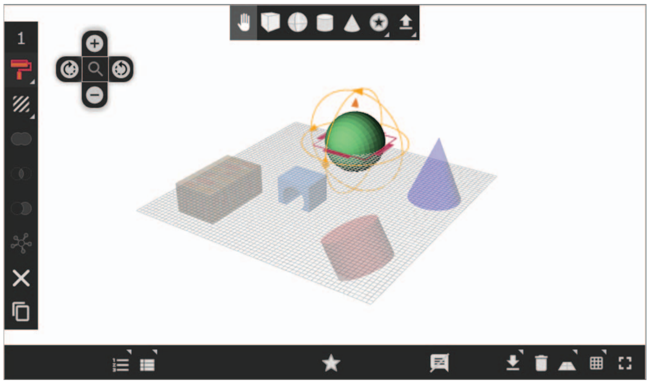

Figure 2. Prototyper user interface visible in a web browser.

The UI design derives from our 2D whiteboard client application. Control elements directly relate to the current view state and are hidden when not needed: when an object is selected (e.g. the sphere in the figure) by the user, a menu on the left is shown that provides options for its modification (e.g. coloring, texturing or combination with other selected objects). The control elements for the sphere's transformation (specifically, rotation and scaling) are also only shown after selection. The navigation bar on the top left was added to allow Prototyper's usage also on devices that are not equipped with a mouse, such as touch screens or digital whiteboards. Some control elements in the figure relate to features that we have integrated later as a result of the user test described in the upcoming section V.

In order to facilitate a real-time synchronization of 3D model data, respective messages have to be sent and received. To ensure that the message processing does not interfere with the user's 3D model interaction, these two parts are run in separate threads within our Prototyper implementation. We rely on HTML5 *WebWorker* API that allows to run a script in the background independently of user interface scripts [20]. With the help of this API it is possible to spawn an operating system level thread that encapsulates our networking and message processing code.

B. Real-Time Synchronization and Server-Side Storage

The goal of our Prototyper system is to enable geographically dispersed teams to simultaneously work together at 3D models. Therefore, modifications of a digital prototype at one location have to be transferred to all connected systems showing this specific prototype. We implemented this synchronization mechanism using the same real-time approach as in our 2D whiteboard application. Any changes of a 3D object or the workplane are synchronized immediately among all involved instances, i.e. users can follow an object's movements and do not see only its final position. As we learned from our whiteboard application this helps to better keep track of an artifacts modification.

Realizing this feature requires a multitude of synchronization messages in both, incoming and outgoing direction. The persistent, bi-directional *Transmission Control Protocol* (TCP) based *WebSocket* [21] protocol and the corresponding web browser API have proved to be very efficient and suitable [22] especially with regard to traditional HTTP-based techniques. Besides the conventional web server delivering HTTP content, such as web pages, CSS and JavaScript resources, we implemented a dedicated central collaboration server that handles 3D model data synchronization. The system's architecture and communication setup is shown in Figure 3. The two services provided by the server part of our system also require two different ports in order to identify these services that also have to be reachable from the client's browser. Sometimes this can cause problems with restrictive firewalls blocking a non-standard port defined for the collaboration server. In order to avoid this, we are using the *Nginx*[8] proxy server that handles both, the web portal (HTTP) and the collaboration server (WebSocket) communication utilizing only one port. Therefore, the encrypted protocol versions *HTTP over TLS* (HTTPS) and *Secure WebSocket* (WSS) can be used with the standard port 443 [23].

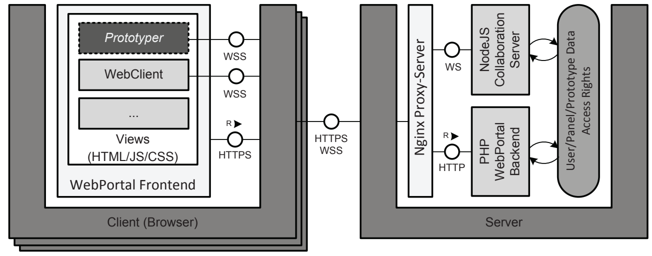

Figure 3. The Tele-Board client-server architecture. Web portal and 3D model synchronization is managed by an Nginx proxy server using a single standard port 443 for encrypted communication.

The collaboration server is written in JavaScript and utilizes the server-side *Node.js*[9] runtime environment. We use *Socket.IO*[10] JavaScript library for connection management and synchronization message relay, essentially broadcasting to all Prototyper instances editing the same 3D model. Hence, modifying a 3D object on the workplane triggers a message with object's serialized JSON representation which in turn updates the corresponding remote systems' objects. In order to facilitate an additional asynchronous working mode where users can follow up a 3D model's course of development, a server-side storage for all 3D model's modifications is needed. The collaboration server therefore stores communication message data in a server database together with a timestamp and the belonging prototype identifier. We adopted this approach from our 2D whiteboard client application [16], [24]. This concept allows us to build the complete 3D model state including all its containing objects at a given point in time. We use this data for a "history" view of a specific 3D prototype.

[8]http://nginx.org/
[9]https://nodejs.org/
[10]http://socket.io/

C. Time Travel: The Prototype's Course of Development

When the Prototyper application is started in the web browser, the most recent state of the respective 3D model is loaded automatically. However, sometimes it can be helpful not to solely rely on the final state of the model, but to also see how the prototype evolved from its beginning. This is especially the case when a member of the working team cannot attend a working session and wants to catch up on the prototype's development afterwards. Another use case relates to taking decisions concerning which direction to evolve the prototype. A more divergent proceeding, where multiple possible development paths are followed before refining a concept, often leads to better results [25]. A dedicated view (Figure 4) makes a prototype's history explorable.

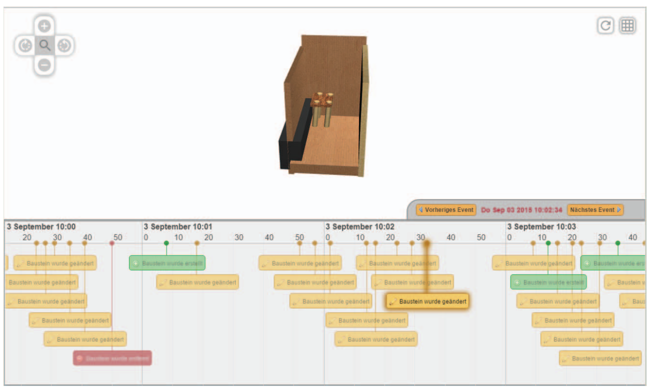

Figure 4. The Tele-Board Prototyper history browser. Users can navigate to any point in time of a prototype's course of development using the timeline. A branch of a prototype can be created at a specific state in order to follow a different development path.

The read-only history view (rotation and zooming is provided) displays a prototype's state at any given point in time with the help of the timeline at the bottom. The level of detail of this timeline can be changed by the user from a broad to a fine grained level. Since each synchronization event is stored in the database together with its temporal occurrence, this level can be reduced to one second. This way, a user can click through the whole history of a prototype. Furthermore, arriving at the required time position, a branch of the prototype and its state at the given position can be made in order to follow a different development path.

V. Evaluation - Feedback on 3D-Modeling with Prototyper

The process of commonly creating 3D prototypes among geographically dispersed teams is the main objective for implementing Prototyper. However, before first distributing this three dimensional content, this data has to be created by users locally. As a first evaluation step, we wanted to elaborate on the tools and functions provided by Prototyper to build actual prototypes. For this, we conducted a test where users should digitally reconstruct physical prototypes in a local setup within a set time. With the help of the experiment, we wanted to answer the following questions: (1) Can the prototypes be rebuilt in the given time allowing us to make conclusions to be drawn related to our goal of an easy to use system? (2) Which of the provided components and functions work well and which do not? (3) Is there any additional functionality that could further improve the system but is currently not provided? These initial prototypes were built in the context of real *Design Thinking* projects that were carried out in 2013/2014 at HPI School of Design Thinking, University of Potsdam, Germany[11]. Design Thinking at d.school can be briefly summarized as an innovation method involving multidisciplinary teams of about five people. Team members use a set of different creative methods in order to solve complex problems [26] based on an iterative process with specific steps. It is a working method we also aim to support with Prototyper in a remote setting.

A. Test Design and Probands' Tasks

Each participant of the user test had to individually rebuild three existing physical prototypes based on the photographs shown in the first row of Figure 5.

The tests were run on a Windows 7 desktop PC using Google Chrome 44 web browser. Each test started with a 10 minute introduction of the Prototyper system. Participants were shown how to work with the user interface and which component has which meaning. This introduction was the same for every participant. After that, the participants had to successively rebuild the three prototypes on their own within a time of at most 20 minutes per prototype. Right after the experiment we handed a questionnaire to the test participants. They had to answer Likert-Scale questions regarding the general usage of the system. It was also necessary to complete free text fields with a special focus on the comparison between the analog and digital prototypes, and the functions and working modes that Prototyper supports.

Six persons (2 female) aged from 27 to 34 took part in the test. Not one of the participants had used Prototyper before. The overall time for each test run was about 90 minutes including the questionnaire (no time limit).

B. Resulting Prototypes

During the tests, every participant could build a 3D model based on the given photographs resulting in 3x6 digital models. The 20 minute time frame was exploited by all participants, with no prototype completed before the time limit. An example of each analog prototype's digital counterpart is shown in Figure 5. We evaluated the reconstruction of each prototype based on observations during the test and the final digital result.

[11]Also referred to as the "d.school", it encompasses the School of Design Thinking at the University of Potsdam, Germany and the Institute of Design at Stanford University.

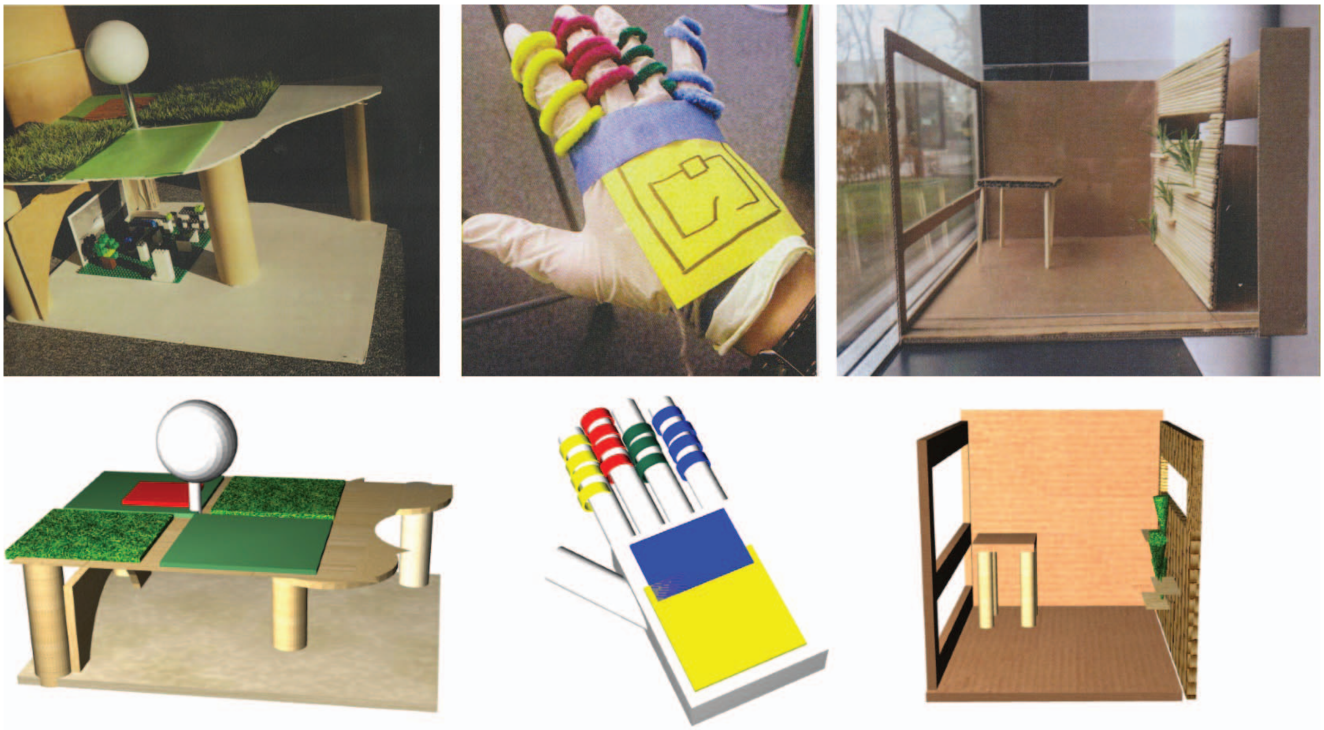

Figure 5. Physical prototypes (first row) and their digital counterparts (second row) reconstructed with Tele-Board Prototyper during the user test. The prototypes are subsequently referred to as prototype a), b) and c) (from left to right).

1) Prototype a): The main characteristics of this prototype could be rebuilt by all participants. This includes the pillars and the two floors with the sphere on top. Users tended to build these components first. The detailed structures with the Lego bricks were left out by all participants due to the time limit. The most time-consuming elements were the objects with rounded edges. We expected this to happen since these require combining multiple elements with the help of logical operations. The lawn on the second floor was modeled by all participants using a trivial cube object given a grass texture instead of creating a multiple blades of grass. We did not assess this as a mistake because it is a single delimited element clearly recognizable in its meaning. Additionally, we learned from our observations that is was sometimes difficult for the participants to correctly position objects, especially when they wanted to align these to other elements. This was mostly caused by perspective issues and the very fine grained modification resolution.

2) Prototype b): Although this prototype does not contain many different elements, it proved to be the most challenging for the participants. All of them were able to model the hand. Each with varying levels of detail ranging from the trivial object-based (palm = cube, fingers = five cylinders) to more organic results using object compositions. Putting the colorful "rings" on the fingers was difficult for most users, since these had to be first created and then positioned correctly. Here we saw the issues related to positioning and alignment that we had already noticed during the modeling of prototype a) become more obvious. The black drawing on the yellow rectangle was not transferred to the digital model, because drawings are not supported by the system. Though it would have been possible to mitigate the drawing with the help of smaller 3D objects, such as cubes and cylinders, this was not done by any participant. They spent much more time modeling the hand.

3) Prototype c): Rebuilding this prototype was not a problem for the test participants. The creation of the elements was mostly straight forward. Being the third prototype to be reconstructed, we noticed that users became more and more used to the concept of creating complex structures. On the one hand, we realized this by the time needed to apply this concept. On the other hand, this could be observed when creating the room. Instead of using separate elements for walls and the floor, we also saw this implemented by simply applying a difference operation of two nested cube objects.

C. Participant Feedback

In the questionnaire, participants were asked to rate the original prototypes' match to their reconstructed models. Users rated prototype b) with the lowest match level and prototype c) with the highest. This way the assessments of participants are consistent with our observations regarding the level of difficulty of the prototypes during the test sessions. Due to time limitation during the reconstruction phase, users had to focus on the main characteristics of the given prototypes and could not transfer all details to the digital model. Hence, no digital model was rated to be an exact match to the original prototype. The lower rating, especially for prototype b), was mainly caused by time consuming positioning and alignment difficulties. These were not only observed by us during the test but were also mentioned by the participants:

> "I couldn't see if the shapes that I created were touching, [...] For instance, when I wanted to put two surfaces on top of each other I had to change perspectives couple of times"

The positive usage aspects of Prototyper mentioned in the questionnaire mostly relate to its easy accessibility, general user interface interaction (e.g. menus, icons and their semantics) and the provided modification options (e.g. texturing).

> "[...] reduced UI, no info overload [...] usable in browser, requires no extra effort"

> "I like the three-step shape creation: starting point, ground shape, height [...] the textures that make things beautiful in no time."

> "[...] It felt fast to get to the right shapes even if they were combinations. It felt natural to play with the given shapes and modify them with the given tools."

The test participants also made suggestions on how the system's usage could be improved. According to these suggestions, it would be very helpful to reuse a digital prototype (or parts of it) as a component in other 3D models. We also assessed such an option as a helpful function, especially with regards to commonplace objects that are frequently used and require many complex building operations. In order to accelerate certain user interface operations, keyboard shortcuts were also considered to be useful during the creation of 3D models.

> "I missed some shortcuts and keyboard functions, especially when I slightly moved shapes; would be good to have the keyboards arrows"

With regards to a future asynchronous usage scenario or for getting general feedback on a prototype, a functionality for commenting on certain areas of a model could also be useful.

D. Findings and Resulting System Adaptions

Our user test focused on the immediate usage of Prototyper in a local setup. The six test participants therefore individually reconstructed three physical prototypes within 20 minutes each. The resulting digital 3D models show most of the analog prototypes' essential characteristics. Considering the short time and the fact that participants did not have any prior knowledge and experience with the system, this is a promising result for us that justifies further development of Prototyper. The test revealed that Prototyper's benefits are its easy accessibility through a web browser, comprehensive user interface interactions (e.g. menus, icons and their semantics), and the provided 3D objects modification options (e.g. texturing). However, we also saw some flaws regarding 3D object modification. The spatial alignment of objects among each other was sometimes difficult and time consuming. The creation of complex shapes with the help of logical operations is a concept users were not that familiar with since this requires more spatial imagination resulting in higher time consumption in the beginning. When users got more experienced with this concept during the test, we observed a faster usage of this tool. In order to further improve its application a preview option might help to make results of such operations more predictable.

Combining basic components to complex structures with a reduced set of operations is a commonplace pattern existing in many construction kits. Following this approach enables us to implement a quickly understandable system but contradicts the user desire for a multi-purpose system offering many out-of-the-box features. This implies more effort when composing complex structures out of basic shapes. As a consequence of the observations made during the reconstruction phase and participants' suggestions, we implemented an additional feature that allows a combination of both approaches. Besides the possibility of duplicating complex structures within one prototyping model, we implemented a function to reuse already built components also in other prototypes. When a user selects one or more 3D objects on the workplane, it is possible to store these components in the server-side database. These components are in turn available in other prototype models and can be re-created just like basic shapes.

As we learned from the user test, an important issue is the positioning and object alignment procedure. We implemented a function for easing the alignment of objects with each other. Users can activate a grid of different sizes on the 3D workplane causing 3D objects to "snap" to discrete positions while moving or resizing. In preparation for future test scenarios, we extended Prototyper with a function to comment on certain parts of the 3D model. This way, it is possible to gather basic feedback on a digital prototype.

VI. Conclusion and Outlook

Today's prevalent way of working is one that is becoming more and more distributed. With advances in computer and network technologies, there is a growing interest in collaborative remote 3D modeling systems. In this paper we presented Tele-Board Prototyper, a standards compliant, web browser-based real-time remote 3D modeling system. By extending our previously developed Tele-Board system with Prototyper's additional dimension we facilitate the further expression of such ideas over distances. Prototyper is implemented utilizing cross-platform WebGL rendering API for hardware accelerated visualization of 3D models and runs across a multitude of hardware devices. Synchronization relies on WebSocket based message interchange over a centralized Node.js real-time collaboration server. Since each synchronization event is stored in a database users can navigate through the whole history of a prototype. In a first co-located user test, participants were able to reconstruct essential characteristics of given physical prototypes without having prior knowledge of the system and only a limited time of 20 minutes. However, the test also identified potential for improvement that in particular relates to positioning and alignment of 3D objects on the workplane.

After showing that it is possible to use the system in a co-located setup, we want to test Prototyper in a synchronous remote setting as this is our intended use case. A possible setup for such a test is depicted in Figure 1. Here we want to combine the visualization of 3D models with our full body video conferencing setup [27] in order to improve participants' communication and understanding over distances. A visual context is provided by enhancing Prototyper's task space of involved artifacts (=3D workplane) by a person space of the remote partner's image and a reference space for pointing and gesturing [28]. This approach also includes the potential for increasing the support for asynchronous working modes. Our star topology-based video conferencing system allows us to record the video sessions at a central location. The combination of a prototype's history with "audio/video" history would allow better asynchronous work comprehension [29].

ACKNOWLEDGMENT

We thank the HPI-Stanford Design Thinking Research Program for funding and supporting this project. The authors are especially grateful to all participants of the user test for their feedback and time. Many thanks to Dr. Sharon Nemeth for her valuable proofreading support.

REFERENCES

[1] B. Buxton, *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, 2007.

[2] S. Houde and C. Hill, "What do Prototypes Prototype ?" *Handbook of human-computer interaction*, vol. 2, pp. 367–381, 1997.

[3] S. P. Dow, K. Heddleston, and S. R. Klemmer, "The efficacy of prototyping under time constraints," in *Proceeding of the seventh ACM conference on Creativity and cognition - C&C '09*. New York, New York, USA: ACM Press, 2009, pp. 165–174.

[4] M. Schrage, "Bringing design to software," T. Winograd, Ed. New York, NY, USA: ACM, 1996, ch. Cultures of Prototyping, pp. 191–213.

[5] A. Wright, "Ready for a Web OS?" *Communications of the ACM*, vol. 52, no. 12, pp. 16–17, Dec. 2009.

[6] R. Gumienny, L. Gericke, M. Wenzel, and C. Meinel, "Supporting creative collaboration in globally distributed companies," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 995–1007.

[7] J. Lee, "Shape Representation and Interoperability for Virtual Prototyping in a Distributed Design Environment," *The International Journal of Advanced Manufacturing Technology*, vol. 17, no. 6, April 2001.

[8] F. E. Tay, Y. P. Khanal, K. Kuen Kwong, and K. Cheng Tan, "Distributed rapid prototyping a framework for Internet prototyping and manufacturing," *Integrated Manufacturing Systems*, vol. 12, nov 2001.

[9] D. Schaefer, J. L. Thames, R. D. W. Jr., D. Wu, and D. W. Rosen, "Distributed collaborative design and manufacture in the cloud - motivation, infrastructure, and education," *Computers in Education Journal*, vol. 3, no. 4, pp. 1–16, 2012.

[10] W. Li, J. Fuh, and Y. Wong, "An Internet-enabled integrated system for co-design and concurrent engineering," *Computers in Industry*, vol. 55, 2004.

[11] K. Sakong and T.-J. Nam, "Supporting telepresence by visual and physical cues in distributed 3D collaborative design environments," *CHI '06 extended abstracts on Human factors in computing systems*, 2006.

[12] S. Brave, H. Ishii, and A. Dahley, "Tangible interfaces for remote collaboration and communication," in *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW '98*. New York, New York, USA: ACM Press, 1998, pp. 169–178.

[13] M. Wenzel, L. Gericke, R. Gumienny, and C. Meinel, "Towards Cross-Platform Collaboration - Transferring Real-Time Groupware To The Browser," in *Proceedings of the 17th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013)*. IEEE, June 2013, pp. 49–54.

[14] M. Wenzel and C. Meinel, "Parallel network data processing in client side javascript applications," in *2015 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, June 2015.

[15] M. Wenzel, L. Gericke, C. Thiele, and C. Meinel, "Globalized design thinking: Bridging the gap between analog and digital for browser-based remote collaboration," in *Design Thinking Research*, ser. Understanding Innovation, H. Plattner, C. Meinel, and L. Leifer, Eds. Springer International Publishing, 2016, pp. 15–33.

[16] L. Gericke, R. Gumienny, and C. Meinel, "Message capturing as a paradigm for asynchronous digital whiteboard interaction," in *Proceedings of the 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*. Chicago, IL, USA: IEEE Press, 10 2010, pp. 1 – 10.

[17] J. Stoller-Conrad, "New Camera Chip Provides Superfine 3-D Resolution," https://www.caltech.edu/news/new-camera-chip-provides-superfine-3-d-resolution-46425, April 2015, California Institute of Technology (Caltech).

[18] C. Naegeli, "Your Smartphone as a 3D Scanner," https://www.ethz.ch/en/news-and-events/eth-news/news/2013/12/your-smartphone-as-a-3d-scanner.html, December 2013, Eidgenössische Technische Hochschule (ETH) Zurich.

[19] A. L. Hors, P. L. Hégaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne, "Document object model (DOM) level 2 core specification," World Wide Web Consortium (W3C), W3C Recommendation, November 2000. [Online]. Available: https://www.w3.org/TR/DOM-Level-2-Core/

[20] I. Hickson, "Web Workers," World Wide Web Consortium (W3C), W3C Working Draft, September 2015, https://www.w3.org/TR/2015/WD-workers-20150924/.

[21] I. Fette and A. Melnikov, "The websocket protocol," Internet Requests for Comments, RFC Editor, RFC 6455, December 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6455.txt

[22] C. A. Gutwin, M. Lippold, and T. C. N. Graham, "Real-time groupware in the browser: Testing the performance of web-based networking," in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, ser. CSCW '11. New York, NY, USA: ACM, 2011.

[23] E. Rescorla, "HTTP Over TLS," Internet Requests for Comments, RFC Editor, RFC 2818, May 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt

[24] L. Gericke and C. Meinel, "Evaluating an Instant Messaging Protocol for Digital Whiteboard Applications," in *Proceedings of the 2011 International Conference on Internet Computing (ICOMP 2011)*. CSREA Press, July 2011, pp. 3–9.

[25] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer, "Parallel prototyping leads to better design results, more divergence, and increased self-efficacy," *ACM Transactions on Computer-Human Interaction*, vol. 17, no. 4, pp. 1–24, 2010.

[26] R. Buchanan, "Wicked Problems in Design Thinking," *Design Issues*, vol. 8, no. 2, pp. 5–21, 1992.

[27] M. Wenzel and C. Meinel, "Full-Body WebRTC Video Conferencing in a Web-Based Real-Time Collaboration System," 2016, unpublished.

[28] B. Buxton, "Mediaspace Meaningspace Meetingspace," in *Media Space 20 + Years of Mediated Life*, ser. Computer Supported Cooperative Work, S. Harrison, Ed. Springer London, 2009.

[29] L. Gericke, M. Wenzel, and C. Meinel, "Asynchronous understanding of creative sessions using archived collaboration artifacts," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, May 2014, pp. 41–48.