Apparition: Intelligent Assistance for Crowdsourcing

Adiba Arif

This report is structured in following sections. In Section 2, we describe the research papers which are related to prototyping and the Apparition tool. Section 3 describes the data which is drawn from the database and Section 4 describes our method approach to do this project and in section 5 we discuss the result obtained.

2. RELATED WORK

In real time environments, prototypes are created before building an actual product, which in turn is a very time consuming task. Almost all interface designers choose to sketch the early interface design either on paper or on a whiteboard. These sketches are valuable because they allow designers to preserve thoughts, user requirements and design details before they forget them. However, there are many drawbacks to this approach. As the design evolves with time, it becomes hard to modify these designs due to lack of support for ‘design memory’. The designers cannot easily remember or search for the design annotations in future to find out why this design was made or modified, and there is lack of communication between the user and the design which are paper based.

James et al. introduced a system SILK (Sketching Interfaces Like Krazy), which allow the designers to sketch the desired interface and test it before it becomes a finished interface, with the help of an electronic pad and stylus [LM95]. It allows the designers to edit (delete, move, copy etc) the sketch and with the help of its history mechanism, it allows the designers to re-use an old portion of old designs.

Lin et al. introduced DENIM to help the designers in the early phases of a website design process [LNHL00]. In order to allow multiple visualizations of a site to the designers, zooming feature was introduced in DENIM. There are five zoom levels in DENIM:

- Site map: gives the complete view of the site along with the links.
- Storyboard: allows the designer to view all the pages together along with the links.
- Sketch: allows the designer to view the page at 100% scale.
- Overview: allows the designer to view the site more closely.

Audio recording is another important aspect of the project. The recording will be done with the help of WebRTC, along with meteor which is used to build JavaScript applications. Once the audio recording is implemented, it will be integrated with the Apparition tool.
Another interesting feature of DENIM is Run Mode, which allows the designer to test the interaction with a site without creating a prototype for it.

Seven professional designers, who later showed interest in using such systems in their work, evaluated the usefulness, ability to communicate with the designers and clients, ease of use, etc. of the tool. Based on this evaluation, the authors concluded that the tool was able to meet its main aim: to provide an ease to the designers during the early phase of designing. They also concluded that the tool required some work on fewer areas: to make DENIM to work with other tools that cover different phases of the web design cycle and to introduce more refined interfaces and GUIs.

Lasecki et al. describe a tool, APPARITION, which uses Amazon mechanical Turk to provide Wizard-of-Oz functionality for rapid prototyping [LRK+15]. This tool is the first crowd sourcing system, in which a group of workers work together to accomplish a task in real time. The users either describe their requirements in natural language or sketch their desired interface on the apparition canvas. This helps the designers to create rapid prototypes using the sketches and the verbal description. Figure 2 shows the Apparition tool.

The tool consists of a canvas, where the users and the designers can draw. It is very similar to Google Draw and is a version of SVG-Edit. Along with the canvas, there are few drawing tools like pencil, crop, text etc. that are listed on the left hand side of the Apparition tool. These drawing tools help the users to quickly create the desired UI elements. On the right hand side of the Apparition tool, there is a TODO list, which lists the elements drawn by the user but not yet converted to UI elements. Every worker can “accept” the task to show what they are working on. When a worker starts working on a task, an “In-Progress” marker is visible to the other workers in order to avoid conflicts.

The following images (3, 4 and 5) show how the user can draw an object on the canvas and play along with it i.e. add colors, rotation or transform the object etc.

Before Apparition there were many tools, such as Balsamiq, JustInMind, SILK, DENIM, d.tools, SketchWizard etc, that
Davis et al. focuses on a tool SketchWizard which allows the designers to create prototypes of pen-based UI. Apparition tools are 90% accurate in meeting user requirements. The SketchWizard tool consists of 3 separate applications:

- User’s View: a canvas is shared between both the end user interface and the designer’s interface. There are finite number of tools that are used for drawing, moving objects etc., which allow the user to interact with the object. The Wizard delivers all the operations to the end user’s view with the help of the “Commit Changes” button.

- Wizard’s Workplace: it allows the designer’s to build and execute a copy of pen based UI. Wizard’s workplace has many powerful tools in order to make fast edits, such as event stream, paste buttons, list of views and scratch pad.

- Session Player: it is used to save files which are then analysed later.

Even though crowdsourcing provides a lot of benefits, there are many challenges as well. In Micro-tasking: decomposition, coordination and quality are the major challenges. Decomposition basically involve how to divide/decompose the task into micro-task i.e. which portion of the task should be decomposed and made parallel, what the micro-tasks will ask the developers to do etc. Coordination deals with how to coordinate the crowd of works to accomplish the task and how to match the micro-tasks efficiently which suits the worker’s knowledge, state proper instructions/specifications to the worker’s. Quality deals with how to produce quality work i.e. when can redundancy provide better and produce high quality of work etc. There are the main road blocks in crowdsourcing.

There are number of ways in which crowdsourcing is being applied to software development. TopCoder, HelpMeOut, StackOverflow implement crowdsourcing. Hence, crowd sourcing is a new way to solve problems, developing software’s or websites etc.

Apparition uses MongoDB to store the data, which is a NoSQL database management system and is widely used in several projects and products such as CraigList, MTV etc. There are many advantages of MongoDB as compared to the Oracle Database. Oracle Database is an object-relational database and is one of the most common RDBMS but MongoDB is a DBMS that stores the data in a document format. The paste buttons allow designs to be placed on the canvas quickly. More work is to be done on the SketchWizard tool such that it can also be used to support the later phase of pen-based interface, as now it is only used in the initial phases of design.

Nowadays Crowdsourcing is getting very popular. It is the process of getting the work done from a group of people. Basically the work is outsourced to a crowd of workers. Example: Wikipedia, Stack overflow etc. The principle of crowdsourcing is that more number of providers/head are better than one. With the help of crowdsourcing, large number of tasks can be performed quickly by the workers. Thomas et al. explains the success theme of crowd sourcing - ‘micro-task’ [LVDH15]. Microtasking involved breaking down the work into small tasks and then sending the work to a crowd of people. With microtasking, we can expect quick results, as the work can be performed independently, quickly, with less errors and in parallel.

Apparition uses MongoDB to store the data, which is a NoSQL database management system and is widely used in several projects and products such as CraigList, MTV etc. There are many advantages of MongoDB as compared to the Oracle Database. Oracle Database is an object-relational database and is one of the most common RDBMS but MongoDB is a DBMS that stores the data in a document format.

Alexandru et al. describe why MongoDB is better that other database management systems [BRA12]. MongoDB stores data as a JSON like documents with dynamic schemas (BSON). The query language used consists in JavaScript, API calls and REST. MongoDB accepts larger data as the maximum value of its size is 16MB and it uses built-in map reduce function to collect a large amount of data. The integrity model used is BASE and its database is written in C++, which makes it cross platform database management system same as SQL database which are written in C, C++ and Java. The time taken to update, insert, delete records is faster in MongoDB as that compared with Oracle Database. In MongoDB, copying databases and deploying them from
One of the main differences between the two mentioned databases is that, the tables in Oracle Database are related to each other (one to one, one to many, or many to many). With this tables can be joined and complex queries can be formed. Databases cannot be easily copied in Oracle Database as that compared with MongoDB. So it can be concluded that, if fast, flexible database is required then MongoDB is preferred and if complex database with fix structure and relations between tables is required then Oracle database is the best option.

At times the designers and the users come across situations in which the terms used by either of them is not clear to each other. Earlier the main focus of the designer was to design non-interactive components like graphics or animations, but now as the technology has advanced so much, the designers have to deal with several other levels of interactivity such as mouse, keyboard and touch screen etc. Programmers generally use natural languages but many tools do not support the same, such as Flash. Sun et al. studied on how the programmers and the designers describe various interactive behaviours using their own language [PMK08]. This will help in building more natural programming language which in turn will make the development of interactive behavior easier for the designers.

The study was done to compare the results of the programmers and the developers i.e. where their expressions for the behaviour are same and where they differ. The study was divided into three parts. The first part focused on the ‘mouse’ interaction i.e. how the designers describe the mouse buttons and its movements. The second part focused on the response and the properties of a graphical object, such as shape, size, color etc. The last part focused on time and causality/connection. The results found that both the programmers and the developers had similar description of a particular object etc. but there were few things which both of them described in a different way. Example, if the object changes it’s size, color or any other property, then the object was represented as one entity but if the object changed it’s shape, then it was represented as two. Secondly, the designers used few set of words, appear/disappear, fade in/out, extend, expand, enlarge, gradient, mask, become larger to describe some concepts and even the behaviour of objects were expressed in different context, like ‘appear fading out’, ‘fading out slowly’, ‘moves to the right’ etc. From this study, many new language features were derived that were frequently used by the designers and few suggestions were proposed which might be used in future tools.

In a crowd-powered environment where the workers may change with time, maintaining consistency is one of the major challenges. Chorus: Mnemonic is a system that increases consistency multi-session interactions. In Chorus: Mnemonic workers can see the messages in the conversation as well as the facts posted. As per the facts which are posted, the workers can vote for or against them depending if they are relevant enough to the current topic. Depending on the votes, the facts are ranked and each time a worker re-ranks and updates the facts, the votes are recalculated. If the message is not visible in the list, it is not considered in the ranking. In order to make sure that the fact is not repeated in the list and the database, the semantic similarity is compared. Hence, Chorus: Mnemonic is a new approach to maintain long-term conversational memory in crowd-powered system as that compared to the other systems.

Nowadays, designers are highly skilled to sketch out ideas and create the prototype of the same easily. Designers can only sketch the design, but the design cannot explain the behaviors of that particular part. For behaviors, designers use storyboards or cloud to explain or we require some programming to be performed, using various different languages such as JavaScript, ActionScript, Flash, etc. The research shows that designers always start with sketching the design and later on add behavior to that using storyboards. In this paper, Brad et al. had made survey on these designers and found new interesting results that were not reported till date [MPN+08].

In the survey, most of the designers agreed that defining the behavior using different programming languages is more difficult than sketching the appearance. Also, the behavior of the designs are very complex and diverse which are defined by the designers. Designers require full programming knowledge to achieve such complex behaviors. Designers often use annotations such as arrows and text to describe the behavior because sometimes sketches and storyboards are not enough to properly convey the behavior. The results shows that 76% of designers find very difficult to communicate the behavior of the design with other designers in the team. The more important thing that was reported is that the interactive behavior is generated during the process of making the design i.e. designer does not have the clear idea of the final behavior of the design at the start.

To perform this study, authors used contextual inquiry methodology with 13 designers, in which the designers are observed and interviewed while they are working natural environment. They took around 90 minutes of each designer for this process. After applying contextual inquiry methodology
ogy, the authors conducted a generalized survey with other designers. The results of the study confirmed that 86% of designers reported that defining behavior is more difficult than appearance and 76% agreed that communication with other designers in the team is big problem. This is a very important information that encourages and helps to produce tools to solve this problem.

As many functionalities are being added to the Apparition tool, audio recording is one of them. The audio recording is done with the help of WebRTC, which is an open framework for the web.

2.1 WebRTC
WebRTC stands for Web Real-Time Communication which is an API that enables Real Time Communications in the browser like audio chat, video chat or peer to peer applications. It also offers the web application developers to write real-time multimedia applications on the web, without any plugins or any other downloads or installs. In order to provide high quality communication on the web, WebRTC includes many fundamental components such as audio, network and video. These components can be accessed through JavaScript API. The following figure (7) shows WebRTC architecture:

The architecture is mainly divided into two layers:

- The Browser: browser developers will be interested in the WebRTC C++ API along with audio capture, video capture and network I/O. The WebRTC C++ API mainly consists of voice engine, video engine and the transport.

- The Web: the web application developers are more drawn towards the Web API as it is mainly used by the third party developers in order to develop web based video chat applications.

WebRTC has many advantages, such as it enables communication in the browser adding to the fact that it is free and provides high quality. It also helps in building the strength of the web browser. Along with the above mentioned advantages, Google does not charge royalties for WebRTC.

Along with WebRTC, the meteor is also used in order to perform audio recording. Meteor is an open source JavaScript platform used for developing web and mobile applications. It allows you to develop in one language i.e. JavaScript, in all environments.

The following figure (8) explains Meteor architecture:

Meteor framework is a client-server model i.e. there is a meteor client and a meteor server. Meteor server is based on NodeJS runtime environment and is connected to the Mongo DB and uses Mongo to query the database. Whereas the meteor client is connected to it local database. The client sends request to the server and through this method call, the client and the server communicate.

There are many advantages of meteor:

- One language: meteor allows you to develop in one language, JavaScript, in all environments (application server, web browser, and mobile device).
- Data on the wire: meteor server sends data, not HTML, and the client renders it.
- Database Everywhere: meteor uses the database consistently.
- Full stack reactivity: meteor allows your UI to reflect the true state of the world with minimum development effort.
- Simplicity equals productivity: meteor API is easy.
- Latency compensation: zero latency.

3. RECORDING
Recording is a process that helps to capture data or then convert the data to a specific format that can be stored.

In order to start the recording, the user will run the application by passing the three parameters i.e. the workerId, session and role in the browser. If one of the parameters is not specified, the recording will not be initiated. Once all the parameters are mentioned, two buttons are displayed on the screen:

- start recording
When the user/client clicks on the 'start recording' button, a window appears on the screen which asks for the access to the microphone and the camera. Once that is approved by the user/client, the audio/video recording begin using RecordRTC (WebRTC). Once the recording is done, the user/client can stop the recording by clicking on the 'stop recording' button. As soon as the 'stop recording' button is clicked, the browser sends the recorded data to the server. The server then saves this data to a file and updates it’s database. Figure 9 is a flow diagram of how the recording is done.

Figure 10 is a still from the video recording of the user.

3.1 Meteor-streams vs RecordRTC

RecordRTC is a JavaScript based WebRTC framework which is used for real time communication of audio/video on web browsers. It allows multiple recording sessions and also supports browser/WebRTC API method called getUserMedia. RecordRTC is supported by the following browsers:

- Firefox

- Google Chrome
- Opera
- Android
- Microsoft Edge

With RecordRTC, we can communicate between

- client to clients
- server to clients
- client to server
- server to servers

Meteor-streams are used for peer to peer connection using JavaScript. A Common stream is created between the two peers for real time communication of the audio/video on the web browser. Meteor-stream allows you to create as many streams as we want in our application. It supports peer to peer, client to server and server to client communication, as that similar to RecordRTC. But unlike RecordRTC, meteor-stream is only supported in Google Chrome.

HomanWebRTC application uses streams for real time communication. In homanWebRTC application, there is a connection between the two clients using one stream i.e. client (client A) to client (client B) connection. So whenever client (client A) is speaking (audio), then that client (client A) is not actually sending the data to the receiver (client B) but the receiver (client B) is listening from that stream. So a pipe (stream) is created between both the clients (client A and B) on each end. There is no way to read the data from that one stream using RTC peer connection object, which maintains that stream. So basically, meteor-streams are mainly used for broadcasting or audio/video conferencing.

So instead of using meteor-streams, RecordRTC framework is used to record the audio. It takes into consideration all the three parameters from the URL and stores the mapping in the mongodb database.

4. DATA

4.1 Animation Data

Animation is a process of creating an artificial motion to an object or a photograph for better visualization. Animations can bring life to the objects.

In Apparition, animations can be represented by moving the image/object from one coordinate to another, by change the angle of rotation or by performing various types of transforms on the image/object. The apparition data is stored in the form of JSON files in MongoDB. The following images (figure 11, 12, 13 and 14) depict a forest and various animation features which were added by the user.

The below listed animations which were performed on the objects drawn by the user in figure (11, 12, 13 and 14):
Figure 11: Animation step 1

Figure 12: Animation step 2

Figure 13: Animation step 3

Figure 14: Animation step 4

Figure 15: Work flow between the User and the worker

- Move
- Rotate
- Resize
- Create/Delete

Figure 15 shows how the users and the workers work on a collaborative canvas:

- **User/Client**: performs various actions like, drags the player towards the ball, moves the ball from one side to another.

- **Workers**: there are ‘n’ number of workers behind the screen that work together to create the prototypes. They animate the ball and provide response to the ball (by moving human objects).

The data of the Apparition tool is saved in MongoDB in the form of a JSON file. Python code was written in order to extract the data from the mongoDB. The following images (16, 17, 18 and 19) shows how the data is represented in every collection of the meteor database.

"Changelog" is the most important collection in order to extract the features of an object. The data for "changelog" is described below:

- type: what action is being performed
- action: what type of action is performed
Table 1: Collections

<table>
<thead>
<tr>
<th>Collections</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 changelog</td>
<td>type, action, session, worker, fill, width, height, rotation angle, x-y coordinates</td>
</tr>
<tr>
<td>2 colors</td>
<td>maincolor, session, id</td>
</tr>
<tr>
<td>3 todos</td>
<td>user, session, id</td>
</tr>
<tr>
<td>4 drawcontent</td>
<td>transform, layerIndex, opacity, fill, type, id, x-y coordinates, session</td>
</tr>
<tr>
<td>5 mixed</td>
<td>workerid, _id, members, timestamp, id</td>
</tr>
<tr>
<td>6 changestack</td>
<td>_id, change, timestamp, id</td>
</tr>
</tbody>
</table>

Table 1 describes each collection briefly:

Table 2 briefly describes each attribute used in the collection:

5. METHODS

The system described here can be found at https://github.com/cromaLab/Apparition/tree/aa-refactor. The software used in the project include:

- Meteor
- Node.js
- MongoDb
- Python (anaconda)
- Apparition
- python (version 2.7)
- pymongo
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 id</td>
<td>unique id assigned to every user</td>
</tr>
<tr>
<td>2 session</td>
<td>name given to a particular session</td>
</tr>
<tr>
<td>3 worker</td>
<td>The person who is drawing on the canvas</td>
</tr>
<tr>
<td>4 rotation angle</td>
<td>rotation angle of an image</td>
</tr>
<tr>
<td>5 x-y coordinates</td>
<td>x and y coordinate of the object</td>
</tr>
<tr>
<td>6 maincolor</td>
<td>color filled in the object</td>
</tr>
<tr>
<td>7 transform</td>
<td>transformation occurred on the object</td>
</tr>
<tr>
<td>8 type</td>
<td>shape of the object (circle, square)</td>
</tr>
<tr>
<td>9 fill</td>
<td>color filled in the object</td>
</tr>
<tr>
<td>10 action</td>
<td>what type of action is performed (move)</td>
</tr>
<tr>
<td>11 layerIndex</td>
<td>unique number assigned to the object</td>
</tr>
<tr>
<td>12 timestamp</td>
<td>encoded information of when a certain event occurred</td>
</tr>
</tbody>
</table>

Table 2: Collections

5.1 Errors while installing Apparition

- mrt: command not found

  Then install npm using the following command: "sudo npm install -g meteorite" and then execute "sudo mrt".

  If "sudo mrt" still throws an error then try using "sudo meteorite".

- error: Package names can only contain lowercase ASCII alphanumerics, dash, dot, or colon, not "M".

  Rename the package folder to lower case i.e. "mv -r package folder lower_case".

- Cannot find module 'cheerio'

  Run this command from the root folder: "npm install cheerio" and then "sudo mrt"

  
  https://github.com/cheeriojs/cheerio

6. CONCLUSION

Through the study of various tools and comparing them with Apparition, we concluded that Apparition tool has many additional functionalities like refined interface, set of tools are provided to the user for easy drawing and the user can describe his/her requirements using natural language. Apart from these, recording is another feature which has been implemented but will be integrated with the tool in the next phase. When the recording function will be integrated, the user audio will be recorded and stored on the server side, which in turn will be used by the workers in order to get the exact requirement of the user/client. Apart from recording, data analysis was also done on the animation data. From that gathered data we saw that every step the user/client is performing, it is being recorded in the database. This data, in the future, can be helpful for creating prediction models and tools.

7. FUTURE WORK

- To achieve significant improvement in the tool by adding new functionalities such as motion and images etc.

- The data analysis has been initiated by detecting some attributes that can be used further to improve the Apparition tool (prediction).

8. REFERENCES


[LKR+15] Walter S Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P Bigham, and Michael S Bernstein. Apparition: Crowdsourced user interfaces that come to life as you sketch them.


