FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Visual Prototyping Tool for Android and iOS

Augusto José Trindade Soares

U.PORTO

FEUP **FACULDADE DE ENGENHARIA**
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Teresa Galvão, PhD

July 19, 2013

Visual Prototyping Tool for Android and iOS

Augusto José Trindade Soares

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Ana Cristina Ramada Paiva, Prof. Auxiliar Faculdade de Engenharia da Universidade do Porto
External Examiner: João Paulo Jorge Pereira, Prof. Auxiliar Instituto Superior de Engenharia do Porto
Supervisor: Maria Teresa Galvão Dias, Prof. Auxiliar Faculdade de Engenharia da Universidade do Porto

_____

July 19, 2013

Abstract

The ubiquity of smartphones in our lives has generated a booming market for mobile cell phone applications. This rapid expansion has created an increasing need for the development of functional prototypes for these applications that mimic, as accurately as possible, the behavior of the final product, in order to be able to perform usability tests.

This poses a problem, as the general lack of technical skills of the usability experts, who usually design these prototypes, conflicts with the need of writing code to produce prototypes that have complex functionalities. On top of this, the usually used tools to perform this kind of tasks often possess, themselves, major usability flaws.

Our proposed solution for this problem is twofold. On the one hand, we will develop a tool that allows HCI experts to visually develop complex prototypes with advanced back end functionality that minimizes the need of writing code. On the other hand, we will try to make this tool as usable as possible, by performing a four-stage study of the most popular approaches to several design environment problems that allows us to create a new design methodology for these kinds of cases.

To sum up, we propose to create a tool that enables HCI specialists with minimal technical skills to quickly, efficiently and cheaply develop high quality functional mobile prototypes.

Resumo

A ubiquidade dos *smartphones* nas nossas vidas gerou um mercado em grande expansão para aplicações de telefones móveis. Esta rápida expansão criou uma necessidade crescente de desenvolver protótipos funcionais para estas aplicações que imitem, de forma tão exata quanto possível, o comportamento do produto final, de forma a ser possível realizar testes de usabilidade.

Isto põe um problema, uma vez que a carência generalizada de competências técnicas dos peritos em usabilidade, que usualmente concebem estes protótipos, entra em conflito com a necessidade de escrever código para produzir protótipos com funcionalidades complexas. Para além disto, as ferramentas normalmente utilizadas para desempenhar este tipo de tarefas possuem frequentemente, elas próprias, grandes falhas de usabilidade.

A solução que propomos para este problema tem duas vertentes. Por um lado, desenvolveremos uma ferramenta que permita aos especialistas de Interação Humano Computador desenvolver visualmente protótipos complexos com funcionalidades de *back end* e que minimize a necessidade de escrever código. Por outro lado, tentaremos tornar esta ferramenta tão usável quanto possível, realizando um estudo em quatro etapas das abordagens mais populares a vários problemas dos ambientes de desenho que nos permita criar uma nova metodologia de *design* para casos deste tipo.

Em suma, propomo-nos criar uma ferramenta que permita aos especialistas de IHC com competências técnicas mínimas desenvolver de forma rápida, eficiente e económica protótipos móveis funcionais de elevada qualidade.

"Another happy landing"

Obi-Wan Kenobi

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

AICOS   Assistive Information and COmmunication Solutions
BPMN   Business Process Modeling Notation
FUSAMI   Fraunhofer USAge Mining
HCI   Human Computer Interaction
HTML   Hypertext Markup Language
IDE   Integrated Development Environment
SMI   SensoMotoric Instruments
UI   User Interface
UML   Unified Modeling Language

Chapter 1

Introduction

In today's world, the smartphone has become an indispensable object for many of us. We use it for everything, from work, to leisure, all the way through entertainment and management of our social lives [ORMR12]. This near omnipresence of the object itself, that is even expected to grow over the next few years [Cis13], has sparked the birth and exponential growth of a market for what makes it so flexible and useful: mobile apps.

This market growth, that currently produces revenue in the order of trillions of dollars [Val11], has generated a corresponding boom in mobile software development companies. These companies are required to work at breakneck speed, while producing top quality products, in order to separate themselves from the competition.

The aforementioned need for quality products encompasses providing the end user with highly usable app interfaces. This implies that, during the development process, high quality prototypes must be created, in order to allow the performance of the necessary usability testing. For these tests to be as significant as possible, the prototypes should, to the highest possible degree, mimic the behavior and functionality of the final application.

These quality and function requirements pose some problems when coupled with the previously discussed need for fast development. On the one hand, the task of visually designing mockups of this nature is, by itself, not devoid of some complexity, due to the intrinsic difficulty of the process and to the general usability issues of the tools usually used. On the other hand, the Human Computer Interaction experts, who usually design these prototypes, don't have the technical skills required for writing the necessary code when back end function is a must.

How then, can these problems be solved?

We propose a two-way solution. Firstly, we propose to develop a visual design tool that allows the non-technical HCI experts to, not only design the prototypes, but also find and invoke a set of predefined back end services, with minimal to no need of programming knowledge. Secondly, we shall aim to optimize this tool's design, in order to make its usage as simple and efficient as possible.

The achievement of these goals will be pursued first by exploring and studying different existing prototyping tools, in order to catalog the different ways in which some of the main design problems are solved. We shall then perform a usability study, that will allow us to find out which are the most effective ones, from the pool of solutions presented, and apply them to our final tool [TA08].

Chapter 2 of this document shows an analisys of the state of the art in the areas of relevance to our work. Chapter 3 features a description of the solution we propose to our problem. In chapter 4 we have a description of the work performed. Chapter 5 features the analysis of the test results and the final chapter in this document shows the conclusions and a summary of future work.

Chapter 2

State of the Art

The present chapter describes the current state of the art in the main areas related to this work. An analysis of the most relevant tools that exist in the area of visual mobile prototyping is made, namely regarding the way in which several design issues are addressed. We show how the different tools can be harvested of individual components that can then be tested, in order to produce a superior final tool. The need for this process arises from the fact that, as far as we were able to determine, no tool currently available can achieve all that we propose to.

In addition we analyse several knowledge areas in the field of Usability research as well as some of the technologies that are used for that end.

2.1 Design Patterns

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them [CH92]. In other words, HCI studies how people and computer systems interact, be it by sight, sound, touch or any other means. In the most traditional context for this interaction, using a physicall computer terminal, designing software has come a long way. We can today identify (and particularly in the field of web software) certain design methods by wich specific and recurrent interaction problems can be solved. These are called Design Patterns [VHJG95]. Design Patterns play a fundamental role in modern software development as they allow us to simply, efficiently and cleanly provide the end user with the most seamless interface experience. This characteristic is of the utmost importance when you try to develop a web tool such as the one we have(2.4.1).

Through the study and application of Design Patterns we can create a tool that is not only pleasent to use but, perhaps more importantly, can even guide the user in the acomplishment of his goals. [DKvD11]

2.2 Usability Testing

Another area of HCI, Usability, is in the context of software development, *"a quality attribute that assesses how easy user interfaces are to use."* [Nie03]. It is defined by things such as how easy it is for users to complete a task the first time they use the system, how quickly can they regain proficiency with it after a long time without using it or simply by how pleasant it is to use.

Usability is then important because, no matter how good your product is, if the users can't use it properly, they won't use it at all [Nie03]. This makes improving usability in to a key aspect of software development. The best way to improve usability is to perform usability tests.

One of the most effective ways of improving a software's usability is to perform tests on different versions of the system in order to conclude which of the changes introduced were beneficial to overall usability. [GL85, WBH88]

Usability tests are an evaluation process through which usability metrics can be recorded. They are specific measurements that allow us to quantify the quality of the users experience. These metrics include how long a user takes to perform a task, if the user performed it successfully or how much of the interface was remembered since the last time. The analysis of these metrics lets us identify and correct usability problems making our software much easier and simpler to use increasing user satisfaction [TA08].

In general, usability metrics are split into three categories [Hor06]:

Measures of effectiveness: in the context of usability testing, effectiveness pertains to the degree of completeness and correctness to which a given task is performed. In other words, was it done all the way and was it done right. These measures can, according to [Hor06], be further categorized into sub-groups. They are binary task completion, accuracy, recall, completeness and quality of outcome.

Binary task completion verifies if users successfully complete tasks or not. Examples of this measures include number or percentage of tasks performed correctly, number of tasks that users failed to finish within a given time [WCW01] or number of tasks where a user gave up [DCC01].

Measuring Accuracy allows the quantification of the number of errors either occurred during a tasks completion or present in its final solution. This could encompass the number of errors in data entry [MFJ01] or you could have hints given by the experiment conductor subtract from an overall accuracy score [NBGR01].

Recall pertains to the extent to which users are able to retain information about the system having used it before. It generally refers to visual components of the system's user interface.

Measures of efficiency: although efficiency may, on the surface, seem to be very closely related to effectiveness they are in fact quite distinct. Using a common adage, if effectiveness is "doing the right thing" then efficiency is "doing the thing right" i.e., the extent to which resources (whatever they might be) are used for an intended purpose. The most commonly

studied measure of efficiency is time. The most prevalent measure of time is time on task or how long it takes a user to conclude a given task. It is however possible to discriminate this broad measure of time into more specific units. Examples of this could be mean time between actions [GHR99], time spent reading dialogue boxes [TH00], time spent in certain parts of the display [Bur00] or even time elapsed until a certain event occurs such as the first key stroke [ME99].

Although they are generally regarded as of great importance, [Dil01] argues that the different actors involved in the software development process may consider time measures to have different degrees of importance thus creating the possibility of its overestimation as a usability metric.

Measures of satisfaction: are a matter of opinion and are therefore intrinsically more difficult to both obtain and analyze. These metrics are generally obtained through either questionnaires or interviews. These can be difficult to produce since most studies in the field usually provide information on the satisfaction metrics whose measurement was intended but not on the questions administered in order to do it.

For this work focus will be on task success and error rate as measures of effectiveness, total time on task and average time between actions as measures of efficiency and for a measure of satisfaction we will use the prevalent System Usability Scale questionnaire [Bro96].

This questionnaire was found by Tullis & Stetson to be one of the most reliable availeable and to be extremely comprehensive despite its simplicity and diminutive size at only 10 questions [TS04].

Another very important aspect of planning a usability test is defining how many users should be necessary to produce a valid study. There are two main established points of view within the usability community. One of them supports that in a usability study five participants are sufficient to validate the tests and that such a small number of subjects will be enough to uncover about 80% of a systems usability flaws [Lew94, NL93, Vir92]. This was based on the findings by Nielsen and Landauer that a single user would be able to, on average, find 31% of the total usability flaws in a system [NL93]. That number, compounded over five users would equate to the previously mentioned figure of 80%.

Another more recent school of thought on the subject defends that the "magic number 5" [TA08] as it is called, will not provide the results predicted by Nielsen [SS01, WC01](Molich et al., 1998). Spool and Schroeder [SS01] found in their research that after the first five participants, only 35% of the complete set of usability flaws had been identified. This study suffered however, due to the enormous scope of the websites in which it was conducted [TA08] and the fact the open ended tasks used meant that it was possible that some parts of the system were effectively not tested [Sau10].

Both camps however build their assumptions on the same thing. The actual percentages and numbers of users can be derived through equation (2.1) where i is the percentage of total issues detected in a usability test, p is the probability of each issue being discovered by a user in each trial and n is the total number of subjects used for the study.

$$i = 1 - (1 - p)^n \tag{2.1}$$

Jeff Sauro [Sau10] defends that the discussion should not be around i or n but rather on the estimation of p as once you have that, you can stipulate a desirable i and then calculate the necessary number of users to achieve it. It is however difficult to get an estimation for p as you can't really know how likely a user is to encounter a problem. Sauro adds that "in fact, you often don't even know if there is a problem—if you did you'd fix it!"

We can nonetheless establish that this probability will (hopefully) be higher for systems that are in early development and lower for systems that are in use with many users. Findings by Virzi [Vir90] and Nielsen & Landauer [NL93] put the average value of p for systems in development at 0.32 and 0.31 respectively. If we define our target mathiti as 80% we get $n = 5$ in both cases.

Tullis and Albert [TA08] also suggest a compromise in which 5 users should be necessary for each "significantly different class of user"(p. 119). They note that in their experience over the years it was rare for new issues of significance to be discovered past the fifth usability trial. They support the validity of this idea as long as the evaluation scope is clearly limited and the user audience is clearly defined and represented.

2.3 Eye Tracking

"Eye tracking is a technique whereby an individual's eye movements are measured so that the researcher knows both where a person is looking at any given time and the sequence in which the person's eyes are shifting from one location to another" [PB05]. This technique allows HCI researchers to better analyze and understand the ways in which users process visual interface environments.

As a discipline, eye tracking study dates as far back as the late nineteenth century [JK03, RP89, Jav79]. Although many advancements and improvements were achieved throughout the early and mid-twentieth century [JMCS05, MM58, MT62, CC73] it was only in the nineteen seventies that one of the greatest improvements to eye tracking occurred when the advent of the minicomputer allowed researchers to quickly and effectively analyze collected data. Prior to this scientists could "spend days processing data that took only minutes to collect" [M$^+$75](pp. 331-332).

It wasn't however until the nineteen eighties and the proliferation of the personal computer that scientists began to investigate how eye tracking technology could be employed in human-computer interaction research [JK03]. Initially some of the main applications of the technology

were in the study of interactions for disabled users [HWJM$^+$89] and the in the field of aviation, namely in flight simulators [TF84].

In more recent times eye tracking in human-computer interaction has served not only as a means of studying the usability of computer interfaces in new information sharing technologies such as email or videoconferencing [ECM$^+$98, CBD02] but also as a means to serve as an interaction mechanism itself [Ver99, ZMI99]. In this work, we focus on the former application of eye tracking technology.

As a human-computer interaction study tool, eye tracking software allows the recording of three main metrics: fixations, saccades and scanpaths.

Fixations are moments when the eye assumes a relatively stationary position while the user takes in or interprets information [PB05]. A fixation is usually quantified in terms of its duration and the area of the interface in which it occurs. For instance, an overall larger number of fixations can indicate less efficient search in the interface while the number of fixations in a specific area can be used as a gage of that area's perceived importance or noticeability to the user [PBP05]. Longer fixation durations usually translate to a difficulty in extracting information [PB05]. In table D.63 we show and interpret the main fixation derived metrics used in this work.

Table 2.1: Main eye tracking metrics pertaining to fixations. Adapted from Poole & Ball(2005) [PB05]

Eye-Movement metric	What it Measures	Reference
Number of overall fixations	More overall fixations indicate less efficient search (perhaps due to sub-optimal layout of the interface).	Goldberg & Kotval(1998) [GK98]
Fixations per area of interest	More fixations on a particular area indicate that it is more noticeable, or more important, to the viewer than other areas.	Poole et al.(2004) [PBP05]
Fixation duration	A longer fixation duration indicates difficulty in extracting information, or it means that the object is more engaging in some way.	Just & Carpenter(1976) [JC76]
Fixation spatial density	Fixations concentrated in a small area indicate focussed and efficient searching. Evenly spread fixations reflect widespread and inefficient search.	Cowen et al.(2002) [CBD02]

"Saccades are quick eye movements occurring between fixations" [PB05](p. 4). They can alternatively be seen as a user's gaze travel path between fixation points and they are usually

measured in terms of their length (in display units travelled) or amplitude (user eye rotation angle) which are directly correlated with amplitude generally preferred. A high number of saccades is indicative of more search being necessary and therefore of a less efficient interface. Larger saccade amplitudes correlate to more meaningful visual cues as the user's attention is drawn from a distance [PB05].

Table 2.2: Main eye tracking metrics pertaining to saccades and scanpaths. Adapted from Poole & Ball(2005) [PB05]

Eye-Movement metric	What it Measures	Reference
Number of saccades	More saccades indicate more searching.	Goldberg & Kotval(1998) [GK98]
Saccade amplitude	Larger saccades indicate more meaningful cues, as attention is drawn from a distance.	Goldberg et al.(2002) [GSL$^+$02]
Scanpath spatial density	Smaller spatial density indicates more direct search.	Cowen et al.(2002) [CBD02]
Spatial coverage calculated with convex hull area	Scanpath length plus convex hull area define scanning on a localized or larger area.	Goldberg & Kotval(1998) [GK98]

Scanpaths are a visual representation of a complete saccade-fixation-saccade sequence [PB05] (figure 2.1 [ML12]). An optimal scanpath is considered to have straight lines to desired targets with short fixation duration at those targets [GK98]. A scan path with a smaller spatial density (the area the scan path's "shape" takes) is indicative of more direct search [JK03]. This can be best observed by examinig a scanpath's convex hull. This is obtained by tracing the smallest polygon that contains all fixation points and has no concavities, i.e. every line segment connecting two vertices either coincides with one of the polygon's edges or is itself completely contained within it. The main metric we analyzed regarding scanpaths was its spatial density as seen in table 2.2.

The eye tracking station in Fraunhofer's Hardware Lab utilized SMI's Begaze Eye Tracking Analysis Software. This software provided the means to easily setup the experiments and collect all data related to eye tracking and user actions. The software allowed the exporting of experience data in raw text based formats to be analysed with third party software but also provided a graphical environment that showed experiment replays with the eye tracking and event data graphically superimposed.
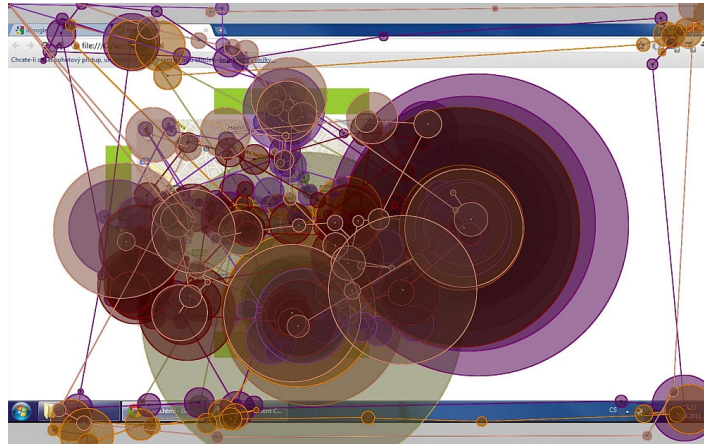
Figure 2.1: An example of a scanpath. Circles represent fixations with diameter denoting duration. Straight lines represent saccades between fixations. [ML12]

2.4 Visual Prototyping

In the market of visual prototyping there are dozens of available tools. We have analyzed several of them and came away with a few that stood out from the lot. We now present an analysis of these few, pertaining to how they approach different design challenges.

2.4.1 Codiqa

Codiqa [Cod11] is one of the most popular web prototyping tools. It generates HTML5, JQuery coded prototypes that can be executed directly on smartphones. This tool presents a list of addable widgets on the left side of the screen. This list has a minimal amount of available widgets, with each of them representing a relatively broad spectrum. More detailed configuration is left for a later editing phase, which starts when an existing widget is selected, as a popup containing detailed configuration options for that component (figure 2.2) . This tool does not support the addition of any kind of back office functionality. Codiqa also allows the user to export and edit the generated code.

2.4.2 Fluid UI

Fluid UI [Sof11] is another web-based tool that produces HTML5 prototypes, that can be executed directly on your smartphone. Unlike Codiqa, the design menu on Fluid UI contains a large amount of very specific widgets, with few configuration options, presented on a scrollable list (figure 2.3), also on the left side of the screen. The few configurations are available through a tool bar that pops up whenever you select a component on the design canvas. In terms of actions, Fluid UI supports only basic transactions between different prototype screens.
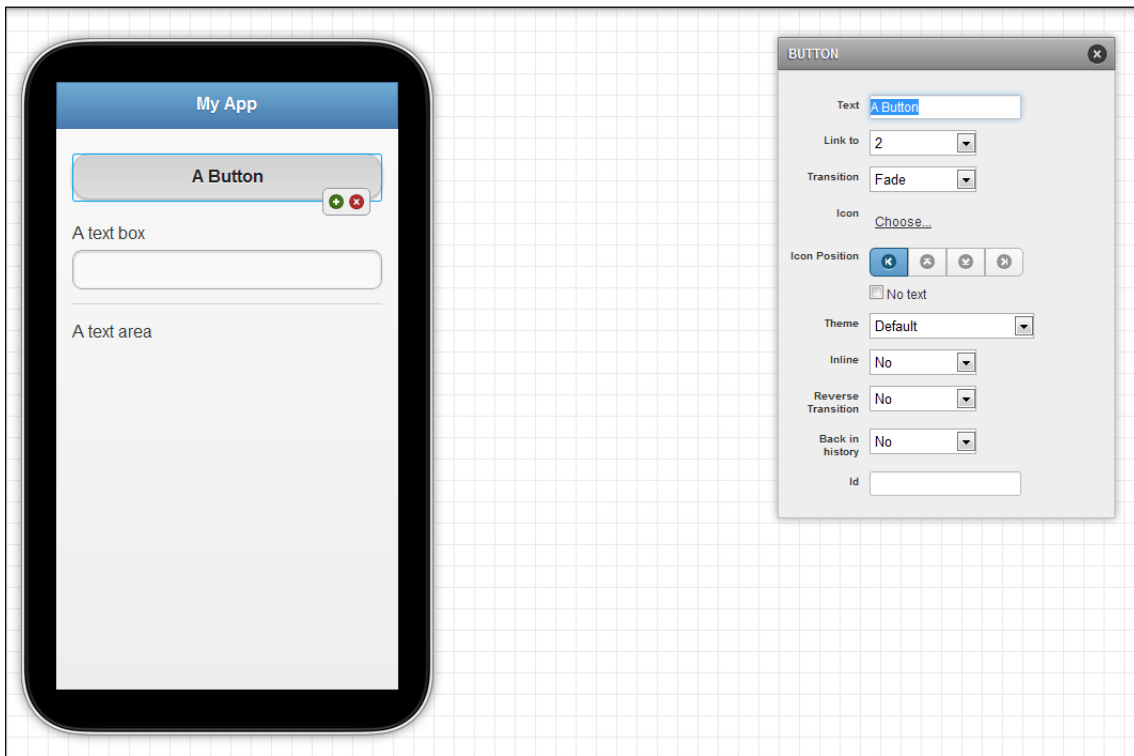
Figure 2.2: Codiqa's configuration interface

2.4.3 Maquetta

Maquetta [IF] is a web tool, developed by the Dojo foundation, utilizing HTML5 and the Dojo Framework itself. Maquetta has a tab based design menu, with each tab containing a set of widgets grouped according to basic type as seen in 2.4. The configuration of added widgets is made through a fixed menu, on the right side of the screen, that presents the configuration options of whichever element is selected. Functionality wise, Maquetta defines its app screens as states that can be transitioned between each other when a specific trigger is detected (for instance a specific button is pressed).

Figure 2.3: Fluid UI's cluttered design interface with just part of the widgets visible
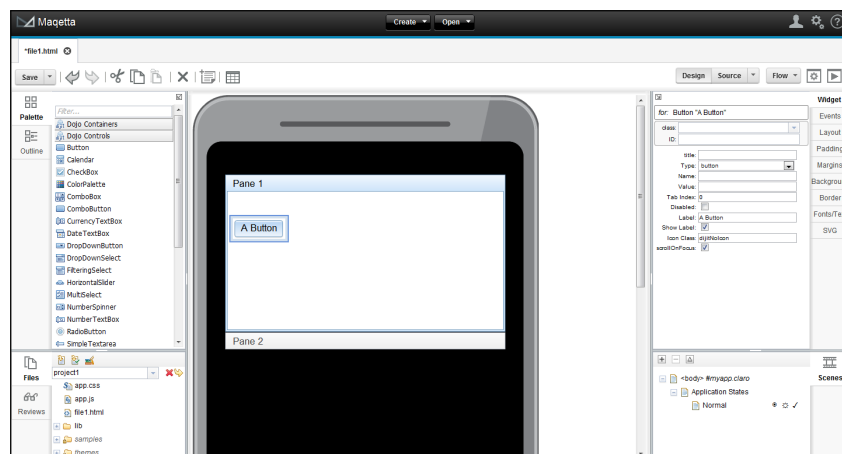
Figure 2.4: Maqetta screenshot showing the tabbed design menu on the left with "Dojo Controls" tab selected

2.4.4 Tigzi

Tigzi [Exa10] is the last selected web prototyping tool. In terms of designing the prototypes, it provides much the same solutions the other tools do. That being said, Tigzi stands out from the others in that it provides much more complex action editing than the rest. It provides an expandable menu on the bottom of the screen that allows the user to choose a screen component (such as a button or text box, for instance), define a trigger and then choose from a list of actions the one to perform when the trigger is set off. Despite this robust option, Tigzi has by far the most complex interface, with some severe usability issues (2.5).
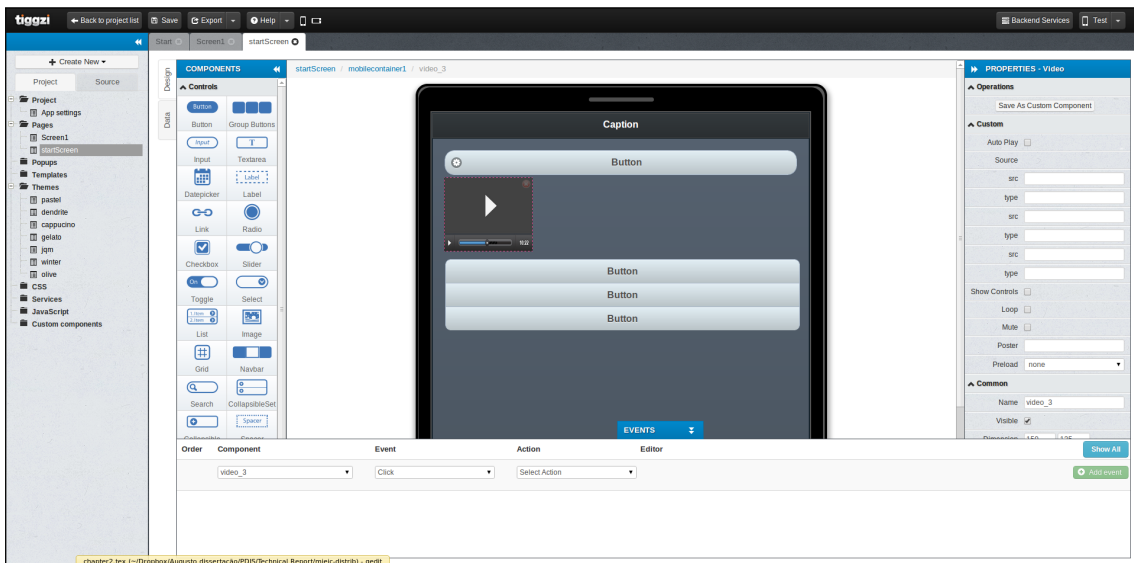
Figure 2.5: Tiggzi's complex interface

2.4.5 Other tools

Some other tools have also been studied, namely the visual editors available in the most popular desktop programming IDE's, such as Eclipse, NetBeans or Microsoft Visual Studio. However, we soon came to the realization that these tools employ much the same design paradigms of the web based tools we previously studied.

2.5 Related Work:FUSAMI

Our work is to be developed by extending FUSAMI, or Fraunhofer Usage Mining. FUSAMI is a web based tool developed by Fraunhofer AICOS, that allows, not only the design of mobile prototypes, but also the collection and analysis of usage data from test sessions with real users of those same prototypes. Specifically, after a researcher has designed an application prototype, FUSAMI allows the gathering of usage data, such as event frequency, event number or action

areas. This information is automatically analyzed and presented visually, as heat maps superimposed on screenshots of the real application. This data proves to be invaluable in the usability study of different design approaches during mobile development.

FUSAMI's editor, in its current state visible in figure 2.6, is currently only in a proof of concept stage of its development not really presenting a real world ready option. It possesses only a very short amount of available widgets and supports only the addition of basic transitions between the different prototype screens. The mobile prototypes design in the FUSAMI editor can be tested, both in browser and live, on a smartphone, through the scanning of a QR Code.

Both FUSAMI and the prototypes it generates are coded using HTML5 and the Dojo Toolkit, a JavaScript Framework. This allows the prototypes to be executed in both Android and iOS devices.
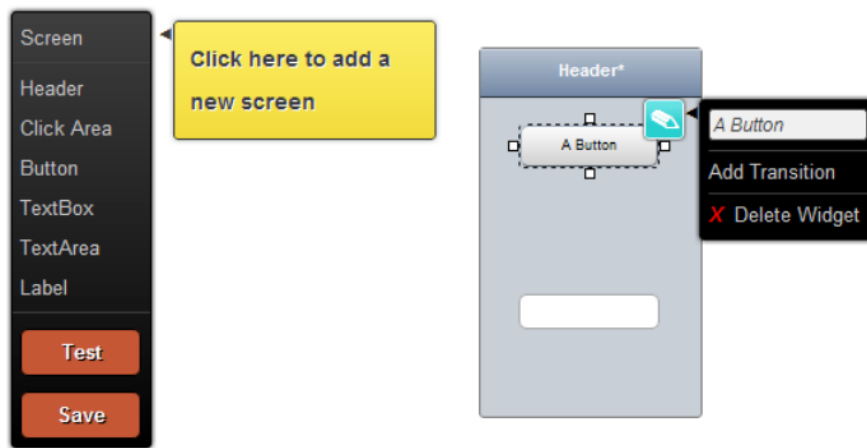
Figure 2.6: An example of FUSAMI's current interface

Chapter 3

Proposed Methodology

In this section we will describe how we aim to perform the proposed work. In other words, we will specify how the work was divided in parts and how each part is organized. As previously explained, our problem is twofold: providing a design tool that is as usable as possible for non-technical staff, while at the same time, providing them with the capability to add back end services with minimal programming knowledge needed.

As far as the back end services are concerned, we will provide our tools' users with a visual abstraction that represents the services available and accurately portraits the control flow of the mobile prototypes. This visual abstraction will be based on the BPMN design syntax. BPMN, or Business Process Modeling Notation, is a graphical notation system that depicts the steps to follow in a business process. It was developed by the Business Process Management Initiative and it is maintained by the Object Management Group [Gro05]. BPMN's focus on the business process [WvdAD+06] allied to Fraunhofer's business centered organization, motivated this choice over more software oriented languages such as UML.

As for the usability refinement of the editor tool, after the analysis of the several tools available, we have come to the conclusion that the process of designing a complex functional prototype can be characterized as the iterative performance of three main tasks:

Design - this task comprises how the prototype is designed, in terms of its screens and the different widgets present on each screen. The most important UI component in this phase is the design menu that allows the creation of screens and the addition of widgets to these screens.

Configuration - in this task advanced configuration is performed on an already existing screen or widget. The most important UI element in this process is usually a Properties menu of the component we wish to edit.

Transition - the way the designer is able to add functionality to the prototype, by creating dynamic transitions between the screens with complex behaviors based on customizable triggers.

Taking this division into account, the development of our tool will be split into five phases.

3.1 Phase One

In the first phase Fraunhofer's design and HCI team will be given surveys concerning the best design options for a visual design tool such as the one we're developing. This survey will poll the designers and HCI experts on how important for the overall quality of the tool are certain features. It will also contain questions aimed at evaluating the quality and effectiveness of several solutions identified to solve the design tasks explained above. The solutions that will be featured in the inquiry, the ones identified in the analysis of the tools detailed in sections 2.4.1 through 2.4.4, are:

For the Design task:

- the design menu will show a concise list of the most common widgets used, with the option to expand the list and show additional widgets.

- a scrollable list of every available widget.

- a tabbed (or collapsible sections) menu, with each tab containing a subset of widgets, grouped by closeness in type.

For the Configuration task:

- a fixed, dedicated section of the editing screen that will always automatically show the configuration options of any widget or screen that is selected.

- a pop up menu containing these options when the widget is selected.

- a toolbar that provides access to different configuration options when an element is selected.

For the Transition task:

- different "function" widgets that can be dragged from the menu and dropped on existing transactions like normal widgets

- an "add function" option on the right click context menu of every transaction.

- the function enabling and editing options on each transactions properties menu, much like the ones from widgets.

After this survey is analyzed a focus group consisting of some of Fraunhofer's designers, developers and HCI experts will be formed in order to analyze the data and eventually find additional viable approaches for the different problems. The solutions that receive sufficiently positive feedback will carry on to the next phases.

3.2 Phases Two through Four

The three intermediate phases will be conceptually identical but each will have a different focus. Specifically, each of them will focus on one of the tasks identified above. Each phase will consist of a round of prototype development, followed by a round of usability testing.

Our goal in the development round is to develop in the vicinity of three prototypes. Any less could compromise the studies validity and any more would critically increase the development overhead [Nie94]. This number is, however, subject to the results of the survey and the focus group. If, for instance, everyone in a group of skilled and experienced designers and developers agrees that a solution for a given problem is overwhelmingly better than all others presented, it makes sense to use that solution and move on to the next phase.

Each of the prototypes developed will use the same approach to the two tasks that are not the focus of the phase. The approach used will be FUSAMI's default one. This will allow the elimination of a bias created by task learning from successive rounds ensuring that the results measure the difference between the features that are actually being tested. This featured being tested will consist of a different and unique approach to the task that is the focus of the development phase in each prototype.

A round of usability testing of the prototypes will follow each of these rounds. Each prototype will be tested by a sample group, who shall be given a list of specific tasks to perform. Along with the tasks, each participant will also be given a usability survey to take upon completion. This process is further illustrated in figure 3.1.

The resulting metrics from these tests, along with the survey data, will be gathered and analyzed. The results will then be used in order to determine which approach constitutes the best solution to the task that was the focus of each of the first three rounds.

Second Phase

Design task

Configuration task

Transition task

Prototype A1:
Solution X1

Prototype A2
Solution X2

Prototype A3
Solution X3

FUSAMI
default

FUSAMI
default

Third Phase

Design task

Configuration task

Transition task

FUSAMI
default

Prototype B1:
Solution Y1

Prototype B2
Solution Y2

FUSAMI
default

Fourth Phase

Design task

Configuration task

Transition task

FUSAMI
default

FUSAMI
default

Prototype C1:
Solution Z1

Prototype C2
Solution Z2

Prototype C3
Solution Z3

Figure 3.1: An illustration of the intermediate development phases, showing an example of the different prototype composition and the selection of the best options.

3.3   Phase Five

The best performing solutions from each intermediate round will carry over to a fifth and final phase, in which they will be assembled into a final prototype (figure 3.2).

This final prototype will undergo a double round of usability tests, similar to the ones on the early phases. These two rounds of tests will serve to gather, not only usability, but also learnability data and will therefore have a wash out period between them of one to two weeks, depending on the available time.

The expected outcome of this final testing is to validate the developed work, by having the final prototype outperform each of the intermediate ones. It is our conviction that we can thus develop the most usable and efficient tool possible.

Final Prototype

Design task	Configuration task	Transition task
Prototype A1: Solution X1	Prototype B2 Solution Y2	Prototype C2 Solution Z2
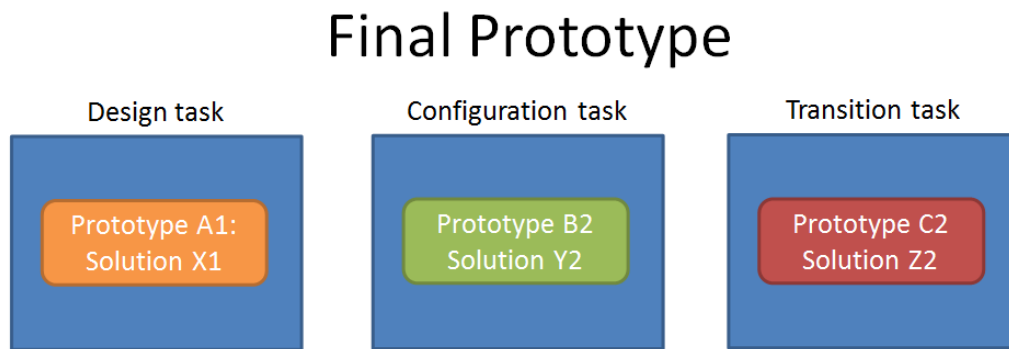
Figure 3.2: The final prototype will feature the selected approach from each of the intermediate phases.

Proposed Methodology

Chapter 4

Proposed Solution

In this section, we will make a detailed accounting of all work phases in this project. It will cover activities ranging from initial tool development to final data analysis and interpretation.

4.1 Initial Development

During the first stage of the project, the main objective was to get familiarized with the application. At the same time, some functional changes were introduced. In terms of mobile prototype editor interface the most important changes were the addition of a few new widgets, the ability to resize both of an element's dimensions simultaneously, keyboard shortcuts for element deletion, selection, clearance, and saving and, finally, the ability to change new configuration options: some element's color, screen orientation and screen resolution.

These changes were not included in the tool version that was analysed in the focus group.

In figure 4.1 we can see a graphic overview of the research work. We can see that the research started with the analysis of the state of the art for the most important areas related to this work. The results from this study were submitted to a survey whose results were then used to conduct a focus group. The result of the focus group consisted on a decision about which interaction options should be implemented and tested. As shown in the figure, two prototypes for Placement and three prototypes for Configuration were selected, implemented and tested. The best prototype from each of the groups was then selected an integrated in the final prototype which was then submitted to a further round of tests. The Transition interface selected during the focus group could not be coded in time for testing and was left as a concept to develop and implement in the future.

4.2 Surveys

A survey was created that aimed to obtain some data from HCI and design professionals in order to enable the further development of FUSAMI in tune with the preferences of the professionals who will ultimately use it.
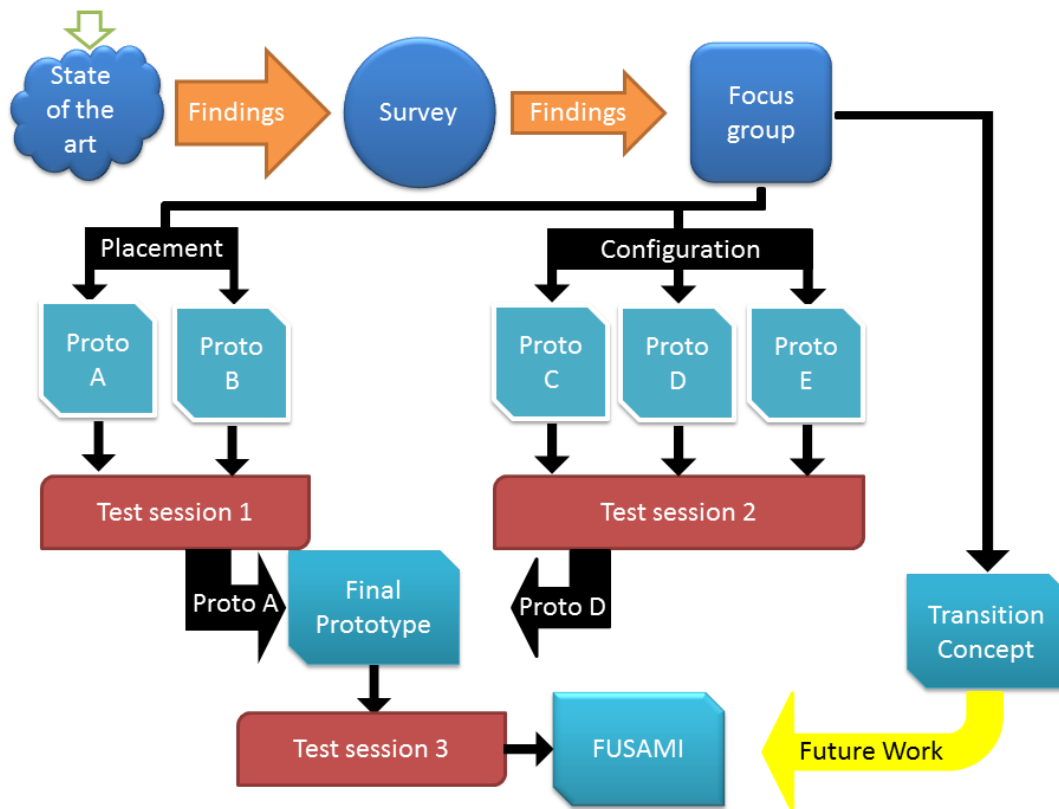
Figure 4.1: A schematic representation of the work flow during research.

This survey (Appendix A) was administered online and was divulged through HCI and mobile development groups in LinkedIn, in HCI and web design boards on reddit.com and through a mailing list for HCI professionals. Answers were collected between March 25 and April 17.

A total of 29 respondents answered, out of which 9 were HCI experts, 8 were designers, 5 were programmers and 7 indicated other occupations (1 HCI designer, 2 HCI students, 1 HCI researcher, 1 product manager and 2 User Experience developers)(figure 4.2).

To begin the survey, users were presented with a short introduction giving context to both the survey and the tool whose development it was a part of. Then, users had to answer 15 questions.

The questions on the survey were divided into three groups. The first group consisted of two questions (1 and 3) that were meant to gather data related to the interface paradigms we were studying. In both cases the respondents were given a description of the situation, were presented with the options identified from the study made in section 3.1 and were asked to select their favorite option for each case. The answers to these questions are further analysed in sections 4.3.3 and 4.3.4.

The second group of questions had four questions (questions 2, 4, 7 and 8) approaching different ways in which FUSAMI could be further refined from a user interface standpoint. Question 2 asked the respondents to choose their five favorite elements from a list of widgets based on
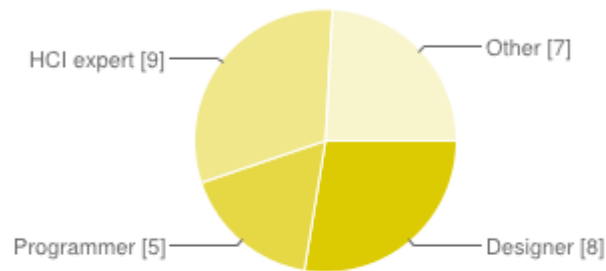
Figure 4.2: Respondents' ocupations.

their perceived value for mobile prototyping. Question 4 required the users to rank several design environment features based on how important they thought each one would be in the context of the tool being discussed. The answer options for these two questions were derived once again from the analysis of similar tools available (section 3.1).Question 7 asked the users opinion on the importance of prototype control flow based on its usefulness for prototyping. Question 8 was a non-required open ended question asking for other features users identified that were not covered in the other questions. The most common suggestions were the possibility of the end user being the one creating new custom widgets and the possibility to change the styles of the target device on fly (for instance instantly switching a prototypes style between Android and iOS).
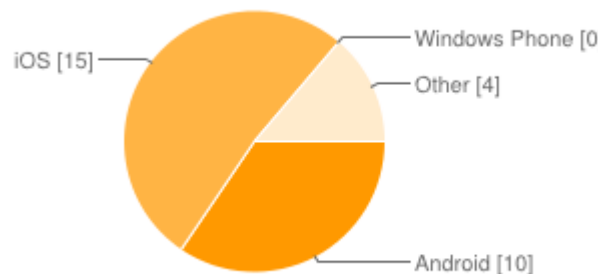
Figure 4.3: Respondents' main development platform.

The third group of questions (questions 5, 6 and 9 through 15) were intended to provide a picture of the respondents' professional roles, their work preferences, the tools and resources they used for development, and the respective platforms. With these questions we were able to find, for instance, that nearly all of the respondents developed for either Android or iOS (figure 4.3), that 72% followed the platform style guides when designing, and that more than half of them created different prototypes for different platforms when tasked with prototyping a cross platform app. This reinforced the notion that the ability to quickly toggle target system styles on the fly and within the editor would be an extremely useful feature. We also found that most users do both low

23
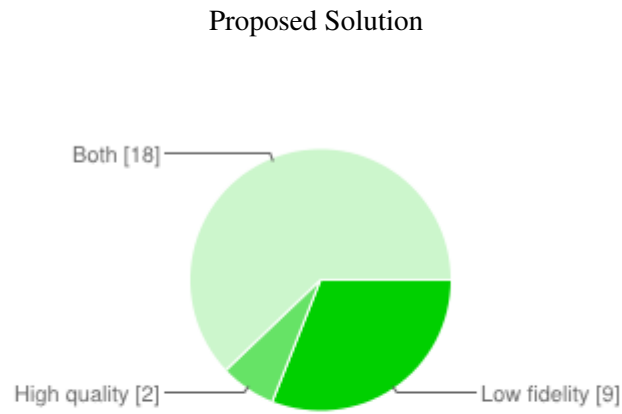
Figure 4.4: Respondents' usage of low fidelity(paper) or high quality mockups.

fidelity (paper) and high quality mockups (figure 4.4), that 83% of them used interactive mockups and that most of them were well experienced with visual prototyping tools (figure 4.5) but not with Javascript (figure 4.6). We also obtained a ranking of some of the most popular tools for mockup and prototype design (figure 4.7).
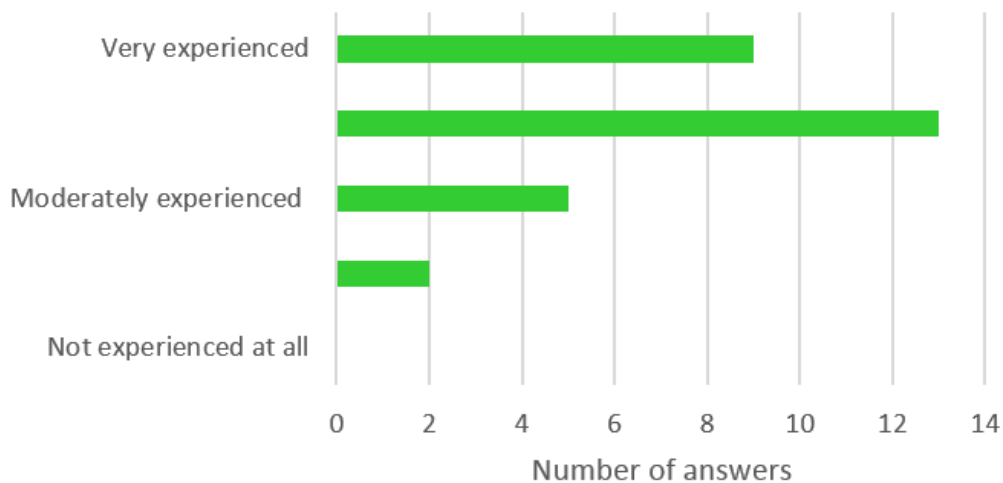
Figure 4.5: Respondent's reported experience level with visual prototyping tools.

The results from the second and third groups of questions were used to further refine and plan the future development of FUSAMI. Questions from the first group were used to set up some of the discussions in the focus group.
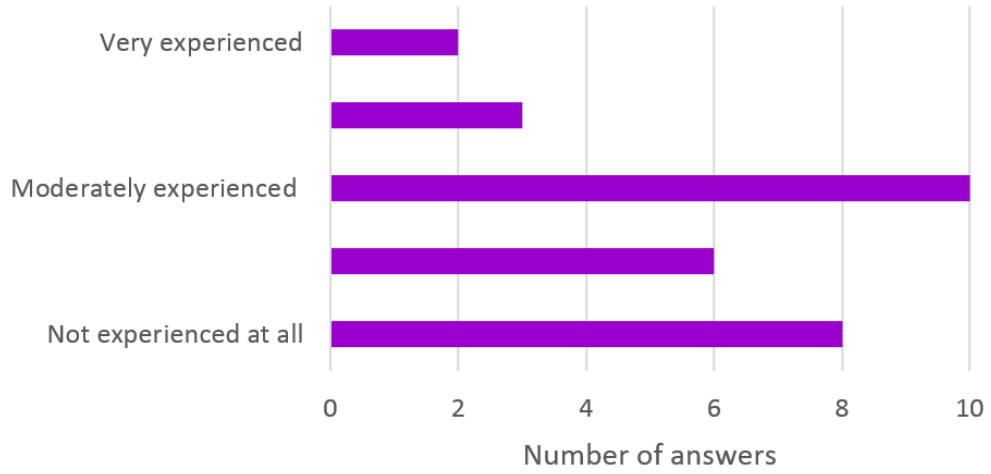
Figure 4.6: Respondent's reported experience level with Javascript.
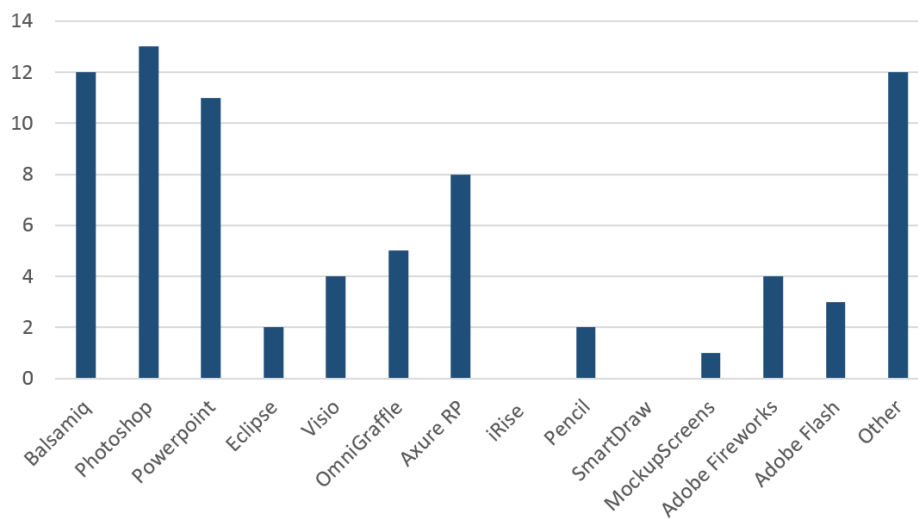
Figure 4.7: Number of respondents using some of the most popular prototyping tools.

4.3   Focus Group

The focus group was intended to explore both the results from the surveys and the initial study on the several available tools, in order to decide which features to include in the test prototypes as well as some information for future development on other areas of FUSAMI.

Participants were invited from within Fraunhofer's design and HCI departments. Six of them attended the focus group, two designers and four HCI specialists, and they were requested to do a brief recon of the tool prior to the session. The focus group was held in Fraunhofer's Living Lab on Thursday the 18th of April 2013. Due to an inability to include a note taker, the session was video recorded for posterior analysis.

The focus group session was divided into five stages each focused on a different aspect.

The first stage consisted of a usability analysis of the tool. The second stage comprised a discussion of which widgets' development should be prioritized. Phase three, four and five were concerned with the analysis of the Placement, Configuration and Transition interfaces respectively. Each of this phases is described in detail in the following sections in this order.

4.3.1   First Stage - Usability Analysis

The first stage was related to the brief study of FUSAMI that participants were asked to perform. Each participant was asked to name the most severe usability flaws he or she had detected in their use. The most prevalent flaw reported was the inability to drag and drop elements from the menus to the work area and the necessity to click the elements in order to select them and then click the desired location in the work area in order to place them. Drag and drop is the most common paradigm in this sort of tool (especially considering that a significant part of design and HCI professionals use computers running Mac OS where the drag and drop paradigm is virtually universal). It was agreed by all participants that it presented a better solution but that ideally both ways of interaction should be supported. To further the idea that drag and drop was necessary a large part of the errors committed during the trials were a result of its absence when users attempted to add new elements.

Another of the topics raised by the participants was the support for multiple screen resolutions and orientations in the editor. At the time the focus group took place the prototype editor supported only a single screen resolution and orientation. All participants agreed that screen orientation (portrait vs. landscape) should be an editable property of each screen. As for screen resolution, although everyone agreed that it was a necessary addition, a consensus could not be reached as how to best implement it. On the one hand it was discussed if it should be a screen specific property or a project wide setting (with a strong inclination towards the latter). On the other it was questioned if this should be editable from within the project at all or if it should be something you defined upon project creation, i.e. once you defined the target device for your project (and therefore the appropriate resolution) those settings would be set for the entire life of the project.

A few other issues were also mentioned. The ability to change the prototype's overall style was mentioned again, as well as the possibility of editing the elements' styles individually. Participants

also thought that keyboard shortcuts were necessary especially for the actions of deleting elements and clearing the current selection. Mentioned along with the keyboard shortcuts was the necessity for an undo and redo mechanism along with copy/cut/paste functionality. It was decided that the implementation of the undo and redo mechanism, should also trigger the elimination of the "Save" button and the creation of an auto-save feature activated after every change.

The final issues raised were the ability to duplicate entire screens, with widgets and configurations, and to select and permanently group together multiple widgets.

4.3.2 Second Stage - Widget Analysis

The second part of the focus group covered the issue of what widgets should be made available to users. The objective was to establish a sort of ranking that defined which widgets should be implemented and the implementation priority of each one. The participants were confronted with the results from the surveys and asked to provide a comment relative to those results and possibly make suggestions for other elements not covered in the surveys. Concerning the existing elements, the general consensus was that Grid widgets should be higher (second to checkboxes) and that switches, while important, should not be given the same importance as checkboxes. The rest of the order should remain as is but with some degree of liberty.
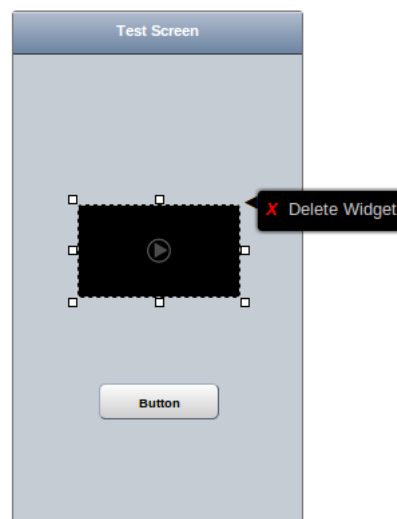
Figure 4.8: An example of a video placeholder used to simulate element placement.

Two main ideas were developed. First, it would be very important to have a widget that somehow simulated media elements (pictures and videos) even if it was just some kind of visual

placeholder that "reserved" the space for the element on the screen (figure 4.8). The second idea was that it would be extremely useful to have a group of "Basic Shape" widgets (geometrical figures) that allowed users to design or draw their own custom elements. The issue was again raised that it should be possible to save these custom elements for further use.

4.3.3 Third Stage - Placement Interface Analysis

The third stage of the focus group was the first to approach subjects directly involved in the testing phase. Namely, this stage concerned the analysis of different paradigms for the selection and addition of widgets to an app mockup. Participants were shown examples of the three possibilities identified in section 3.1. These examples, as well as those on the next section, were merely static screenshots of the paradigms being studied applied in real life applications. First they were shown an image of Gmail's expandable menu shown on figure 4.9a. Although not directly applied for the intended purpose in a similar tool, this example served to illustrate the paradigm of a menu with a small subset of the most common or used options with the possibility to expand and show every option. The second possibility showed a permanently expanded scroll list containing every possible widget (figure 4.9b). The final option presented a tabbed or accordion menu, pictured in figure 4.10 with widgets grouped by categories contained in tabs that could be expanded and collapsed at will.

(a) GMail's expandable menu.   (b) A fixed fully expanded scroll list.

Figure 4.9: Two of the alternatives presented

Participants validated the findings from the surveys (figure 4.11): a full scrollable menu was not a very good idea as it would be easy to get lost in it as the number of options grew with the tool's development and the tabbed interface was thought to be better than simple expandable one although it would be worth it to test them both. These two options were therefore carried to the first test phase with the accordion menu as prototype A and the expandable menu as prototype B (figure 4.1 in section 4.2).

Proposed Solution
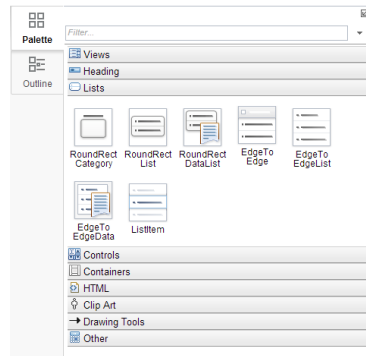
Figure 4.10: Widgets on an accordion menu.

Although no further interface possibilities were presented in this phase, an idea came up for future development. The widgets presented in the default tab of the accordion menu or in the always visible section of the expandable menu should be customizable. It was not decided if this customization should be manual, which would incur some initial configuration work but would pay off quicker or automatic which would remove the need for customization but would only reach full effectiveness on the long run.
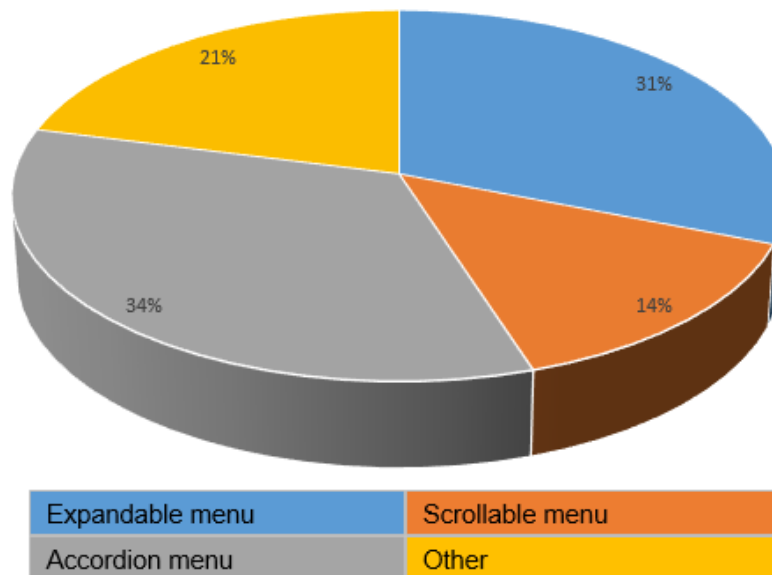
Figure 4.11: Placement interface options' survey results.

4.3.4   Fourth Stage - Configuration Interface Analysis

The session's fourth stage focused on options for displaying and changing configuration settings of already existing elements. Participants were again shown examples of the most common solutions identified in section 3.1. The first possibility in this screen showed a predetermined section of the work area being used to display a configuration menu as shown on figure 4.12.
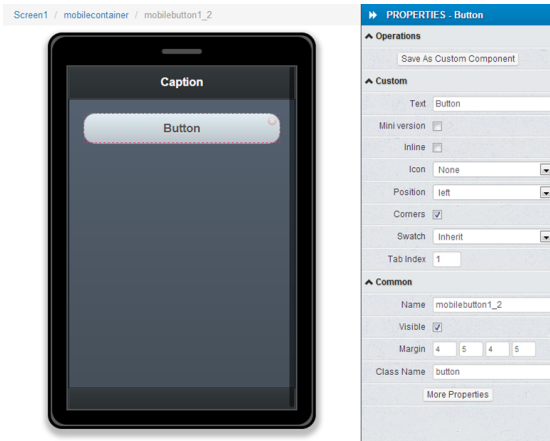
Figure 4.12: Configuration options on fixed side menu.

The second and third possibilities were visually very similar with the same configuration menu appearing right next to the element being configured, anchored on its upper right corner. They had however a key difference. The second option showed the menu immediately upon element selection while the third first displayed a small button on the same upper right corner and only after clicking the button displayed the menu (figure 4.13).
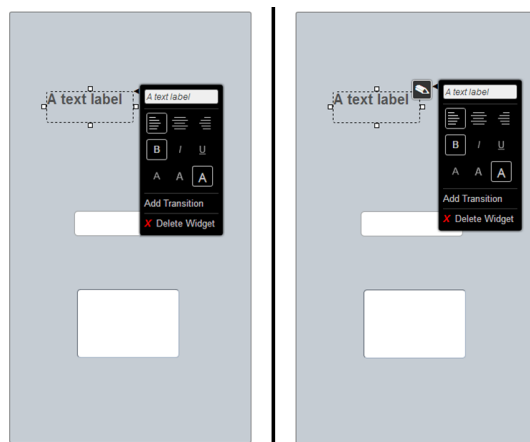
Figure 4.13: Pop-up menu with and without intermidiate button.

The fourth possibility displayed the configuration options on tool bar located next to element that showed up on element selection as seen in figure 4.14.

Proposed Solution

Figure 4.14: Configuration on a toolbar.

In this instance, the participants did not completely validate survey results(figure 4.15). They did, like survey respondents, agree with the fixed screen section and immediate pop-up solutions. Unlike them, however, they thought the pop-up with the small activation button warranted some more testing because it provided a much less visually intrusive solution when wishing to manipulate widgets without necessarily editing their properties. On the other hand, the bar interface was found to be too visually cumbersome and confusing and was therefore abandoned as an option.
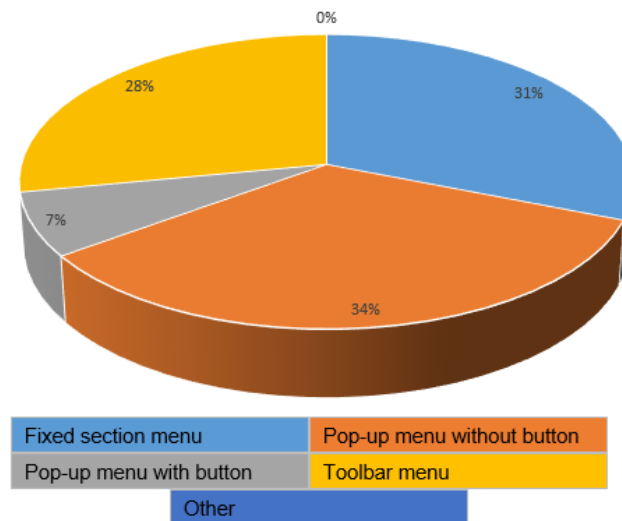
Figure 4.15: Configuration interface options' survey results.

We then had three options to implement and test. The pop-up menu with activation button was designated as prototype C, the pop-up menu without activation button as prototype D and the screen fixed section menu as prototype E (figure 4.1 in section 4.2).

31

4.3.5 Fifth Stage - Transition Interface Analysis

The final stage of the focus group session concerned the configuration of complex transitions between screens. Right of the bat some confusion arose among the participants about the exact meaning of a transition. The participants who had more design oriented roles at first interpreted transitions as being animations of screens or elements (much like a Microsoft PowerPoint transition between slides for instance). In the scope of the FUSAMI mobile prototype editor, although it can encompass those animations, a transition has a broader meaning. The term Transition refers to the passage of prototype focus from one app screen to another on a given trigger event and to all other actions (such as the mentioned animations) that may be associated with said passage.

The discussion then shifted to how to actually solve the problem. Participants were shown a mockup of an envisioned alternative. The context was that of creating a transition that could conditionally go to different places based on another element on the mockup. A practical example could be that when a button was pressed on screen A there could be a transition to either screen B or screen C depending on whether a check box on screen A was ticked or not.

The alternative consisted of a graphical element, akin to a widget, that could be dragged on top of an existing transition (figure 4.16). After the element was placed the user would be prompted to select the second destination screen. Once that was done the transitions context menu would have the option to select the element on which the condition would be based.
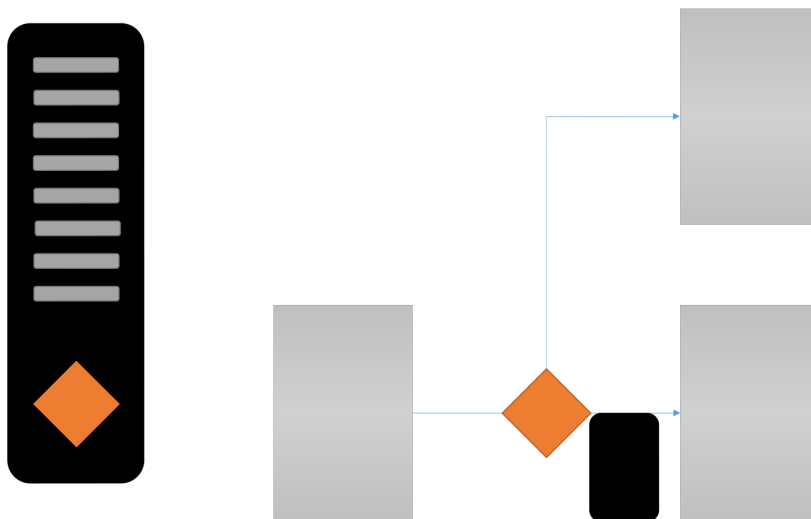
Figure 4.16: One of the static mock ups created for the Transition interface.

This scenario was rejected for two reasons. First, the participants found this scenario was unlikely to occur as this situation is not common in mobile prototyping. It was then suggested by one of the participants that a much more relevant scenario would be that of associating different

events to different kinds of transitions. The second reason was that the interface presented was considered to be too confusing and cumbersome to use.

We then abandoned the concept of a split transition, focused on the new scenario and tried to come up with a better approach.

Right from the start, the idea to explore was inspired by having a menu appearing upon selecting a transition that would allow its configuration. Specifically, one would be able to change the transitions origin (possibly a widget or an entire screen) and its destination screen. It would also be possible to select the transition trigger. The trigger could fire on events such as tapping and element, holding it or swiping left or right. Finally, the user would be able to select the transition effect such as a fade or a side swipe. The understanding was then that this would be the best option for the scenario but it would also work if, in the future, this functionality was expanded to include the aforementioned transition or even other backend services such as database interaction or the calling of web services.

After the focus group an interface mockup was constructed in FUSAMI according to the proposed specifications (figure 4.17). Although this solution was proposed, agreed upon and even designed it was not carried to testing as there was not sufficient time to produce a full function prototype.
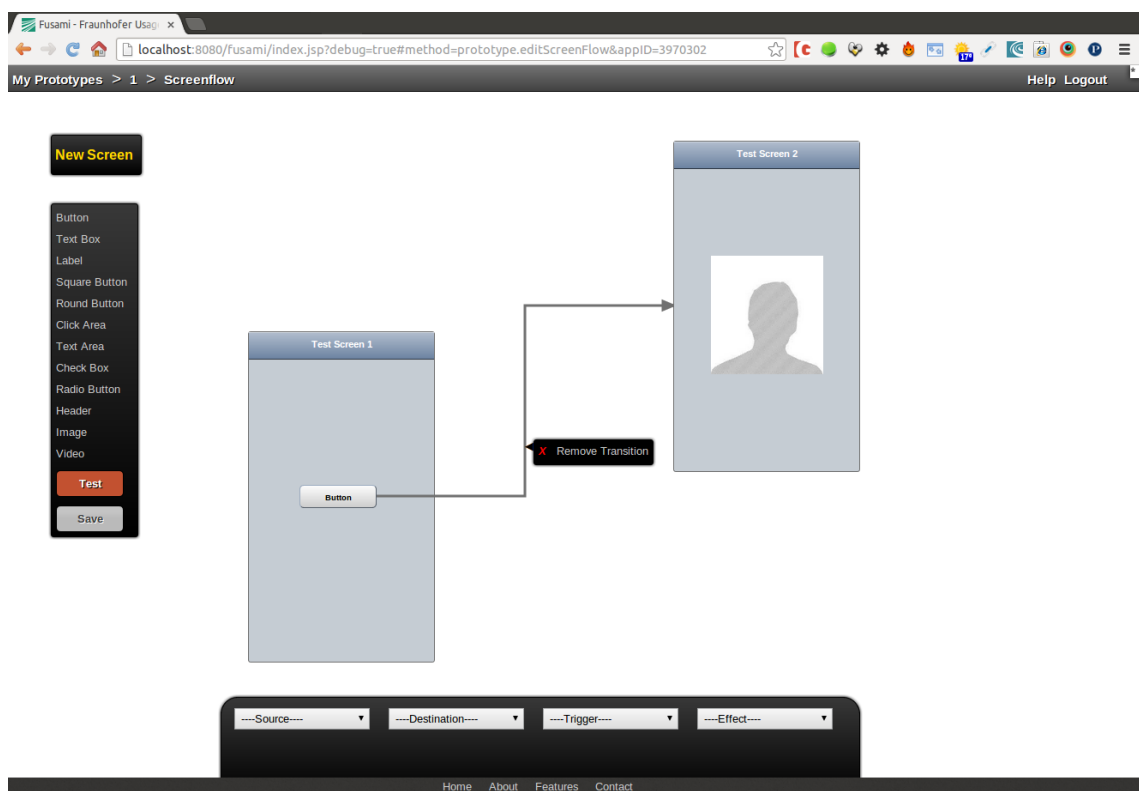
Figure 4.17: The FUSAMI transition interface mockup

4.4 Implementation and Testing of Functional Prototypes

Users for the tests were selected as a sample of convenience. The invitation was made to the entire Fraunhofer development team. Eleven users accepted to participate. The group was made up of designers, HCI experts, software developers and student researchers in these areas providing a reasonably good representation of the target demographic. In total these 11 people represented 19% of Fraunhofer's R&D group.

Although 11 appplied only nine valid data sets resulted. The first user's tests were invalidated due to a previously undetected error in the prototypes while the second was abroad during the second round of tests. In either case there was neither the time nor the opportunity to repeat or perform the necessary tests. These two users did however both fill the SUS questionnaire and answer the preference questions when availeable.

The prototypes selected during the focus group were implemented as functioning, fully featured versions of the existing editor. They were developed on top of the FUSAMI editor using the same technologies. Specifically the prototypes were coded using HTML5 with Javascript for interaction logic using the Dojo framework.

Primary testing was divided in two sessions each focusing on a different aspect of the interface (figure 4.1 in section 4.2.

Session one comprised tests of the placement interface. Following up on survey and focus group results two prototypes were created. The first prototype is based on an accordion menu and will be henceforth designated prototype A. The accordion menu is constituted by a series of sections or "tabs". It has a "Favorites" section that is active by default and four other sections that can be toggled on (figure 4.18) with the widgets available in each tab organized by similarity of type or function. Toggling a section on will automatically toggle off the previously selected section.

The second prototype, prototype B, has a widget menu with a main area visible by default containing some more commonly used widgets and a secondary area containing all of the remaining elements available whose visibility can be toggled on or off (figure 4.19).

The default tab on prototype A and the main area in prototype B contain the exact same number and order of widgets, i.e. the widgets directly available with no menu navigation are identical in both prototypes.

In this first test session, users were given two different tasks and asked to perform both tasks using both prototypes adding up to a total of four tasks on the session. Users performed tasks 1 and 2 on one of the prototypes and then performed them again on the other. This caused a familiarity bias because, for the second iteration, users were already familiar with both the general interface and, possibly more importantly, the tasks. To counteract this effect the order by which each prototype was used was alternated between each user.

In the first task users were presented with a blank new app and then they were required to create three new mobile screens and then place, in each of the new screens, a "Button", a "Text Box" and a "Label" widget. These were in fact all the elements that were readily accessible to the
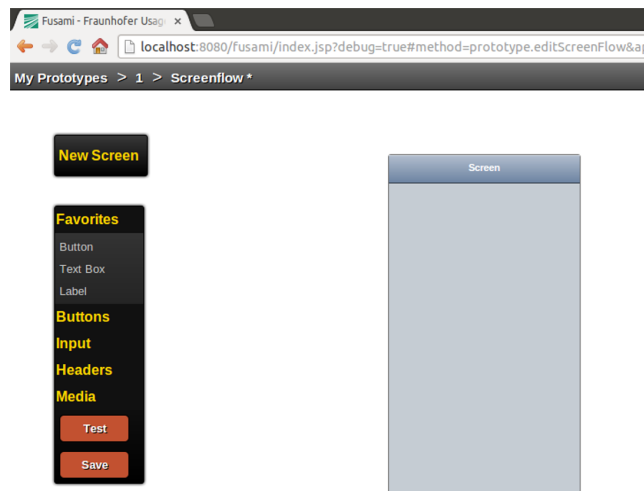
Figure 4.18: FUSAMI Accordion Menu.
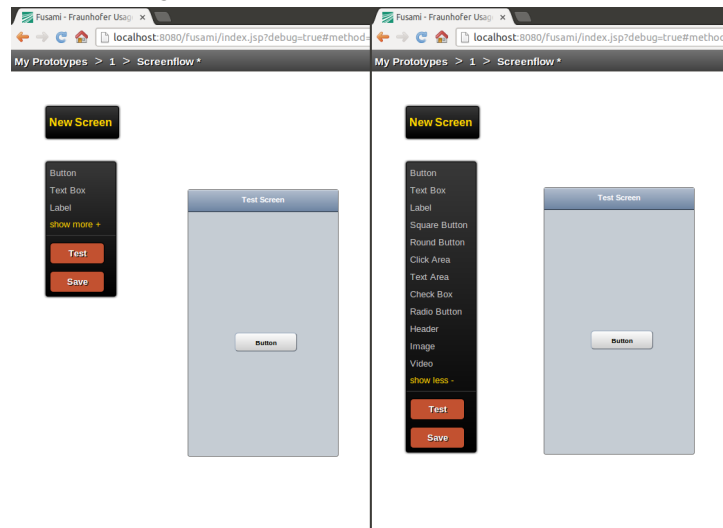
Figure 4.19: FUSAMI Exapandable Menu.

user without navigation. The two prototypes were therefore, within the scope of this task, virtually identical. The objective was to establish a baseline between the prototypes and determine if the visual presentation differences themselves exerted a significant difference when the content was identical.

The second task was somewhat the opposite of the first. In this task users were again presented with a blank app and asked to create three new screens and place an identical set of three elements in each screen. The set of elements to place was however different. This time users should place a "Square Button", a "Check Box" and a "Video" widget in each screen. The choice of these particular elements had two reasons. First they were all elements that could not be accessed in either prototype without navigating the menu. Second, in the particular case of prototype A, the elements were, not only all located in different tabs, but one of them was in a non-contiguous tab

meaning that in order to place a video users should ignore the "Headers" tab altogether and go directly to the "Media" tab (Appendix B, section B.1).

In the second session, the focus was on the Configuration interface. Again, in accordance with the findings of the surveys and focus group, three prototypes were created for this session. The first one, prototype C adopts the incumbent approach of a small button that appears next to an element's top right corner upon its selection. When the button is clicked, it triggers a menu that appeared on the immediate right of this button and contains property editing options for the selected element (figure 4.20).
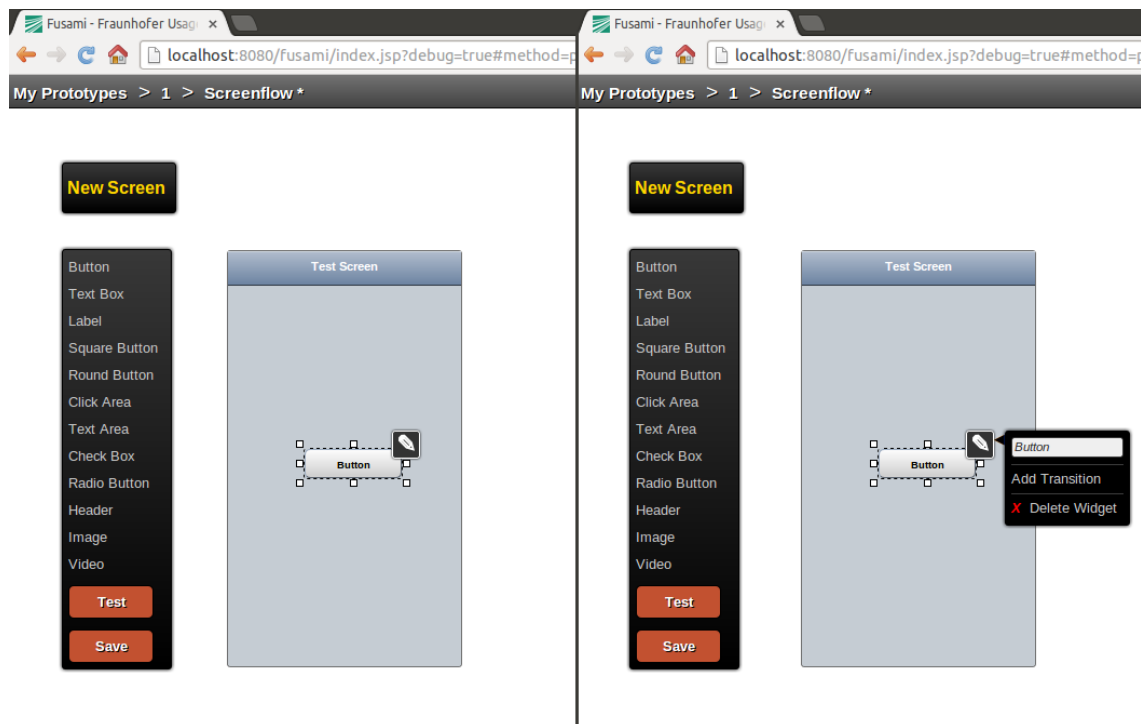
Figure 4.20: FUSAMI Pop-up menu with button before and after activation

The second prototype for this section, designated prototype D, is somewhat similar to the previous one with one major difference. In prototype C, when an element is selected it automatically triggers the appearance of the editing menu next to the element without the intermediate steps related to the small button (figure 4.21).

Prototype E, the final in this session, adopts a slightly different approach. Like in prototype D the configuration menu appears immediately on element selection. However in this case, instead of appearing right next to the selected element, the menu will fill a pre-determined area on the right side of the work area (figure 4.22).

There was only one task defined for the second session. Users were presented with a pre-made app with some screens and unique widgets already placed and were then asked to make changes to specific properties of specific elements in the app (Appendix B, section B.2).

Users were asked to:
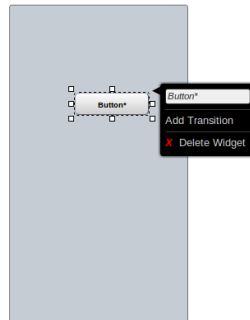
Figure 4.21: FUSAMI Pop-up menu without button. The menu is shown immediatly upon element selection.
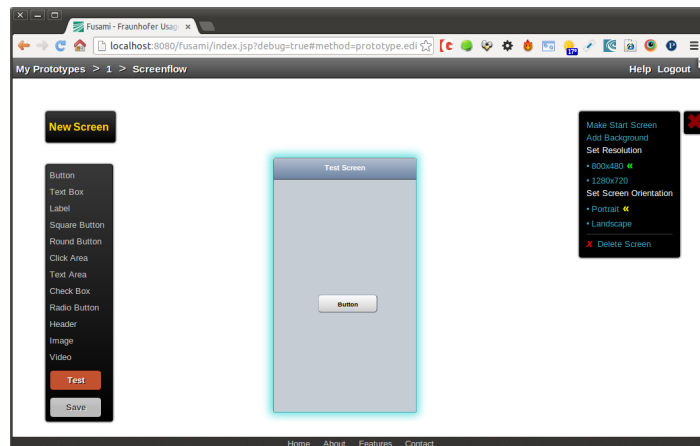
Figure 4.22: FUSAMI fixed section menu.

- Change the smaller screen's resolution

- Change the horizontal screen's orientation

- Change the labels text alignment

- Change the text boxes text color

- Change the combo box's default name

- Create a transition

The users performed the task with each of the prototypes. The order in which each user tested each prototype was alternated throughout the tests for the same reasons highlighted for session 1.

The selection of the elements to integrate the final testing phase was based on the analysis of the performance of each prototype derived from the study of usage data and eye tracking metrics. These results are discussed in greater detail in chapter 5.

These selected elements - one for placement and one for configuration - were then merged into a final prototype. This prototype was the subject of test session number 3. In the final session users were asked to perform two tasks using the prototype. The tasks' nature was mixed as the first task concerned placement and the second configuration.

On the first task users should create two new screens and then place a "Square Button", "Label", "Text Area" and "Video" widgets arbitrarily in whichever screen they desired. Following this they were asked to save their work.

For the second task users were presented with the prototype they had just created. On this prototype they should then proceed to change the prototype's default start screen, change the text alignment on the "Label", change the text color on the "Text Area", create a transition between the two screens and finally run a test on their prototype (Appendix B, section B.3).

As with the first two sessions these tests' results are discussed in more detail in the next chapter.

Chapter 5

Result Analysis

On this chapter we discuss in detail the results obtained for the test sessions with the functional prototypes the test sessions.

There was a wealth of data to analyze. The eye tracking software, Begaze, provided raw data for each test trial in each session covering user event data and eye tracking data. It also provided video replays of the trials with superimposed event and eye tracking data. Additionally we also had access to eye tracking analysis data, specifically the scan paths that could be viewed individually or grouped through whichever criteria was desired: user, session, task or prototype.

Besides this concrete data there were some more subjective satisfaction results to consider. After sessions 1 and 2 users were very simply asked which of the tested prototypes they had preferred. After session 3 they were given the System Usability Scale [Bro96] questionnaire and asked to answer it considering the tool they had just used.

The raw event data was preprocessed in conjunction with the trial video replays. During this process each event was given a text describer. Events were given a classification of "PLACE-MENT", "SELECTION", "NAVIGATION", "ERROR" or "OTHER.

PLACEMENT Placement events were those where the user placed an element on the work area, either a new screen, a new widget on an existing screen or a transition on its destination.

SELECTION Selection events involved the selection and activation of an option. In other words a selection event occurred every time a user pressed a button in order to directly perform and action.

NAVIGATION Navigation events were those where a user did an action that aimed at reaching a feature that was otherwise inaccessible. This included actions such selecting a widget to reach its configuration menu or toggling a menu to access a new option.

ERROR An error tag was given every time an action was performed attempting to complete an objective in a task that didn't directly contribute to reaching that same goal.

OTHER Finally events were classified as other when users performed actions that weren't in direct pursuit of the tasks goal. Some users for example, after placing screens on the prototype, took the time to move to perfectly align them. Although this serves no useful purpose in the completion of the task neither can it be qualified as error in itself.

This pre-processed event data and the raw eye tracking data were treated with purpose written software that processed every text file and returned a summary of the data sorted by user, session, task and metric. This treated data was then imported to Microsoft Excel which allowed us to easily perform calculations over the data and generate graphics to better analyze the results.

Once this process was concluded we were left with comprehensive data covering:

- Time on task (excluding time spent on "OTHER" events)

- Average time between events

- Error introduction rates

- Total number of saccades

- Average saccade times

- Average saccade distances

- Average saccade amplitudes

- Total fixations

- Average fixation duration

We could then proceed to a detailed analysis of the data for each test session.

5.1 Session 1

As previously mentioned, test session number 1 focused on the placement interface with two prototypes, A and B, respectively with an accordion interface (figure 4.18) and an expandable menu (figure 4.19). Users were given two tasks and asked to perform each task with both prototypes for a total of four trials. To counteract the familiarity bias half the users performed the tasks on prototype A first and the other half on prototype B.

Analyzing the results by task, for task 1 the results were exactly as expected. The fact that there was no navigation involved meant that the data was virtually identical for both prototypes (figures 5.1, 5.2, 5.3 and 5.4).

Furthermore analyzing the scan path revealed that, on the menu area (the main area of interest) the scanpaths for both prototypes were almost identical as seen in figure 5.5.
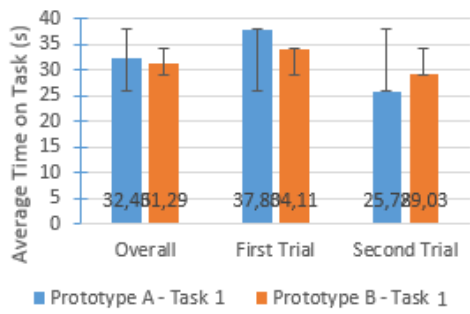
Figure 5.1: Session 1 task 1 average time on task.
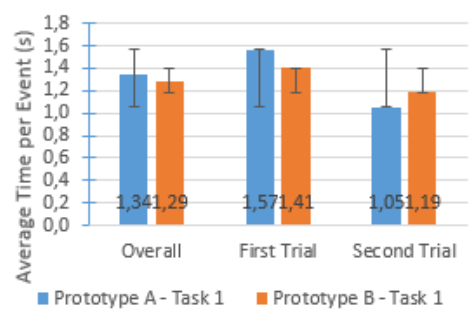
Figure 5.2: Session 1 task 1 average time per event
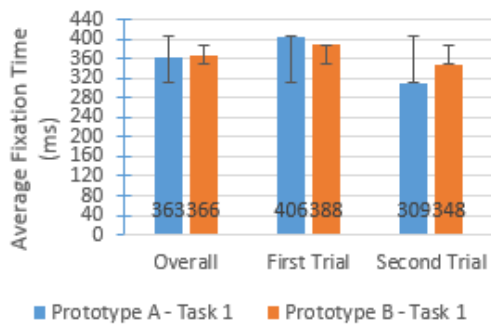
Figure 5.3: Session 1 task 1 average fixation time
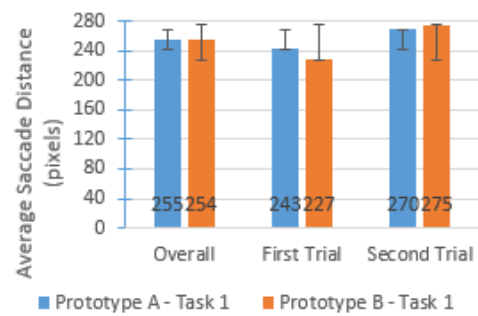
Figure 5.4: Session 1 task 1 average saccade distance
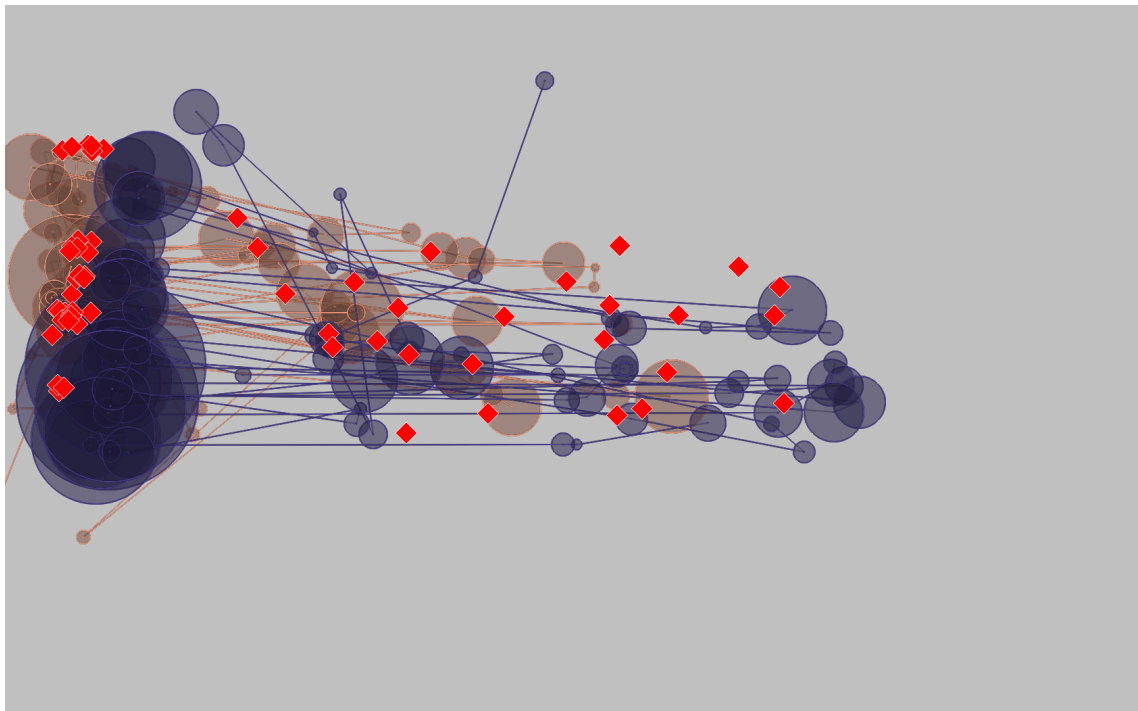
Figure 5.5: A comparison of the scanpaths in task 1 of session 1 for both prototyprs. In the leftmost section (the menu area) they are almost identical.

For task 2 it was an altogether different story. Although the overall results remained reasonably close, things were significantly different even among trials of the same prototype if you considered results separately according to order of usage, i.e. one group of results for when prototype A was used first and another when it was used last and then the same for prototype B. Analysing the results this way a marked trend quickly became apparent: Prototype A had worst results when executed in first place but easily outperformed prototype B when executed in second place. If we take average time per event for example prototype A went from being 7% slower when used in first place to 10% faster when used in second place, a 16% swing (figure 5.6). As far as time on task goes it went from 36% worse on first use to within less than 3% difference on second use (figure 5.7). This after both metrics having nearly identical results when analysing overall numbers.
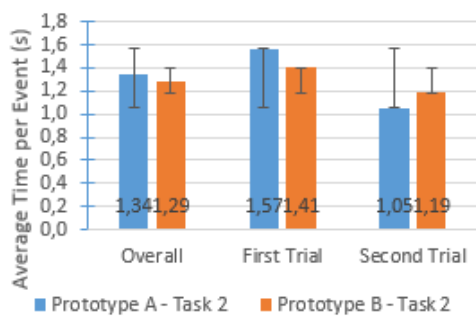
Figure 5.6: Session 1 task 2 average time per event.
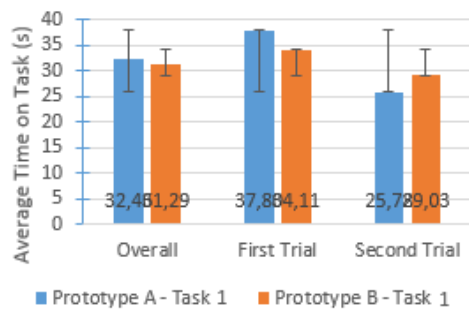
Figure 5.7: Session 1 task 2 average time on task.

The eye tracking data ended up confirming the results from the event data. Average fixation time went from 20% worse to, again, within less than 3% difference [figura]. Although other eye tracking metrics proved inconclusive, there was something else that corroborated these results. When watching video replays of the trials it became obvious that users were getting "lost" in the expanded menu. The gaze indicator could be clearly seen wandering up and down in the menu. Meanwhile, on the accordion menu, the section headers were extremely effective as guides directing the users to the correct sections. Proof of this is the fact that the "Headers" section which needn't be opened to perform the task was opened only once throughout all tests.
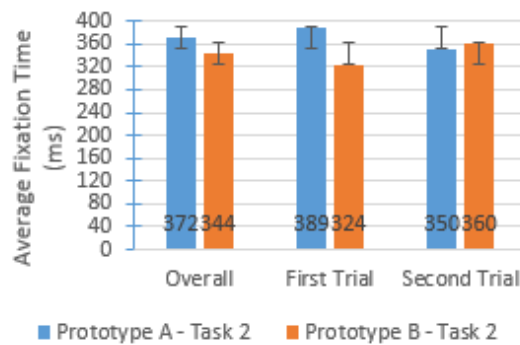
Figure 5.8: Session 1 task 2 average fixation time.

This results validity is based on the fact that if such marked differences were observed after just one use of the tool and performing a task once (and that those differences weren't visible in task 1 in essentially the same circumstances) they should at the very least be maintained for future uses if not increased.

When we asked the test subjects for their preference, seven out of the eleven elected prototype A while only two selected prototype B with another two manifesting no preference between the two.

This superiority anchored on both hard data and satisfaction manifestation led to the selection of prototype A as the one to carry on to the final testing phase.

5.2   Session 2

Test session 2 covered element configuration interface. Three prototypes were created. Prototype C with a popup menu with activation button (figure   4.20]), prototype D with a popup menu without activation button that appeared on element selection (figure   4.21]) and and prototype E with a fixed section of the screen reserved for the configuration menu (figure   4.22]). For this session users had only one task to perform, once with every prototype. In this instance the task consisted of editing the properties of several elements already placed in a pre-made prototype.

Once again it proved relevant to group results by order of execution. In this case one group was created for first execution while second and third were merged in a single group.

Overall time on task measures for session 2 favored prototype E. However, after discriminating first execution from the following ones, prototype D was shown to require the least time after familiarization by a margin of 17% to prototype E who in turn was only 6% better than prototype C (figure   5.9).

The results for average time per event behaved in a similar manner. When considering all occurrences, it was this time prototype C who provided better times. However, after the split analysis it was also prototype D that yielded the better results with times 24% faster than prototype E but only 2% faster than prototype C (figure   5.10).
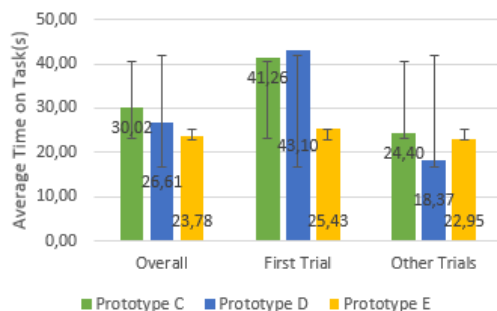
Figure 5.9: Session 2 average time on task.
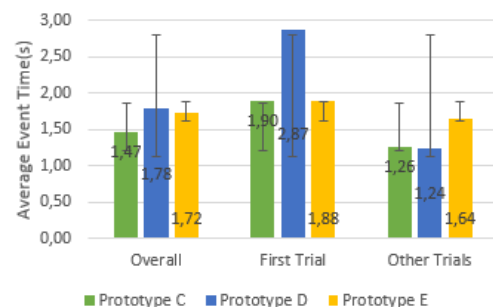
Figure 5.10:   Session 2 average time per event

43

When it came to error rate however prototype D proved to be consistently better. It had better results overall, it was only slightly (0,33 errors) behind prototype C on first use and again, when it came to usage after familiarization, it proved much better with a performance eight times better than prototype C and seventeen times better than prototype E (figure 5.11).
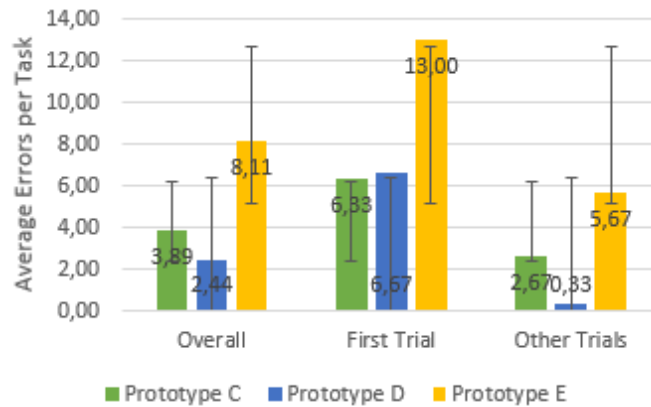
Figure 5.11: Session 2 average error rate.

When analyzing the scanpaths it was, in this case, interesting to trace the convex hull polygons for each of the three prototypes' combined scanpath. As evident in figure 5.12 Prototype D's convex hull was significantly smaller than C's and E's (somewhat surprisingly with prototype C considering their great structural similarity) which indicates a more direct and efficient search and consequently a better interface [CBD02, GK98].

Analyzing the prototypes more subjectively Prototype C had too big of a navigation overhead. All the necessary button clicking was too time consuming and the eye tracking data proved that users sometimes had difficulty locating said button which ended up creating a high number of errors. Prototype E was much closer to prototype D but ultimately the increased travel time (for both mouse pointer and gaze) caused by the menus larger distance to the elements being edited proved too much.

Users' opinion validated the test results with six out of the ten participating users electing prototype D as their favorite in session 2 while only three opted for prototype E. The remaining user was undecided and none of them opted for prototype C.

Prototype D was therefore the one selected to carry on for final testing.

Result Analysis
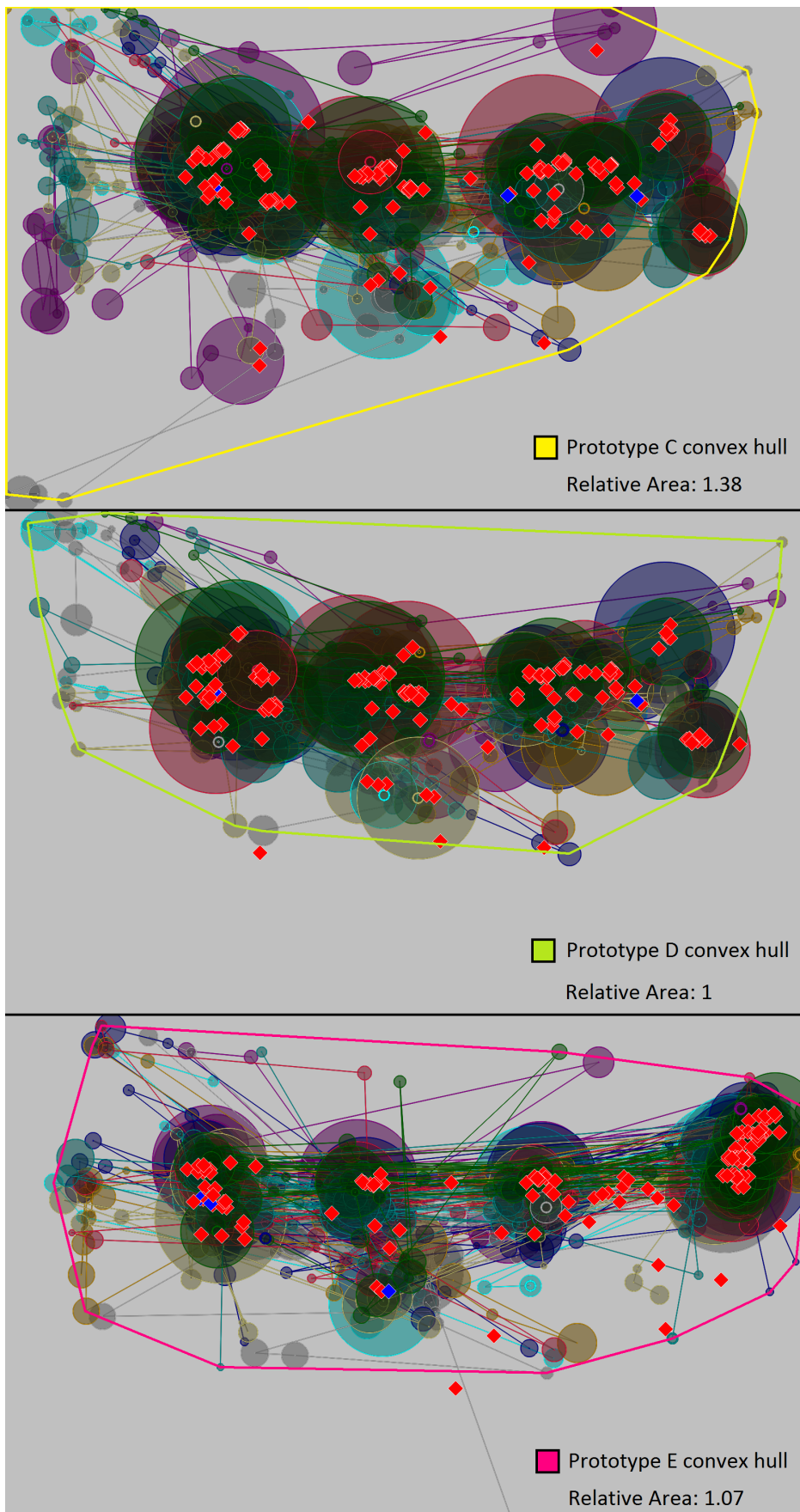
Figure 5.12: Convex hull comparison of Configuration prototypes.

45

5.3 Session 3

For session 3 we then had a prototype consisting of a merger between prototype A and D. An accordion menu for widget selection and a popup menu for element configuration. In this session users had two tasks, the first focused on element placement and the second on element configuration.

Analyzing overall metrics such as time on task, total saccades or total fixations made little sense since the tasks were intrinsically different from the ones in the previous sessions. We then had to focus on the averages which are more a product of the tool than of the task itself.

In terms of average time between events for task 2 we verified that the number for session 3 was right between those for first execution and for the others (figure 5.13). This would indicate that the users remembered some of the interface but not as much as they did immediately after the first use.
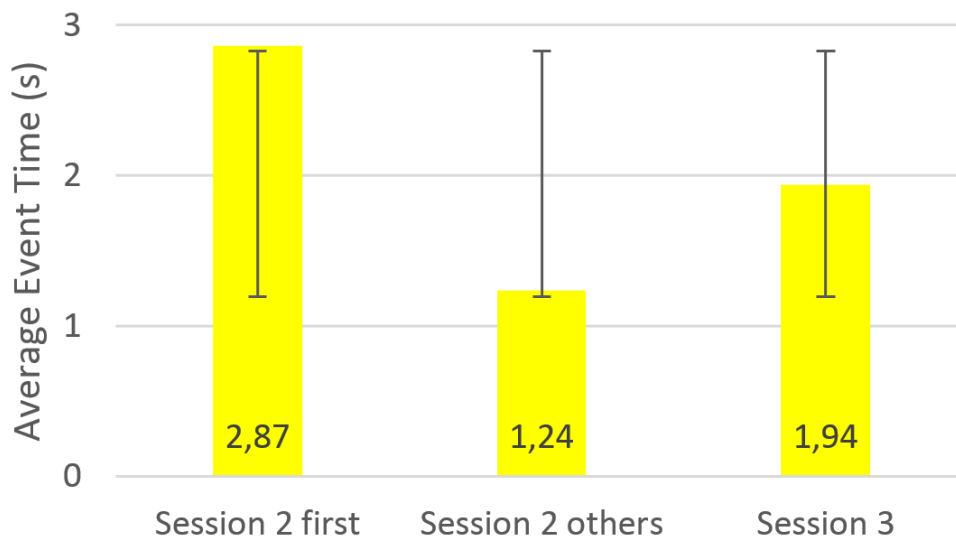
Figure 5.13: Comparison of different event times of prototype D and the final prototype.

Interestingly this pattern in data held for eye tracking metrics therefore corroborating the results as visible on figures 5.14 and 5.15.

For task number 1 it was slightly more difficult as the event data didn't show the same trend. The eye tracking data however did. We can see in figures 5.16 and 5.17 that these metrics follow the same pattern indicating the same thing: there was some learning of the system although it understandibily wasn't as much at this point as it was right after the first execution.

Although initially intended, time constraints prevented the realisation of a second round of tests on the final prototype intended precisely to assert the system's learnability.
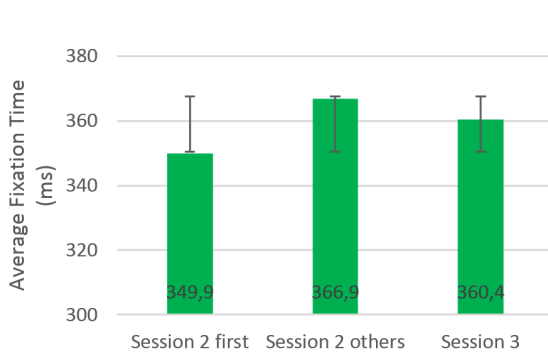
Figure 5.14: Comparison of different fixation times of prototype D and the final prototype.
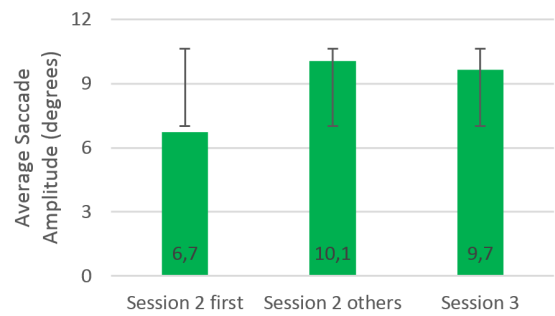
Figure 5.15: Comparison of different saccade amplitudes of prototype D and the final prototype.

After the tests users were then given the System Usability Scale questionnaire regarding the system they had used (Appendix C). After tallying all scores we calculated the average result as 79.8 out of 100 possible points (table 5.1). This corresponds to the 89% percentile [Sau11] which is quite remarkable for a tool that isn't even fully developed yet with a long way still to go.

This early development was however evident when performing a detailed analysis of the individual questions. While questions 2 through 10 presented a close average score (close to 3.5) question 1 presented a much lower average score (2.4). The question was if users would like to use the system frequently. This answer indicates that despite the improvements introduced the tool is yet a ways away from being professionaly viable.

Table 5.1: System Usability Scale results with average result of 79.8 out of 100.

SUS	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
User 1	0	3	2	2	1	1	1	2	2	2	40
User 2	3	3	3	3	3	3	3	4	3	3	77.5
User 3	3	4	4	4	3	4	4	4	3	4	92.5
User 4	2	4	3	4	3	3	3	3	3	4	80
User 5	4	3	4	3	4	4	4	4	4	4	95
User 6	3	3	4	4	3	4	3	3	2	4	82.5
User 7	3	4	3	4	3	4	3	4	3	4	87.5
User 8	3	4	4	4	4	4	4	4	3	0	85
User 9	2	4	4	4	4	4	4	4	2	4	90
User 10	3	3	4	4	3	4	3	4	4	4	92
User 11	2	1	1	3	3	3	1	2	3	3	55
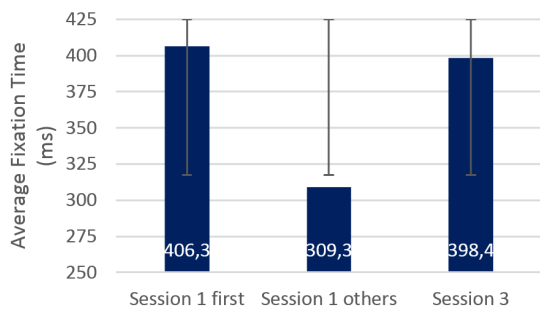
Figure 5.16: Comparison of different fixation times of prototype A and the final prototype.
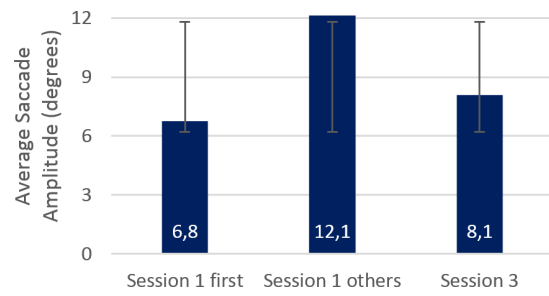
Figure 5.17: Comparison of different saccade amplitudes of prototype A and the final prototype.

Chapter 6

Conclusions

The current pervasiveness of smart mobile devices in today's world coupled with its exponential growth projections over the next few years opens up a similarly increasing market for mobile apps. Despite its growth tendencies the crowd of developers and the sheer number of new apps released every day make this market a very competitive one. It is therefore increasingly important to be able to produce and release applications on as short a development cycle as possible. This need for speed however, must not compromise product quality. One of the chief ways of assuring application quality, from a user interaction standpoint is to perform usability tests. They however, represent one of the great time overheads in this development cycle. This overhead comes from the need to produce prototypes for testing. These prototypes are usually either onerous to produce or devoid of the complex behaviour present in the final product needed to produce relevant test results. On top of this the people responsible for creating such prototypes often don't have the technical skills to produce said functionality.The solution to this problem is to create a tool that allows technically leigh usability experts to create high functionality prototypes very easily and quickly allowing the shortening of the development cycle.

The research done in this work took place on top of a previously existing tool developed at Fraunhofer AICOS. The FUSAMI framework provided a suite of analytical tools that allowed the processing of data from externally created prototypes as well as a mobile prototype editor that allowed the creation of mobile apps that could be quickly edited and ran live on a smartphone. This editor was however in the early stages of development and was not only difficult to use but also lacking in the necessary functionality to make a viable prototyping tool. Our job was then to further the development of FUSAMI's prototype editor into a complete functional tool.

6.1   Work Summary

The work done was divided into four main stages:

- Research

- Development

- Testing

- Result Analysis

The research phase consisted of three steps. First a review of the state of the art was performed. It comprised the analysis of several similar mobile prototyping tools in order to discover which particular design paradigms worked for a few particular problems and could be tested in our editor. This review also contemplated some of the tools, technologies and techniques used for testing and improving usability that were later applied in the development process. The final research step consisted of performing surveys and conducting a focus group among usability, HCI and design professionals that allowed us to restrict and refine the data gathered during the literature review in order to obtain a more relevant set of features to implement and test.

The development focused on two different thing. For the early part of the work period the focus was on implementing several features that were deemed necessary to create a useful tool. These features included the creation of new usable elements (widgets) to use on the prototypes, the addition of new configuration options for elements such as resolution and orientation setting for screens and color and default text edition for widgets and also some usability features such as keyboard shortcuts and proportional resizing. After the research phase ended and the features to test were defined, development centered on implementing the user interface features necessary to the different prototypes that would enter testing.

After development was concluded the resulting five prototypes were submitted to usability tests. They were distributed through two sessions according to the main focus of each prototype. The first session tested the object placement interface and the second session tested the object configuration interface. Each session consisted of tasks aimed at testing the relevant elements of the interface performed in all prototypes available for that session.

The data from these tests was gathered, processed and analysed in order to assess the performance of each prototype. The best performing prototypes from each round were then combined into a final version and submitted to a final testing round.

6.2   Future Work

There is still a significant amount of work to be done to improve the FUSAMI prototype editor. The first concern is to finish the development of the transition interface that we were unable to complete due to time constraints. At this point it is only completed from a conceptual standpoint.

Conclusions

In the near future development will focus on implementing the features and functionalities identified in section 4.3.1. Specifically, the implementation of the drag and drop placement paradigm, the undo and redo system coupled with the auto-save feature, the project wide resolution settings, the widget customization features and the style selection according to target environment.

On the long run the main objectives are the expansion of the transition interface functionality to support a wider range of actions such as web services or database calls as well as the integration of the prototype editor with the analytics suit in order to collect usage data directly from the prototypes with no need for development outside of FUSAMI.

Upon completion, these features would truly make FUSAMI a fully featured and self sufficient tool for the prototyping and usability testing of mobile apps.

Conclusions

References

[Bro96]   John Brooke. System usability scale. Available at http://hell.meiert.org/core/pdf/sus.pdf, last accessed on June 16 2013, 1996.

[Bur00]   Catherine M Burns. Navigation strategies with ecological displays. *International Journal of Human-Computer Studies*, 52(1):111–129, 2000.

[CBD02]   Laura Cowen, Linden Js Ball, and Judy Delin. An eye movement analysis of web page usability. In *People and Computers XVI-Memorable Yet Invisible*, pages 317–335. Springer, 2002.

[CC73]   Tom N Cornsweet and Hewitt D Crane. Accurate two-dimensional eye tracker using first and fourth purkinje images. *JOSA*, 63(8):921–928, 1973.

[CH92]   T.T. Chairman-Hewett. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

[Cis13]   Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017. Available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf, last accessed on February 6 2013, 2013.

[Cod11]   Codiqa. Codiqa. Available at http://www.codiqa.com/, last accessed on February 6 2013, 2011.

[DCC01]   Susan Dumais, Edward Cutrell, and Hao Chen. Optimizing search by showing results in context. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 277–284. ACM, 2001.

[Dil01]   Andrew Dillon. Beyond usability: process, outcome and affect in human-computer interactions. *Canadian Journal of Information Science*, 26(4):57–69, 2001.

[DKvD11]   Jason I. Hong Douglas K. van Duyne, James A. Landay. Deign patterns at a glance. Available at http://www.designofsites.com/design-patterns/, last accessed on February 6 2013, 2011.

[ECM+98]   Steve Ellis, Ron Candrea, Jason Misner, Christopher Sean Craig, Christopher P Lankford, and Thomas E Hutchinson. Windows to the soul? what eye movements tell us about software usability. *Usability Professional's Association, Washington DC*, 1998.

[Exa10]   Exadel. Tiggzi. Available at http://tiggzi.com/home, last accessed on February 6 2013, 2010.

REFERENCES

[GHR99]   David Golightly, Kate S Hone, and Frank E Ritter. Speech interaction can support problem solving. In *Proceedings of IFIP TC. 13 International Conference on Human–Computer Interaction. IOS Press, Amsterdam*, pages 149–155. Citeseer, 1999.

[GK98]   JH Goldberg and XP Kotval. Eye movement-based evaluation of the computer interface. *Advances in occupational ergonomics and safety*, pages 529–532, 1998.

[GL85]   John D Gould and Clayton Lewis. Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.

[Gro05]   Object Management Group. Bpmn. Available at http://www.omg.org/spec/BPMN/2.0/, last accessed on February 6 2013, 2005.

[GSL+02]   Joseph H Goldberg, Mark J Stimson, Marion Lewenstein, Neil Scott, and Anna M Wichansky. Eye tracking in web search tasks: design implications. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 51–58. ACM, 2002.

[Hor06]   K. Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies*, 64(2):79–102, 2006.

[HWJM+89]   Thomas E Hutchinson, K Preston White Jr, Worthy N Martin, Kelly C Reichert, and Lisa A Frey. Human-computer interaction using eye-gaze input. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1527–1534, 1989.

[IF]   IBM and Dojo Foundation. Maqetta. Available at http://maqetta.org/, last accessed on February 6 2013.

[Jav79]   Emile Javal. Essai sur la physiologie de la lecture. In *Annales D\'Oculistique*, 1879.

[JC76]   Marcel Adam Just and Patricia A Carpenter. Eye fixations and cognitive processes. *Cognitive Psychology*, 8(4):441–480, 1976.

[JK03]   Robert JK Jacob and Keith S Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

[JMCS05]   CH Judd, CN McAllister, N Cloyd, and WM Steele. General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Monographs*, 1905.

[Lew94]   James R Lewis. Sample sizes for usability studies: Additional considerations. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 36(2):368–378, 1994.

[M+75]   Richard A Monty et al. An advanced eye-movement measuring and recording system. *The American psychologist*, 30(3):331, 1975.

[ME99]   Evangelos N Mitsopoulos and Alistair DN Edwards. A principled design methodology for auditory interaction. In *Proceedings of IFIP TC. 13 International Conference on Human–Computer Interaction. IOS Press, Amsterdam*, pages 263–271, 1999.

REFERENCES

[MFJ01]   Diarmid Marshall, John C Foster, and Mervyn A Jack. User performance and attitude towards schemes for alphanumeric data entry using restricted input devices. *Behaviour & information technology*, 20(3):167–188, 2001.

[ML12]   Netek Martin Landa. Possibilities of contactless control of web map applications by sight. Available at http://geoinformatics.fsv.cvut.cz/gwiki/Possibilities_of_contactless_control_of_web_map_applications_by_sight, last accessed on June 16 2013, 2012.

[MM58]   JANE F MACKWORTH and NH Mackworth. Eye fixations recorded on changing visual scenes by the television eye-marker. *JOSA*, 48(7):439–444, 1958.

[MT62]   Norman H Mackworth and Edward Llewellyn Thomas. Head-mounted eye-marker camera. *JOSA*, 52(6):713–716, 1962.

[NBGR01]   Marcus Nyberg, Staffan Bjork, Mikael Goldstein, and Johan Redstrom. Handheld applications design: merging information appliances without affecting usability. In *Proceedings of IFIP TC. 13 Conference on Human–Computer Interaction. IOS Press, Amsterdam*, pages 391–398, 2001.

[Nie94]   J. Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM, 1994.

[Nie03]   J. Nielsen. Usability 101: Introduction to usability. *Jakob Nielse's Alertbox, August*, 25, 2003.

[NL93]   Jakob Nielsen and Thomas K Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 206–213. ACM, 1993.

[ORMR12]   A. Oulasvirta, T. Rattenbury, L. Ma, and E. Raita. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing*, 16(1):105–114, 2012.

[PB05]   Alex Poole and Linden J Ball. Eye tracking in human-computer interaction and usability research: Current status and future. In *Prospects", Chapter in C. Ghaoui (Ed.): Encyclopedia of Human-Computer Interaction. Pennsylvania: Idea Group, Inc*. Citeseer, 2005.

[PBP05]   Alex Poole, Linden J Ball, and Peter Phillips. In search of salience: A response-time and eye-movement analysis of bookmark recognition. In *People and Computers XVIII—Design for Life*, pages 363–378. Springer, 2005.

[RP89]   Keith Rayner and Alexander Pollatsek. *The psychology of reading*. Routledge, 1989.

[Sau10]   Jeff Sauro. Why you only need to test with five users (explained). Available at http://www.measuringusability.com/five-users.php, last accessed on June 16 2013, 2010.

[Sau11]   Jeff Sauro. Measuring usability with the system usability scale. Available at http://www.measuringusability.com/sus.php, last accessed on June 16 2013, 2011.

REFERENCES

[Sof11]   Fluid Software. Fluid ui. Available at http://www.fluidui.com/, last accessed on February 6 2013, 2011.

[SS01]   Jared Spool and Will Schroeder. Testing web sites: Five users is nowhere near enough. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 285–286. ACM, 2001.

[TA08]   T. Tullis and W. Albert. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Morgan Kaufmann, 2008.

[TF84]   HM Tong and RA Fisher. Progress report on an eye-slaved area-of-interest visual display. Technical report, DTIC Document, 1984.

[TH00]   David A Tyfa and Mark Howes. Speech recognition for command entry in multi-modal interaction. *International Journal of Human-Computer Studies*, 52(4):637–667, 2000.

[TS04]   Thomas S Tullis and Jacqueline N Stetson. A comparison of questionnaires for assessing website usability. In *Usability Professional Association Conference*, 2004.

[Val11]   Carlos Valdecanos. All the numbers, all the facts on mobile the trillion-dollar industry. why is google saying: Put your best people on mobile. Available at http://consultantvalueadded.com/2011/03/07/mobile-as-a-trillion-dollar-industry-check-this/, last accessed on February 6 2013, 2011.

[Ver99]   Roel Vertegaal. The gaze groupware system: mediating joint attention in multi-party communication and collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 294–301. ACM, 1999.

[VHJG95]   J. Vlissides, R. Helm, R. Johnson, and E. Gamma. Design patterns: Elements of reusable object-oriented software. *Reading: Addison-Wesley*, 1995.

[Vir90]   Robert A Virzi. Streamlining the design process: Running fewer subjects. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 34, pages 291–294. SAGE Publications, 1990.

[Vir92]   Robert A Virzi. Refining the test phase of usability evaluation: how many subjects is enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 34(4):457–468, 1992.

[WBH88]   John Whiteside, John Bennett, and Karen Holtzblatt. Usability engineering: Our experience and evolution. *Handbook of Human–Computer Interaction, edited by M. Helander, Amsterdam: North Holland*, 1988.

[WC01]   Alan Woolrych and Gilbert Cockton. Why and when five test users aren't enough. In *Proceedings of IHM-HCI 2001 conference*, volume 2, pages 105–108. Cépadèus Toulouse,, France, 2001.

[WCW01]   SJ Westerman, T Cribbin, and R Wilson. Virtual information space navigation: Evaluating the use of head tracking. *Behaviour & Information Technology*, 20(6):419–426, 2001.

REFERENCES

[WvdAD⁺06] P. Wohed, W. van der Aalst, M. Dumas, A. ter Hofstede, and N. Russell. On the suitability of bpmn for business process modelling. *Business Process Management*, pages 161–176, 2006.

[ZMI99] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 246–253. ACM, 1999.

REFERENCES

58

Appendix A

Online Survey

WYSIWYG Design environment usability study

The goal of this questionnaire is the collection of data to be used in the further development of a web based mobile app prototype editor. All the questions should be considered within that context.

This questionnaire is aimed at people who work in the design, prototyping and development of mobile apps. It is completely anonymous and should take about 10 minutes of your time.
* Required

Part I

Questions on the first part of this survey pertain to your opinions on visual design environments.

Question 1 *

Consider the problem of a design menu containing a very large number of widgets than can't all be displayed simultaneously. What would, in your opinion, be the best way to display them?

- ☐ A menu with a small number of the most used widgets and an option to show more.
- ○ A scrollable list containig all available widgets.
- ○ A tabbed list with widgets grouped by category.
- ○ Other: _____

Question 2

Please select the widgets you would find more valuable from those available in the list below. (At most FIVE)

- ☐ Checkboxes
- ☐ Radio buttons
- ☐ Text links
- ☐ Combo boxes
- ☐ Grids(Tables)
- ☐ Date/Time pickers
- ☐ Switches(Toggles)
- ☐ Star ratings
- ☐ Sliders
- ☐ Swap views
- ☐ Maps
- ☐ Inputs/Text boxes

Question 3 *

You've added some widgets to your prototype screen and now you need to configure them. When a widget is selected on a prototype, edit options are presented. In your opinion, what's the best way to display these options?

○ A fixed portion of the editing environment is dedicated to showing the configuration options of whatever element is selected.

○ Whenever a widget is selected, a pop-up appears next to it containing the configuration options.

○ Whenever a widget is selected, a button apears next to it that, when clicked, shows a configuration context menu (that is otherwise hidden).

○ When a widget is selected, a toolbar is shown containing shortcuts to several smaller configuration menus.

○ Other: [   ]

Question 4 *

Grade the usefulness of the following features in a visual design environment from 1 to 6. (1=least usefull, 6=most usefull. Each grade should be given only once)

	Least useful					Most useful
Snap grids	○	○	○	○	○	○
Keyboard shortcuts	○	○	○	○	○	○
Tooltips	○	○	○	○	○	○
Color editing	○	○	○	○	○	○
Different resolution settings	○	○	○	○	○	○
Locking/Grouping elements in a screen	○	○	○	○	○	○

Question 5 *

Out of the following, which would best describe your current ocupation?

○ Designer
○ Programmer
○ HCI expert
○ Other: [   ]

Question 6 *

In a scale of 1 (Not experienced at all) to 5 (Very experienced) how experienced would you say you are with the usage of digital prototyping tools and Javascript?

	Not experienced at all		Moderately experienced		Very experienced
Digital prototyping tools	○	○	○	○	○
Javascript	○	○	○	○	○

Question 7 *

How useful would you find if-else logic? (prototype control flow)

	Not useful		Moderatly useful		Extremely useful
Usefulness	○	○	○	○	○

Question 8

Optionally, use the space below to tell us about some features/functionalities you'd like to see in a tool like this that may not have been covered in this survey.

Question 9 *

Which platform do you design/develop for? (If more than one, please choose your primary one)

○ Android

○ iOS

○ Windows Phone

○ Other: [_____]

Question 10 *

When designing/developing, do you follow platform style guides?

○ Yes

○ No

Question 11 *

When tasked with creating a prototype for a cross-platform app, do you create a different prototype for each platform?

○ Yes

○ No

Question 12 *

Do you do low fidelity(paper) or high quality mockups?

○ Low fidelity

○ High quality

○ Both

Question 13 *

Do you use interactive mockups?

○ Yes

○ No

Question 14 *

Which tools do you currently use for prototype design?

☐ Balsamiq

☐ Photoshop

☐ Powerpoint

☐ Eclipse

☐ Visio

☐ OmniGraffle

☐ Axure RP

☐ iRise

☐ Pencil

☐ SmartDraw

☐ MockupScreens

☐ Adobe Fireworks

☐ Adobe Flash

☐ Other: _____

Question 15

In your opinion, how useful is crowd testing in the development process of design environments?

	Not useful		Moderatly useful		Extremely useful
Usefulness	○	○	○	○	○

Appendix B

Usability Test Scripts

B.1 Session 1

You will be performing usability tests on FUSAMI's mobile prototype editor. In this session enphasis will be on the placement of new elements on the work area. You will be given two different tasks and asked to perform each of them in two different versions of the editor's interface for a total of four tasks.

I would ask you to try and keep focus while working at your normal speed.

B.1.1 Task 1.1

You will be presented with a new, empty prototype. In this prototype you will be required to add three new screens.

After you've done that, you will add the following elements to each of the screens you just created:

- A Button

- A Text Box

- A Label

B.1.2 Task 1.2

You will be presented with a new, empty prototype. In this prototype you will be required to add three new screens.

After you've done that, you will add the following elements to each of the screens you just created:

- A Square Button

- A Check Box

- A Video

B.2 Session 2

In this session we will focus on the configuration of existing elements in the work area. You will be given one task and will perform that task on three different versions of the prototype editor's interface.

B.2.1 Task 2.1

You will be presented with a prototype containing three screens each containing different elements. In this prototype you will be required to perform the following actions:

- One of the three screens has a smaller resolution then the others. Change it to match the other two

- One of the three screens has horizontal orientation (landscape). Change it to vertical (portrait)

- There is one "Label" element in the prototype. Change its text alignment from the default value

- There is one "Text Box" element in the prototype. Change its text color from the default value

- There is one "Check Box" element in the prototype. Make any change to its default text

- Add a transition between two screens

B.3   Session 3

In this session, tests will cover both placement and configuration of elements. You will be given two tasks and will perform them on one version of the prototype editor. This version is constituted by elements you have already used on previous sessions.

B.3.1   Task 3.1

You will be presented with a new, empty prototype. In this prototype you will be required to add two new screens.

After you've done that, you will add one the following elements to the project, in whichever screen you desire:

- A Label

- A Button

- A Text Area

- A Video

Finally you will save your work.

B.3.2   Task 3.2

You wil be presented with the prototype you have just created.

In this prototype you will be required to perform the following actions:

- Change the prototype's default start screen

- Change the Label's text color

- Make a change to the Text Area's default text

- Create a transition between the two screens

Finally you will test your prototype.

Appendix C

Post Test Questionnaire - System Usability Scale (SUS)

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Using SUS

The SU scale is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents should be asked to record their immediate response to each item, rather than thinking about items for a long time.

All items should be checked. If a respondent feels that they cannot respond to a particular item, they should mark the centre point of the scale.

Scoring SUS

SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.

To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.

SUS scores have a range of 0 to 100.

The following section gives an example of a scored SU scale.

Appendix D

Test Results

D.1 Time on Task

Here we show time on task values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for the time on task results of all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in seconds.

Table D.1: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	37,85	18,86	32,25	29,13	41,51	24,30	39,51	30,81	37,90	32,46	7,12
	Prototype B	29,53	22,65	21,38	36,78	35,60	34,65	28,74	42,38	29,92	31,29	6,39
Task 2	Prototype A	41,65	23,77	26,01	39,74	53,62	27,83	37,88	37,31	46,46	37,14	9,28
	Prototype B	28,87	22,77	23,58	36,71	41,34	23,18	29,43	38,26	33,26	30,82	6,57

Table D.2: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	37,85	32,25	41,51	39,51	37,90	37,80	3,08
	Prototype B	22,65	36,78	34,65	42,38		34,11	7,20
Task 2	Prototype A	41,65	26,01	53,62	37,88	46,46	41,12	9,21
	Prototype B	22,77	36,71	23,18	38,26		30,23	7,28

Table D.3: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	18,86	29,13	24,30	30,81		25,78	4,65
	Prototype B	29,53	21,38	35,60	28,74	29,92	29,03	4,53
Task 2	Prototype A	23,77	39,74	27,83	37,31		32,16	6,58
	Prototype B	28,87	23,58	41,34	29,43	33,26	31,30	5,89

Table D.4: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	37,85	18,86	32,25	29,13	41,51	24,30	39,51	30,81	37,90	32,46	7,12
	Prototype D	29,53	22,65	21,38	36,78	35,60	34,65	28,74	42,38	29,92	31,29	6,39
	Prototype E	28,87	22,77	23,58	36,71	41,34	23,18	29,43	38,26	33,26	30,82	6,57

Table D.5: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	49,11	34,00	40,68	41,26	6,18
	Prototype D	58,98	29,14	41,17	43,10	12,26
	Prototype E	18,73	26,74	30,82	25,43	5,02

Table D.6: Session 2 results for other uses.

								Average	Standard Deviation
Task 1	Prototype C	22,63	26,07	18,30	26,09	29,14	24,19	24,40	3,38
	Prototype D	11,80	17,99	18,96	20,07	16,62	24,77	18,37	3,89
	Prototype E	23,57	21,85	19,68	30,28	24,25	18,07	22,95	3,91

Table D.7: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	25,78	32,16	29,03	31,30	24,40	18,37	22,95
Standard Error	1,70	2,40	1,85	2,41	1,51	1,74	1,75
Median	25,78	32,16	29,28	30,36	25,13	18,48	22,71
Standard Deviation	4,16	5,88	4,53	5,89	3,70	4,26	4,28
Confidence Level (95%)	4,37	6,18	4,76	6,18	3,88	4,47	4,49

D.2 Average Event Time

Here we show average event time values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for the average event time results of all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in seconds.

Table D.8: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	1.48	0.76	1.36	1.13	1.73	1.03	1.67	1.30	1.59	1.34	0.32
	Prototype B	1.24	0.93	0.90	1.53	1.47	1.45	1.11	1.72	1.23	1.29	0.25
Task 2	Prototype A	1.27	0.91	0.97	1.33	1.63	1.04	1.27	1.34	1.39	1.24	0.27
	Prototype B	1.17	0.92	0.96	1.47	1.67	0.94	1.20	1.55	1.33	1.24	0.27

Table D.9: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	1.48	1.36	1.73	1.67	1.59	1.57	0.15
	Prototype B	0.93	1.53	1.45	1.72		1.41	0.34
Task 2	Prototype A	1.27	0.97	1.63	1.27	1.39	1.31	0.24
	Prototype B	0.92	1.47	0.94	1.55		1.22	0.34

Table D.10: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	0.76	1.13	1.03	1.30		1.05	0.23
	Prototype B	1.24	0.90	1.47	1.11	1.23	1.19	0.21
Task 2	Prototype A	0.91	1.33	1.04	1.34		1.15	0.21
	Prototype B	1.17	0.96	1.67	1.20	1.33	1.26	0.26

Test Results

Table D.11: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	1.15	1.37	0.88	1.30	2.30	1.36	2.03	1.30	1.53	1.47	0.37
	Prototype D	3.85	2.25	2.50	1.06	1.45	1.29	1.12	1.20	1.29	1.78	0.72
	Prototype E	1.78	1.54	1.45	1.52	2.12	1.66	1.28	1.85	2.29	1.72	0.68

Table D.12: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	2.30	1.36	2.03	1.90	0.48
	Prototype D	3.85	2.25	2.50	2.87	0.86
	Prototype E	1.52	1.85	2.29	1.88	0.39

Table D.13: Session 2 results for other uses.

								Average	Standard Deviation
Task 1	Prototype C	1.15	1.37	0.88	1.30	1.30	1.53	1.26	0.22
	Prototype D	1.06	1.45	1.29	1.12	1.20	1.29	1.24	0.14
	Prototype E	1.78	1.54	1.45	2.12	1.66	1.28	1.64	0.29

Table D.14: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	1.05	1.15	1.19	1.26	1.26	1.24	1.64
Standard Error	0.07	0.07	0.08	0.09	0.09	0.06	0.12
Median	1.05	1.15	1.21	1.23	1.30	1.24	1.60
Standard Deviation	0.18	0.16	0.19	0.23	0.22	0.14	0.29
Confidence Level (95%)	0.19	0.17	0.20	0.24	0.23	0.15	0.31

D.3 Error Rate

Here we show errors commited per trial for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for errors commited for all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

Table D.15: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	0	1	4	2	0	0	0	1	0	0.89	1.36
	Prototype B	0	1	0	5	0	6	3	1	0	1.78	1.79
Task 2	Prototype A	2	3	1	0	0	1	1	1	0	1.00	1.16
	Prototype B	0	0	0	2	1	0	4	0	0	0.78	1.93

Table D.16: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	0	4	0	0	0	0.80	1.79
	Prototype B	1	5	6	1		3.25	2.63
Task 2	Prototype A	2	1	0	1	0	0.80	0.84
	Prototype B	0	2	0	0		0.50	1.00

Table D.17: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	1	2	0	1		1.00	0.82
	Prototype B	0	0	0	3	0	0.60	1.34
Task 2	Prototype A	3	0	1	1		1.25	1.26
	Prototype B	0	0	1	4	0	1.00	1.73

Test Results

Table D.18: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	1	5	3	5	8	7	4	1	1	3.89	2.59
	Prototype D	3	13	4	1	0	0	0	0	1	2.44	3.49
	Prototype E	2	24	2	34	6	0	0	2	3	8.11	9.34

Table D.19: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	8	7	4	6.33	2.08
	Prototype D	3	13	4	6.67	5.51
	Prototype E	34	2	3	13.00	18.19

Table D.20: Session 2 results for other uses.

							Average	Standard Deviation
Task 1	Prototype C	1	5	3	5	1	1	2.67	1.97
	Prototype D	1	0	0	0	0	1	0.33	0.52
	Prototype E	2	24	2	6	0	0	5.67	9.24

Table D.21: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	1	1.25	0.6	1	2.67	0.33	5.67
Standard Error	0.26	0.40	0.49	0.63	0.80	0.21	3.77
Median	1	1.125	0	0.5	2	0	2
Standard Deviation	0.63	0.97	1.2	1.55	1.97	0.52	9.24
Confidence Level (95%)	0.66	1.02	1.26	1.63	2.06	0.54	9.70

D.4 Total Saccades

Here we show total saccades done during the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for saccades performed in all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

Table D.22: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	98	65	167	82	101	79	88	102	90	96.89	28.82
	Prototype B	94	69	91	120	81	87	116	102	85	93.89	19.89
Task 2	Prototype A	135	80	86	95	117	134	88	96	128	106.56	25.34
	Prototype B	99	82	76	98	99	39	105	193	98	98.78	30.25

Table D.23: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	98	167	101	88	90	108.80	32.98
	Prototype B	69	120	87	102		94.50	21.70
Task 2	Prototype A	135	86	117	88	128	110.80	22.66
	Prototype B	82	98	39	193		103.00	64.97

Table D.24: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	65	82	79	102		82.00	15.25
	Prototype B	94	91	81	116	85	93.40	13.61
Task 2	Prototype A	80	95	134	96		101.25	23.03
	Prototype B	99	76	99	105	98	95.40	11.19

Test Results

Table D.25: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	94	82	91	98	101	87	88	193	85	102.11	36.73
	Prototype D	98	69	167	82	81	79	88	102	128	99.33	31.75
	Prototype E	135	65	86	120	117	39	116	102	90	96.67	29.56

Table D.26: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	101	87	88	92.00	7.81
	Prototype D	98	69	167	111.33	50.34
	Prototype E	120	102	90	104.00	15.10

Table D.27: Session 2 results for other uses.

								Average	Standard Deviation
Task 1	Prototype C	94	82	91	98	193	85	107.17	42.45
	Prototype D	82	81	79	88	102	128	93.33	18.93
	Prototype E	135	65	86	117	39	116	93.00	36.38

Table D.28: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	82	101	93	95	107	93	93
Standard Error	05	07	05	04	17	08	15
Median	82	99	92	99	93	85	101
Standard Deviation	12	18	12	10	42	19	36
Confidence Level (95%)	12	19	13	11	45	20	38

D.5 Average Saccade Time

Here we show average saccade duration values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for the average saccade duration results for all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in miliseconds

Table D.29: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	46.93	51.96	79.30	58.27	41.89	174.85	44.53	53.69	57.05	67.61	41.68
	Prototype B	78.18	41.20	80.43	37.65	42.08	116.55	42.89	71.26	48.72	62.11	25.57
Task 2	Prototype A	47.49	53.60	128.07	48.29	39.83	55.71	50.70	61.60	59.21	60.50	33.96
	Prototype B	62.95	46.98	48.11	38.44	48.02	148.78	40.92	69.61	52.42	61.80	29.63

Table D.30: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	46.93	79.30	41.89	44.53	57.05	53.94	15.29
	Prototype B	41.20	37.65	116.55	71.26		66.66	36.51
Task 2	Prototype A	47.49	128.07	39.83	50.70	59.21	65.06	35.90
	Prototype B	46.98	38.44	148.78	69.61		75.95	50.30

Table D.31: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	51.96	58.27	174.85	53.69		84.69	60.16
	Prototype B	78.18	80.43	42.08	42.89	48.72	58.46	19.22
Task 2	Prototype A	53.60	48.29	55.71	61.60		54.80	5.50
	Prototype B	62.95	48.11	48.02	40.92	52.42	50.48	8.10

Test Results

Table D.32: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	78.18	46.98	80.43	38.44	41.89	116.55	44.53	69.61	48.72	62.81	29.30
	Prototype D	46.93	41.20	79.30	58.27	42.08	174.85	50.70	53.69	59.21	67.36	33.81
	Prototype E	47.49	51.96	128.07	37.65	39.83	148.78	42.89	71.26	57.05	69.44	40.10

Table D.33: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	41.89	116.55	44.53	67.65	42.36
	Prototype D	46.93	41.20	79.30	55.81	20.54
	Prototype E	37.65	71.26	57.05	55.32	16.87

Table D.34: Session 2 results for other uses.

							Average	Standard Deviation
Task 1	Prototype C	78.18	46.98	80.43	38.44	69.61	48.72	60.39	17.89
	Prototype D	58.27	42.08	174.85	50.70	53.69	59.21	73.13	50.21
	Prototype E	47.49	51.96	128.07	39.83	148.78	42.89	76.50	48.59

Table D.35: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	84.69	54.80	58.46	50.48	60.39	73.13	76.50
Standard Error	19.02	1.74	7.02	2.96	7.31	20.50	19.83
Median	71.48	54.80	53.59	49.30	59.16	55.98	49.73
Standard Deviation	46.60	4.26	17.19	7.24	17.89	50.21	48.59
Confidence Level (95%)	48.91	4.47	18.04	7.60	18.78	52.69	50.99

D.6   Average Saccade Distance

Here we show average saccade distance values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for the average saccade distance for all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in pixels.

Table D.36: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	203.37	286.34	151.56	262.50	275.94	265.40	293.37	264.56	288.65	254.63	46.97
	Prototype B	278.25	270.24	223.80	215.01	316.58	216.83	258.30	207.01	299.01	253.89	37.57
Task 2	Prototype A	250.19	212.83	223.05	187.32	217.61	180.37	269.84	225.33	259.12	225.07	41.31
	Prototype B	222.11	239.31	196.37	244.33	238.33	327.66	197.52	291.76	236.10	243.72	40.30

Table D.37: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	203.37	151.56	275.94	293.37	288.65	242.58	62.53
	Prototype B	270.24	215.01	216.83	207.01		227.27	28.96
Task 2	Prototype A	250.19	223.05	217.61	269.84	259.12	243.96	22.75
	Prototype B	239.31	244.33	327.66	291.76		275.76	41.90

Table D.38: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	286.34	262.50	265.40	264.56		269.70	11.16
	Prototype B	278.25	223.80	316.58	258.30	299.01	275.19	36.11
Task 2	Prototype A	212.83	187.32	180.37	225.33		201.46	21.16
	Prototype B	222.11	196.37	238.33	197.52	236.10	218.09	20.28

Test Results

Table D.39: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	278.25	239.31	223.80	244.33	275.94	216.83	293.37	291.76	299.01	262.51	37.51
	Prototype D	203.37	270.24	151.56	262.50	316.58	265.40	269.84	264.56	259.12	251.46	39.36
	Prototype E	250.19	286.34	223.05	215.01	217.61	327.66	258.30	207.01	288.65	252.65	43.03

Table D.40: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	275.94	216.83	293.37	262.05	40.12
	Prototype D	203.37	270.24	151.56	208.39	59.50
	Prototype E	215.01	207.01	288.65	236.89	45.00

Table D.41: Session 2 results for other uses.

							Average	Standard Deviation
Task 1	Prototype C	278.25	239.31	223.80	244.33	291.76	299.01	262.74	30.99
	Prototype D	262.50	316.58	265.40	269.84	264.56	259.12	273.00	21.64
	Prototype E	250.19	286.34	223.05	217.61	327.66	258.30	260.52	41.31

Table D.42: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	269.70	201.46	275.19	218.09	262.74	273.00	260.52
Standard Error	3.53	6.69	13.19	7.40	12.65	8.83	16.86
Median	267.55	201.46	276.72	220.10	261.29	264.98	254.25
Standard Deviation	8.64	16.39	32.30	18.14	30.99	21.64	41.31
Confidence Level (95%)	9.07	17.20	33.89	19.03	32.52	22.71	43.35

D.7 Average Saccade Amplitude

Here we show average saccade amplitude values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for the average saccade amplitudes for all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in degrees.

Table D.43: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	6.58	6.68	8.07	8.58	8.14	24.77	4.97	8.50	6.10	9.15	5.98
	Prototype B	11.01	5.57	7.92	4.45	6.73	17.18	4.80	11.62	6.40	8.41	3.94
Task 2	Prototype A	6.19	6.29	17.35	7.59	4.47	4.60	5.73	9.50	8.25	7.78	4.97
	Prototype B	8.18	5.83	5.15	4.89	8.69	22.64	3.42	11.53	7.09	8.60	4.89

Table D.44: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	6.58	8.07	8.14	4.97	6.10	6.77	1.35
	Prototype B	5.57	4.45	17.18	11.62		9.70	5.90
Task 2	Prototype A	6.19	17.35	4.47	5.73	8.25	8.40	5.18
	Prototype B	5.83	4.89	22.64	11.53		11.22	8.16

Table D.45: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	6.68	8.58	24.77	8.50		12.13	8.47
	Prototype B	11.01	7.92	6.73	4.80	6.40	7.37	2.32
Task 2	Prototype A	6.29	7.59	4.60	9.50		7.00	2.07
	Prototype B	8.18	5.15	8.69	3.42	7.09	6.50	2.20

Test Results

Table D.46: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	11.01	5.83	7.92	4.89	8.14	17.18	4.97	11.53	6.40	8.65	4.83
	Prototype D	6.58	5.57	8.07	8.58	6.73	24.77	5.73	8.50	8.25	9.20	4.93
	Prototype E	6.19	6.68	17.35	4.45	4.47	22.64	4.80	11.62	6.10	9.37	6.07

Table D.47: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	8.14	17.18	4.97	10.09	6.34
	Prototype D	6.58	5.57	8.07	6.74	1.26
	Prototype E	4.45	11.62	6.10	7.39	3.76

Table D.48: Session 2 results for other uses.

							Average	Standard Deviation
Task 1	Prototype C	11.01	5.83	7.92	4.89	11.53	6.40	7.93	2.77
	Prototype D	8.58	6.73	24.77	5.73	8.50	8.25	10.43	7.12
	Prototype E	6.19	6.68	17.35	4.47	22.64	4.80	10.36	7.70

Table D.49: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	12.13	7.00	7.37	6.50	7.93	10.43	10.36
Standard Error	2.68	0.66	0.85	0.80	1.13	2.91	3.14
Median	10.36	7.00	7.05	6.80	7.16	8.38	6.43
Standard Deviation	6.56	1.61	2.07	1.96	2.77	7.12	7.70
Confidence Level (95%)	6.89	1.68	2.18	2.06	2.91	7.47	8.08

D.8   Total Fixations

Here we show total fixations performed during the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for total fixations in all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

Table D.50: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	96	63	164	82	95	84	87	99	90	95.56	27.79
	Prototype B	93	69	92	118	79	101	115	101	85	94.78	19.58
Task 2	Prototype A	135	80	85	95	116	133	88	95	129	106.22	24.94
	Prototype B	98	84	74	99	99	44	106	193	97	99.33	29.61

Table D.51: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	96	164	95	87	90	106.40	32.41
	Prototype B	69	118	101	101		97.25	20.47
Task 2	Prototype A	135	85	116	88	129	110.60	23.07
	Prototype B	84	99	44	193		105.00	63.09

Table D.52: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	63	82	84	99		82.00	14.76
	Prototype B	93	92	79	115	85	92.80	13.65
Task 2	Prototype A	80	95	133	95		100.75	22.63
	Prototype B	98	74	99	106	97	94.80	12.15

Test Results

Table D.53: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	93	84	92	99	95	101	87	193	85	103.22	36.12
	Prototype D	96	69	164	82	79	84	88	99	129	98.89	31.15
	Prototype E	135	63	85	118	116	44	115	101	90	96.33	28.53

Table D.54: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	95	101	87	94.33	7.02
	Prototype D	96	69	164	109.67	48.95
	Prototype E	118	101	90	103.00	14.11

Table D.55: Session 2 results for other uses.

								Average	Standard Deviation
Task 1	Prototype C	93	84	92	99	193	85	107.67	42.17
	Prototype D	82	79	84	88	99	129	93.50	18.73
	Prototype E	135	63	85	116	44	115	93.00	35.05

Table D.56: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	82.00	100.75	92.80	94.80	107.67	93.50	93.00
Standard Error	4.67	7.16	4.98	4.44	17.22	7.65	14.31
Median	82.00	97.88	92.40	97.50	92.50	86.00	100.00
Standard Deviation	11.44	17.53	12.20	10.87	42.17	18.73	35.05
Confidence Level (95%)	12.00	18.40	12.81	11.41	44.25	19.65	36.78

D.9   Average Fixation Time

Here we show average fixation time values for the first two test sessions. We give overall results followed by a discrimination of first use and other uses.

We also present descriptive statistics for average fixation times on all tasks and prototypes in the first two sessions. Trials are labeled as SxTyPz where x is the session number, y is the task number, and z indicates the prototype used.

All results are presented in miliseconds.

Table D.57: Session 1 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype A	406.41	321.41	250.72	367.58	466.35	202.91	480.03	345.11	428.01	363.17	94.07
	Prototype B	309.18	392.50	241.44	498.24	490.45	245.25	328.74	417.87	372.43	366.23	81.97
Task 2	Prototype A	361.41	391.27	283.04	398.92	462.01	238.84	461.02	372.67	379.67	372.09	81.92
	Prototype B	311.05	317.21	326.05	407.60	424.13	288.84	390.22	282.86	350.29	344.25	74.69

Table D.58: Session 1 results for first usage.

							Average	Standard Deviation
Task 1	Prototype A	406.41	250.72	466.35	480.03	428.01	406.30	91.81
	Prototype B	392.50	498.24	245.25	417.87		388.47	105.58
Task 2	Prototype A	361.41	283.04	462.01	461.02	379.67	389.43	75.15
	Prototype B	317.21	407.60	288.84	282.86		324.13	57.63

Table D.59: Session 1 results for second usage.

							Average	Standard Deviation
Task 1	Prototype A	321.41	367.58	202.91	345.11		309.25	73.36
	Prototype B	309.18	241.44	490.45	328.74	372.43	348.45	92.36
Task 2	Prototype A	391.27	398.92	238.84	372.67		350.42	75.20
	Prototype B	311.05	326.05	424.13	390.22	350.29	360.35	46.55

Test Results

Table D.60: Session 2 overall results.

		User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	Average	Standard Deviation
Task 1	Prototype C	309.18	317.21	241.44	407.60	466.35	245.25	480.03	282.86	372.43	346.93	71.03
	Prototype D	406.41	392.50	250.72	367.58	490.45	202.91	461.02	345.11	379.67	366.26	88.53
	Prototype E	361.41	321.41	283.04	498.24	462.01	288.84	328.74	417.87	428.01	376.62	82.71

Table D.61: Session 2 results for first usage.

					Average	Standard Deviation
Task 1	Prototype C	466.35	245.25	480.03	397.21	131.78
	Prototype D	406.41	392.50	250.72	349.88	86.15
	Prototype E	498.24	417.87	428.01	448.04	43.77

Table D.62: Session 2 results for other uses.

								Average	Standard Deviation
Task 1	Prototype C	309.18	317.21	241.44	407.60	282.86	372.43	321.79	60.13
	Prototype D	367.58	490.45	202.91	461.02	345.11	379.67	374.46	101.30
	Prototype E	361.41	321.41	283.04	462.01	288.84	328.74	340.91	65.84

Table D.63: Descriptive statistics for the results of each trial.

	S1T1PA	S1T2PA	S1T1PB	S1T2PB	S2T1PC	S2T1PD	S2T1PE
Mean	309.25	350.42	348.45	360.35	321.79	374.46	340.91
Standard Error	23.20	23.78	33.72	17.00	24.55	41.36	26.88
Median	315.33	361.54	338.59	355.32	313.20	373.62	325.07
Standard Deviation	56.82	58.25	82.61	41.64	60.13	101.30	65.84
Confidence Level (95%)	59.63	61.13	86.69	43.70	63.10	106.31	69.09