Designing User Interfaces for Multi-Device Environments

Master Thesis

Author(s):
Mintsi, Theano

Publication date:
2013-08

Permanent link:
https://doi.org/10.3929/ethz-a-009980080

Designing User Interfaces for Multi-Device Environments

Master Thesis

Theano Mintsi
<theano.mintsi@epfl.ch>

Prof. Dr. Moira C. Norrie
Dr. Michael Nebeling
Maria Husmann

Global Information Systems Group
Institute of Information Systems
Department of Computer Science
ETHZ

Prof. Dr. Pierre Dillenbourg
Computer-Human Interaction in Learning and Instruction
Department of Computer Science
EPFL

15th August 2013

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Everything should be made as simple as possible, but not simpler
Albert Einstein

Abstract

Collaborative use of multiple interactive devices has become an increasing trend in everyday life. Yet, there is a lack of effective methods and tools to design user interfaces that target multi-device and multi-user environments. For now, most applications are designed for a specific device type and type of user. Iterations of the design and implementation are necessary to make the same user interface available for other device types and user roles. This makes the design of user interfaces for multi-device environments cumbersome and time-consuming. Also, keeping the design consistent across the different devices is an issue that is so far not handled sufficiently.

This thesis presents XDStudio, a user interface editor, which addresses these issues and focuses on the design of distributed user interfaces in multi-device and multi-user environments. XDStudio aims to support both expert and non-expert users in designing the distribution of existing web-based user interfaces using two seamlessly connected design modes: a simulated mode, where the design and testing are realized in a central authoring device, and an on-device mode, where both a central device and the peripheral target devices are actively collaborating. XDStudio also offers an instant testing feature enabling the designer to check the distribution at any moment and return to the design process for further editing.

A user study reveals interesting aspects regarding the usability of the tool, helps identify advantages and limitations of the two design modes and shows what kind of distribution scenarios and tasks are better performed in each design mode. The results of the user study finally enable us to propose improvements and areas of future work.

Acknowledgements

First of all, I would like to thank Prof. Moira Norrie and Dr Michael Nebeling for giving me the opportunity to do my master thesis in the GlobIS group and work on a very interesting project. I would also like to thank Prof. Pierre Dillenbourg for accepting to supervise my thesis and for the helpful guidance. I owe special thanks to Michael Nebeling and Maria Husmann for their valuable feedback, advice and all the help during my thesis.
Over the six months of my master thesis, many people helped in different ways and I would like to thank all of them. Many thanks go to Christina for her scenario illustrations. Finally, I would like to express my gratitude to everyone who voluntarily participated in the user study.

The completion of this thesis was part of the graduate program that was co-funded through the Act "I.K.Y Bursary Program process with individualized assessment, academic year 2011-2012" by the O.P. resources "Education and Lifelong Learning" of the European Social Fund (ESF) and the NSRF 2007-2013.

Contents

1 Introduction

In recent years we have witnessed a phenomenal growth in the number and variety of interactive devices available in the mass market. This new technological era has created a novel way in which people perform everyday tasks. Using not only one, but a wide set of interactive devices, such as mobile phones, laptops or tablets, is nowadays a common case and provides people with multiple ways to access information and support work and personal activities.
According to Dearman and Pierce [2] and Kane et al. [8], users gradually tend to use the available resources in collaboration rather than only one at a time. Apart from that, sharing applications with colleagues or friends has also become a trend. This collaborative usage of interactive devices introduces the notion of Distributed User Interfaces (DUIs) which are gaining popularity for:

- empowering users to efficiently handle access over the application by making parts of the user interface (UI) accessible on specific devices or by certain users
- leveraging device-specific characteristics since they offer full functionality of UI parts on different devices and platforms.

Whether the UI distribution considers the case of multiple collaborative devices for performing a single task or of multiple users working on the same application, the collective way of interacting with the application poses the question of how to handle simultaneous usage of multiple devices and the involvement of multiple users. In this context, one of the main matters that rise is how to design UIs that are distributed over multiple devices and accessible by multiple users, so that the application remains consistent and collaborative. This question gives the incentive for this master thesis, which focuses on how to design the distribution of a UI for multi-device environments.
Existing UI Editors are graphical tools with plenty of functionalities, when targeting the design and development of applications for a specific device or platform. A typical example

is displayed in Figure 1.1. This is the Interface Builder of Xcode[1], Apple's development environment, a powerful tool for creating applications for Mac, iPhone and iPad.

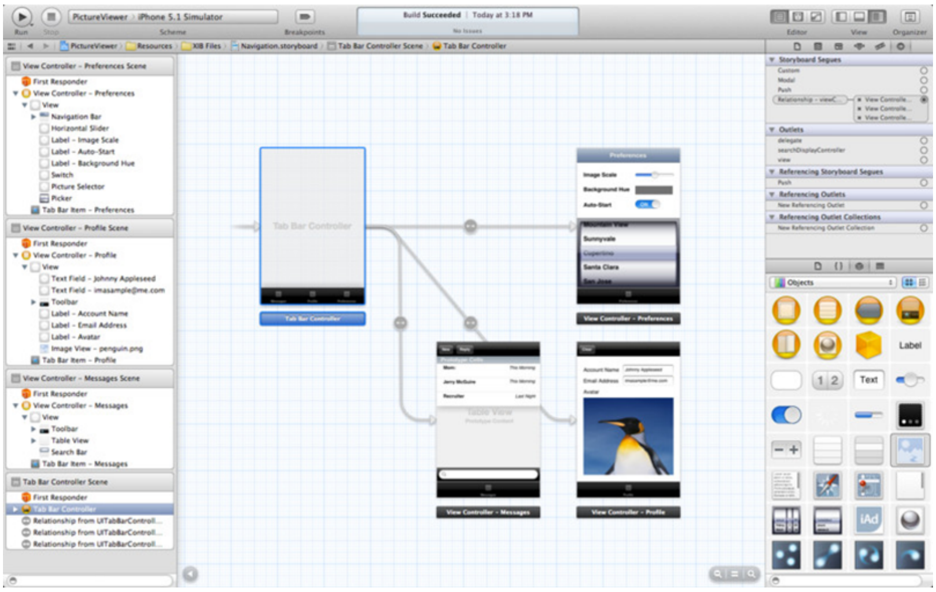


Figure 1.1: A typical example of available GUI builders: Xcode Interface Builder

Yet, when an application is designed and created with such a Graphical User Interface (GUI) Builder, it is usable only on specific types of devices. The whole design and development process would have to start anew for making the same application accessible to other device types, as depicted in Figure 1.2. The restricted flexibility in terms of the target platform is indisputably a major limitation of existing UI editors, when it comes to designing multi-device UIs. Nevertheless, the main shortcoming concerns the distribution of the whole application or parts of it on multiple collaborating devices. To the best of our knowledge, none of the available tools can fully support this task.

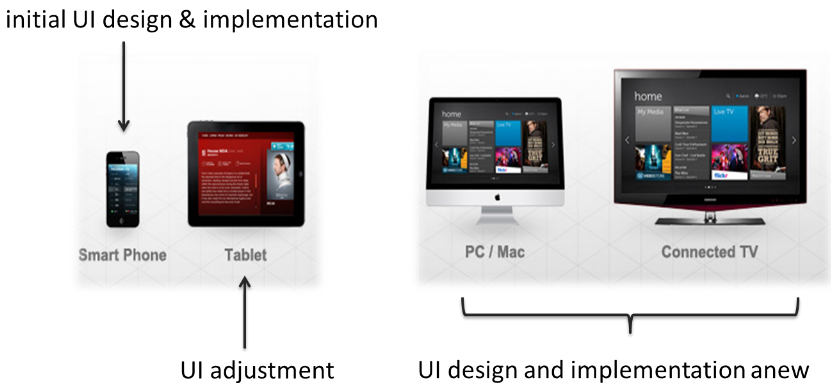


Figure 1.2: Limitation of available GUI builders

[1]https://developer.apple.com/xcode/

Taking into consideration that available UI editors support sufficiently the initial design of a UI and not at all its adaptation to multi-device environments, we decide to put the emphasis of the project in distributing existing UIs on multiple devices. Specifically, we introduce XDStudio, a GUI builder, which aims to assist both expert and non-expert users in designing the distributed version of an existing UI according to the involved devices and user roles. The tool offers flexibility in terms of the design workspace allowing the user to alternate anytime between performing the design on one central device or on the target devices. It also provides context-awareness and enables users to instantly test their design. The design approach behind the tool and the actual features of XDStudio are thoroughly presented later in the thesis.

1.1 Aims of Research

The general goal of the thesis is to introduce a GUI builder for designing the distribution of an existing web-based user interface in a multi-device environment. The web-based approach complies with the overall trend towards web-based applications and supports the flexibility of the tool in terms of targeting multiple devices and platforms. The specific objectives are to:

- Prototype the design tool and introduce the basic concepts that will constitute the basic elements of the editor
- Identify application scenarios where the introduced tool could provide particular value
- Explore two different design modes of the UI editor: a simulated and an on-device mode. In the simulated mode the tool is expected to handle the design of the distributed UI versions from a single central device, while on the on-device mode the target devices should participate in the design process. The tool should support a seamless way to change between the two modes.
- Implement a prototype of the tool in order to prove and evaluate the proposed design approach
- Identify benefits and tradeoffs of the tool for the two design modes for different tasks and explore the usability of the tool through a well-defined user study. Qualitative and quantitative results should be collected and analysed as suggestions for refinement of the UI editor and future work.

1.2 Main Contributions

In this section, we present the main scientific contributions of this work. The proposed design approach and XDStudio, the accompanying tool, enable users to design the distribution of an existing web-based UI based on *multiple context-factors*, such as the target device type or group of devices, the user or group of users, or a combination of these factors. This brings in the notion of *distribution profiles*, one of the key aspects of the project. Distribution profiles let the designer identify the target versions of the distributed UI, design for them and use them in the deployment for easily defining which version of the distributed UI should be available on every involved device.

Another important contribution is the flexibility in terms of the *design workspace*. In more details, the design process can take place either on one central device or on multiple ones, while the switching between the two workspaces is a perfectly consistent and coherent process. This design principle lets XDStudio address different application scenarios and types of users.
Finally, *context awareness* is suggested and supported to a certain extent. XDStudio is able to perceive certain factors of the environment, for the time being surrounding devices, and provide this information to the designer, who can in turn design for these device or view the existing design on connected devices based on their type and the matching distribution profiles. The design principles and actual tool are more explicitly described later in the thesis.

1.3 Structure of the Thesis

In Chapter 2 related and background work is discussed giving an overview of other design approaches and frameworks that handle the design of multi-device UIs and automatic UI adaptation. Chapter 3 describes two example scenarios where our work could provide value. In Chapter 4 the design process, introduced concepts and implementation of XDStudio, the proposed tool for designing UIs for multi-device environments, are presented. Chapter 5 describes the user study that was conducted to evaluate the tool, and also presents and discusses the results of the evaluation. Conclusion, limitations and future work are included in the final Chapter 6.

2
Related Work

This thesis presents XDStudio, a new tool in the area of multi-device UI design. Related work includes GUI builders and logical frameworks for designing user interfaces available either on different devices and platforms or distributed on multiple collaborating devices. Although the two cases have substantial differences for the end-user, the design process shares many similarities. In both cases, the design process needs to take into consideration the fact that multiple kinds of platforms, devices and even users are involved, while consistency across UI versions is a common issue as well.

In the area of multi-device UI design, tools follow two basic approaches: a model-based and a design-oriented one. The model-based approach makes use of an abstract model of the user interface, which enables an automatic UI transformation according to the target device or platform. The design-oriented one is a manual approach, where the designer can define and change the visual design rather than abstract UI models.

The idea of using an abstract model in the core of the design process increases the flexibility regarding the automatic generation of new UIs for multiple target devices and platforms. Yet, it demands from the designer to learn and use a new abstract language. This is one of the basic drawbacks of the model-based approach, which also leads its supporters to keep the abstract model hidden from the designer. Additionally, designers prefer to start the design from a concrete interface rather than having to imagine what the interface would finally look like based on an initial abstract description of it [14] [1]. This observation stands for the more intuitive design-oriented approach, while it is also the reason why model-based tools let the designer create the initial version of the UI in the traditional manual way.

There are multiple tools in the area, however the work presented in this thesis is mostly related to the following frameworks.

A prototyping tool for designing cross-device UIs following the model-based approach is Damask [9]. It introduces two main concepts for supporting cross-device UI prototyping: design patterns and layers. Patterns enable the transformation of the UI in a way that it is optimized for every device, while layers let the user define whether a UI component is available on every device or only on one. The user initially sketches the UI for a device and includes device-specific patterns and layers into it. Damask uses the design to build an abstract model, based on which it can later generate UIs for other devices. The designer can further refine the generated UIs to reach the desired result. In this process an instant testing feature of the tool through an integrated Run window has proven quite helpful, according to the evaluation of this work. The user study additionally revealed that the concept and use of design patterns was positively accepted by the users. Nevertheless, supporting a greater variety of design patterns or the creation of custom ones remains as an open issue to be investigated.

Combining features from both approaches, Gummy [10] presents a multi-platform design method that can automatically create a UI for a new target platform based on existing UIs for the same application. Designers can create an initial UI for a target platform as they would do with typical GUI builders. Based on this design, Gummy creates an abstract model for the UI that gets updated everytime there is a change in the initial version. The abstract model is used by the transformation engine of the tool to automatically generate versions of the UI for new target platforms. The designer can further refine the automatically generated UI versions and the updated UIs remain available to the transformation engine for later automatic generations. Although Gummy makes use of an abstract model to support multi-platform UI design, it offers an intuitive design process. It keeps the UI abstract description hidden from the user, who neither needs to master an abstract language nor has to foresee how the final UI would look like based on a model. This is one the main contributions of this GUI builder. Yet, the tool does not offer consistency across device-specific UIs but only between the initial design and the generated ones.

The idea of an object-oriented data model is supported by another framework, which specifically targets the design of distributed user interfaces. ZOIL [7] introduces both a design approach as well as a software framework which implements the suggested design principles. The basic idea behind ZOIL is that there is a common workspace which includes all the objects, concepts and functionalities available on every involved device. This workspace is supported by an object-oriented data model, that makes it possible to keep the distributed UI versions connected in real-time and consistent, but also different according to the target device, platform or user. One of the suggested design principles of this work is semantic zooming, which enables every device or user to zoom in and visualize parts of the workspace. Another main suggestion is the attached behaviour pattern supported also by the framework. This enables the user to attach behaviours, such as 'draggable' or 'zoomable' to the different objects of the data model, making this way interaction design possible.

The design of distributed user-interfaces is also handled by the work of Ghiani et al. [5]. This work introduces an environment for supporting migratory web user interfaces. Through an HTML proxy-based approach the framework allows users to migrate a web application or parts of it across multiple devices. The proposed framework ensures that the current state of the application is maintained throughout the push and pull process and lets the users continue the running task from the state that they left it. A user study on the tool showed encouraging results and revealed areas of improvements, such as the scalability of the tool for large groups of users.

Towards the manual approach, Jelly [11] is a design environment for handling consistency across devices and overcoming incompatibilities between device- or platform-specific design tools. Jelly supports multiple workspaces for the UI design of different target platforms or devices. This tool offers the flexibility of selecting different UI components and functionalities according to the target platform or device, while at the same time it supports transformation of UI parts across the workspaces. A semantic UI network in the core of the tool enables the cross-device widget copies. A linked editing feature provides content consistency across workspaces. Another contribution of Jelly is the fast-editing feature, which enables the designer to instantly test the UI either through emulators or on the actual target device. Thanks to the see-through approach of the tool, the deployment time is close to a few milliseconds making it possible for the designer to test changes any moment and even go back to design with a click of a button.

A different approach to designing multi-device UIs is introduced by D-Macs [12], a tool that records and replays design actions, which are also provided to other designers as source of inspiration. The tool captures repetitive design actions and through a central web-based repository enables designers to access and assess them. Recorded actions can be replayed and this automatic execution can help designers save time and effort. Yet, the automatic repetition can lead to undesirable results. Thus, the tool provides visual feedback for the designers to have a better understanding of the replication process. The basic idea is that repetitive design actions across designers and device-specific UIs can be automated and used instead of the UI transformation mechanisms of the above-mentioned tools.

Regarding UI adaptation there are two novel approaches, CrowdAdapt [17] and W3Touch [16], that propose design concepts and techniques in relation to our work. CrowdAdapt is a web-design tool, that uses crowd-sourcing in order to collectively improve the adaptability of a web-site under different viewing conditions. W3Touch is a website plug-in, that also turns to crowd-sourcing in order to collect context-aware interaction data and suggest UI adaptations based on a set of introduced usability metrics. Although neither CrowdAdapt nor W3Touch address the design of multi-device user interfaces per se, they suggest and discuss design approaches and techniques, which could be of value for the design of distributed user interfaces.

XDStudio draws inspiration from all these works and builds on top of them to offer a web-based solution focused on designing the distribution of multi-device user interfaces. The approach we take is closer to the manual design-oriented one, since no abstract model needs to be defined or is automatically generated. Notions such as the layers introduced by Damask or the multiple workspaces and linked editing suggested by Jelly are conceptually close to key features of our tool. However, this thesis is primarily interested in experimenting with different distribution scenarios and tasks, and the benefits and tradeoffs of different design modes.

3 Scenarios

In this chapter, we present two application scenarios, which we believe can benefit from the suggested UI editor. These scenarios aim to clarify the needs of the project and are envisioned as test cases. The first one takes place in a meeting room equipped with multiple interactive devices and showcases the collaboration of multiple types of devices. The second scenario revolves around an interactive collaborative learning method suggested by Dillenbourg and Hong [3] and demonstrates that XDStudio can handle different roles of users participating in the distribution. In both cases the scenarios aim to highlight the three main design concepts of XDStudio: *multifactor-based distribution*, *design workspace* and *context awareness*.
For each scenario we describe both the design- and run-time version. Yet, here we put an emphasis on demonstrating how the end-user interacts with the distributed UI as it has been designed with XDStudio. The design process is more thoroughly described in Chapter 4. We consider that the overall application and corresponding UIs have already been designed and implemented and are available to the designer, who using the suggested design principles and tool distributes them targeting the involved devices and users.

3.1 Scenario 1: Meeting Room

As illustrated in Figure 3.1, the scenario takes place in a meeting room, where the whole process is supported by a software application distributed over multiple interactive devices. The application handles the meeting agenda, presentations, questions and comments from the audience and the overall project planning of the group attending the meeting. Additionally, remote participation to the meeting is offered by the application. To support these tasks an interactive whiteboard, an all-in-one PC, a tablet and smart phones are engaged. On the interactive whiteboard the currently presented slide and comments and questions from the audience are displayed. The tablet is used by the presenter to handle the presentation and also see comments and questions inserted by other group members through their smart phones. Finally, the all-in-one PC demonstrates the meeting agenda and project planning. The meeting agenda displays information concerning the presentations of the meeting, such as the presenter, title and related projects to every presentation, whereas more information about the projects the group is involved in is displayed through the project planning. Any remote participants should be able to see on their device a replicated version of the common workspaces and also have the option to submit questions and comments.

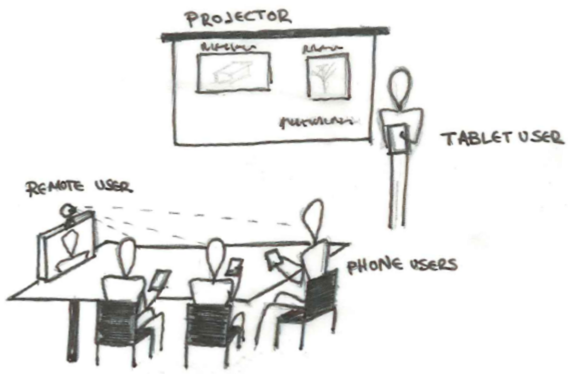


Figure 3.1: Meeting Room Scenario

3.1.1 Design-time scenario

The software application and the corresponding web-based UI have already been developed. A group member has undertaken the role of the designer and has thus taken over the distribution of the UI over the interactive devices, which are expected to be available at the meeting room. Such devices are an interactive whiteboard, an All-in-One desktop, a tablet and smart phones. The designer initially determines which device types are supported and what user roles will be handling which devices. For all different combinations of user roles and devices she carries out the distribution of the UI by assigning the different UI elements on one or multiple views of device (types), see Figure 3.2, and defining the interaction between them, the input throughput and the adaptability on run-time. After completing the distribution from the design perspective, she can go into the use mode to have a more realistic view of what the scenario would look like at run-time. Based on how she likes this overview she can do the changes on the run going back to the design mode.

3.1.2 Run-time scenario

In this plot a research group is in the meeting room and about to start the semestrial meeting. One group member will be attending the meeting remotely, but should still be able to see the presentation and participate in the Q&A session. The meeting coordinator welcomes the participants and makes sure everyone is logged in the session through their smart phones. All mobile devices get detected by the central managing device, which is responsible for providing every device with the respective view and for the sharing among the attached devices.
The coordinator also welcomes the remote participant, who is displayed on a screen on one side of the table and has remote access to the common space using one of his devices. A camera enables the remote group member to see and hear what is happening at the meeting.

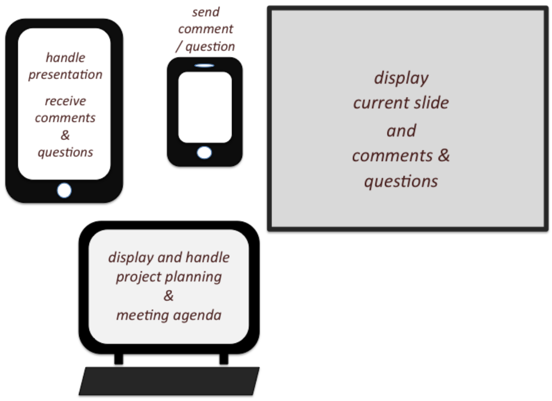


Figure 3.2: Involved Devices and Distribution of UI elements

Another group member joins the meeting a little bit later and gets connected to the distributed application by inserting personal credentials on the smart phone. Based on the type of the device and the role of the user, i.e. participant, he gets access to the respective view.
The coordinator goes to the All-in-One and shortly presents the projects of the group, gives an overview of the involved steps for every project, introduces the plan of the day and relates upcoming presentation with the projects. He then introduces the first presenter and by selecting the corresponding entry in the meeting agenda the view of the whiteboard and the tablet get updated to show the respective presentation. The first presenter takes the tablet and moves towards the interactive whiteboard. Through the tablet he handles the presentation and can see incoming comments and questions from the audience. At some points he stops to discuss about questions, while he prefers to leave general remarks and ideas for discussion at the end of the presentation. The same procedure is followed for the next presentations until the meeting is completed and all group members disconnect from the session.

3.2 Scenario 2: Classroom

The second scenario takes place in a classroom where both students and teacher are equipped with a tablet, see Figure 3.3. The teacher uses an educational method to foster interaction among students by splitting the class into three groups and making different parts of the information available to each group. The idea is that students of different groups will have to interact in order to get the missing information and reach a conclusion depending on the given task.
In this case the lecture is about electromagnetism and the students are introduced to an experiment where a wire is wrapped around a nail and connected to a battery turning this way the nail into an electromagnet which then attracts paper clips, as displayed in Figure 3.3. The students can measure the strength of the magnetic field either by counting the attached paper clips or by using a dedicated device. The class is split in three groups, one that can affect only the battery voltage value, another one that can handle the turns of the wire around the nail and a third one that can alter the length of the nail. Thus, every group can see the effect

of only one variable on the magnetic field, but has to interact with the other groups to find the total impact of the variables.

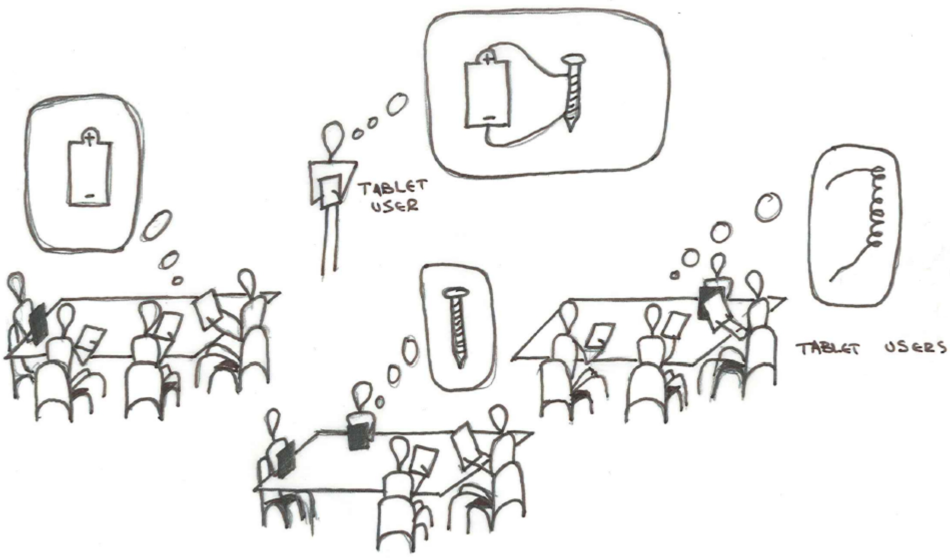


Figure 3.3: Classroom Scenario

The educational process is supported by an application that lets the users change and fix the values of the variables, run the experiment, get the results, draw a graph based on the calculated values and note down some comments and conclusions. During the design process parts of the application UI have been distributed in a way that fits the scenario, i.e. only the teacher can see all three variable inputs and menu that fixes the variables, while the students depending on the group they belong to can see only one of the three variables: battery voltage, turns of wire, size of nail. Everyone can run the experiment, get the results and draw a diagram but only the teacher can see the solution formula for the experiment. In Figure 3.4 the involved device type, user role and different UI elements on each UI view are displayed. In this scenario the UI distribution is realized based on the different roles of the users. The device they are using is the same, yet they have access to another version of the UI.

3.2.1 Design-time scenario

The teacher distributes the UI of the educational software used in classroom using the suggested UI editor. First, she defines the device types that the UI will need to be distributed over. She also defines the different users or groups of users, that will use the distributed UI. Then, she starts the distribution by selecting the corresponding profile, i.e. combination of device type and user role, and migrating parts of the UI over the device view. After completing this task with all target devices and possible users, she goes to use mode to see how it would look like in real-time.

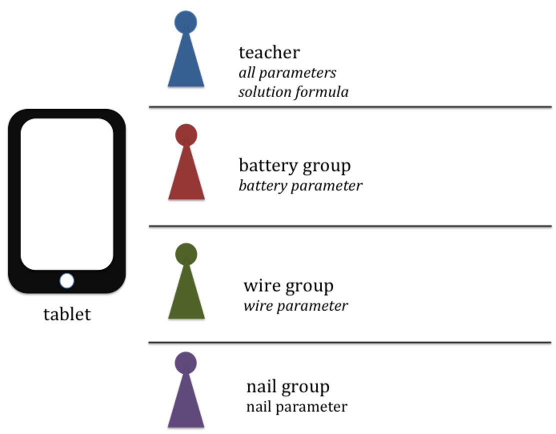


Figure 3.4: Involved Roles and different UI elements

3.2.2 Run-time scenario

In the classroom the teacher divides the class into three groups and the students gain access to the respective version of the application based on the group they belong to. Every student connects to the shared application and plays with the experiment through the provided software. At first, the students discuss the results in their group and come to a conclusion regarding the variable they can change. The battery group notices that the higher the voltage the more paper clips attach to the nail and the higher the value of the magnetic field, thus they write down on the conclusions that the battery and the magnetic field are proportionally related. The same applies for the nail and wire group, who notice that an increase in the number of turns of the wire and in the length of the nail increase the amount of attached paper clips and strength of magnetic field. Afterwards, the three groups discuss their results and come up with a formula combining the effect of all the variables on the value of the magnetic field. The teacher finally pushes the correct formula to the UIs available to the students, who then compare to their findings and verify their solution.

3.3 Selection of Scenarios

We have selected two application scenarios, each of which lets the designer focus on different design aspects. The meeting room scenario is a device-centric one, where the involvement of many different device types in the distribution needs to be handled. What is more, the initial UI is composed by multiple UI items, that are interrelated and distributed across the UI views. On the other hand, the classroom scenario is a rather role-centric one and reveals how XDStudio can handle the involvement of multiple user roles working on the same device type. In this second scenario the initial UI is more simple and only a few elements are differently displayed on each UI view. Additionaly, the different UI views are not interconnected, since the students have to communicate in order to overcome the lack of available information. These differences between the two application scenarios enable a more complete evaluation of XDStudio, when applying them as test cases and make it possible to reveal different features of the proposed tool.

4 XDStudio

In this chapter we thoroughly present XDStudio, the proposed UI editor for distributing web-based UIs in multi-device environments. XDStudio is a new tool that aims to support both expert and non-expert users in designing UIs that are distributed over multiple devices. The tool gives the freedom of working either on one central device in a *simulated design mode*, or involving the target devices in the design process in an *on-device design mode*. The two design modes are consistent and make the alternation possible at any moment as requested by the user. The tool also offers a design-time and run-time support and provides context awareness in terms of detecting certain characteristics of attached target devices and involved users. In the following sections we address the design process and the key concepts introduced by this work. We also present a workflow through the tool and the implementation of the current version.

4.1 Design Process

One of the main challenges was to design the tool in a way that it can offer the desired functionalities, but at the same time be easy-to-use and intuitive. Specifically, throughout the design process of XDStudio we took into consideration the Interaction Design principles, as they are suggested by Moggridge [13]. For a tool to be usable and useful it needs to:

- have a clear mental model
- provide reassuring feedback
- offer navigability, i.e. let the users be aware of where they are in the system, what they can do there, where they can go next and how to get back
- be consistent in terms of interaction and layout
- provide intuitive interaction
- show how it behaves, i.e. the quality of how the users and the system interact.

4.1.1 Design Questions

Before going into details about the design process we would like to present the main design questions we believe that were crucial to be addressed.

1. Why would people distribute a UI in the first place?

   We have identified two main reasons for distributing an interface: The first one is to give access to specific parts of the UI to specific devices or users, such as in case of the classroom scenario presented in section 3.2. The second one is that parts of the UI may require different devices and interactions between them to offer full functionality. A typical example in that direction is the meeting room scenario described in section 3.1.

2. How many people would be able to participate in the design of the distribution?

   At design-time we target a single user for designing the initial distribution of the UI.

3. How many people are expected to use the distributed UI (DUI)?

   Multiple users would be able to work on the DUI. Answering to this question also raises another one regarding whether the number of users or devices is static or dynamic. Our approach targets a dynamic scenario where users and devices can join or leave the distribution scenario at any time.

4. What criteria should the distribution be based on?

   We believe that, according to the scenarios outlined in Chapter 3, the distribution can be based on different aspects:

   - the device type, such as mobile phone or tablet
   - a specific device, such as iPhone or Windows Mobile Phone
   - the role of the user handling the device (type)
   - location and other context factors

   The distribution on device type refers also to sets of device types or specific devices. This can be regarded as views that are targeting a group of device types or specific devices.

5. Is designing the distribution possible at run-time and, if so, how?

   The distribution should be possible at run-time as well. Specific users should be able to distribute parts of the UI running on the devices they are handling to the devices or views accessed by other users. To whom this run-time distribution right is assigned is another design question to be handled. In the current version of the tool distribution in terms of user roles is not yet fully supported at run-time. Nevertheless, we believe that this feature can prove useful and effective, as it can offer users more freedom and flexibility. This is one of the issues that remains open for future work.

6. Which tasks should be supported and, consequently, what features should be provided?

   Here we differentiate between design- and run-time, since we want the tool to support both.

   At design-time users should be able to:

- **Define the criteria based on which the design of the distribution should be realized**
- **Select parts of the initial UI and assign them to one or more views**
- Replace parts of the UI elements with similar ones according to the device characteristics
- Adjust size and position of the UI elements
- Change interaction between UI elements on different devices, such as decoupling connected elements
- **Define user roles, give them access to specific views**
- **Create device-users scenarios that can be used in future distributions. This could also serve in a scenario when the target UI changes or a similar UI needs to be distributed.**
- Define UI views that are necessary for the scenario to run and others that are optional

At run-time users should be able to:

- **Access the UI as defined at design-time**
- **Interact with it**
- Based on their rights and relations to other users, share parts of their UI with other users or device types or specific devices

All features in **bold** are the ones supported by the current implementation. The rest remain to be implemented as future work. Yet, we believe that it is important to introduce them here as desired features in order to demonstrate the bigger picture of the tool we envision.

4.1.2 Paper Prototypes and Digital Mockups

To reach the current form of XDStudio many design iterations were realized. UI prototypes were initially sketched on paper and more refined versions of them were designed as digital mockups using Balsamiq Mockups [1] and Axure RP [2]. The paper prototypes aimed to make an initial approach targeting the aforementioned design questions and other more specific ones that are discussed later. Based on the paper prototypes, the digital mockups were created and led to a more concrete design of the tool. This refined design was used as a model for the version of the tool that was implemented.
At this point, we would like to demonstrate some of the low fidelity paper prototypes. These sketches helped us draw a clearer picture of how the tool should support the desired features and how it should be designed to be usable and useful.
In Figure 4.1, we can see a first approach for the entry screen of the tool. The designer can select between creating a new distribution or opening an already existing one. Here it is also possible to define the URL of the initial UI, create custom devices or detect connected ones and also identify the involved user roles. The designer should also be able to select an already existing configuration of devices and roles from previous projects, in case a new distributed UI version for the same set of devices and roles needs to be designed.

[1] http://balsamiq.com/products/mockups/
[2] http://www.axure.com/features

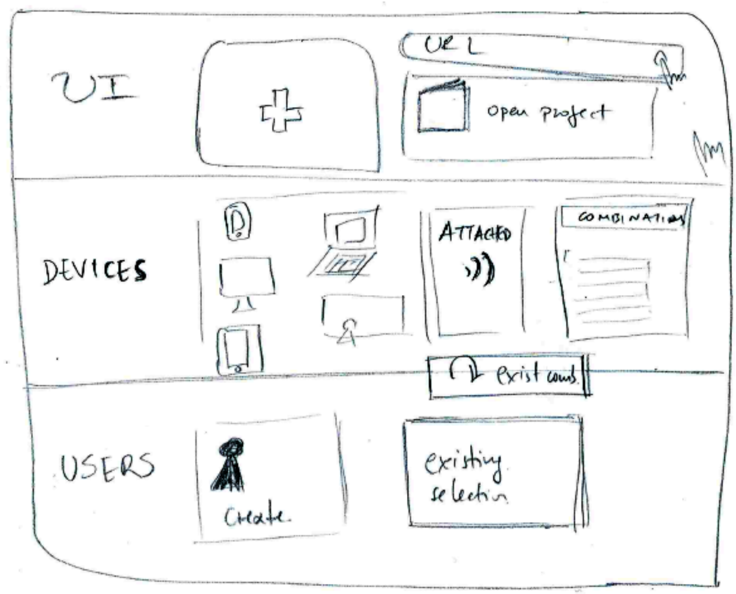


Figure 4.1: First approach for entry screen of XDStudio

Figure 4.2 shows the main design screen, where the selected UI, involved devices and roles are displayed and can be further handled. In Figure 4.3, we can see how the user can select elements from the reference UI on the left and drag and drop them to a specific UI view on the right. By selecting an element the user gets an indication about other UI elements that interact with the selected one. As shown in the respective figure, this could be done by highlighting connected elements. This kind of information can benefit the users when deciding what UI elements are included in every distributed UI view. Interactivity across the UI versions was also one of the design questions that we had to address.

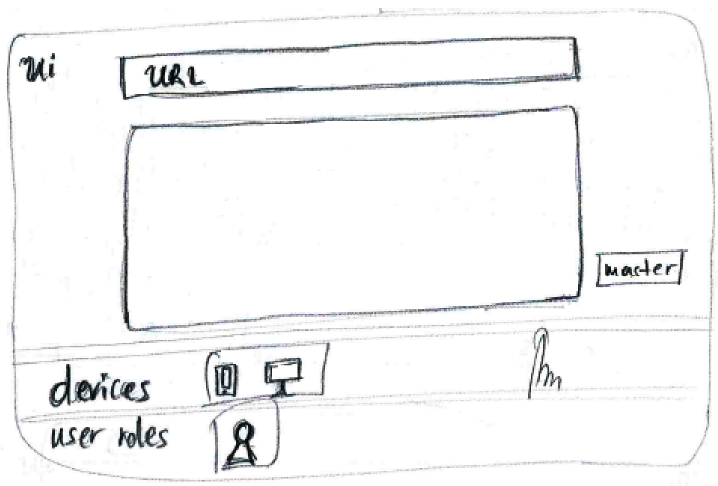


Figure 4.2: Main design screen

One way to handle interaction across devices could be to maintain it by-default according to the initial UI and remove it in specific cases between particular UI views or elements, if

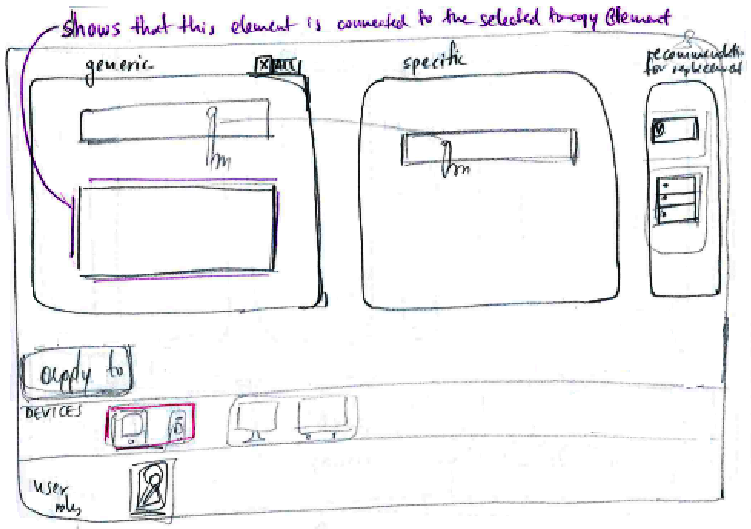


Figure 4.3: Design with drag & drop and get feedback about connected elements

necessary. As we will describe later, XDStudio follows this technique regarding interaction across devices. The other approach could be to initially disable any interactivity across UI views and introduce a new method to interconnect particular UI views or elements, if needed. For this second case a possible solution is displayed in Figure 4.4.

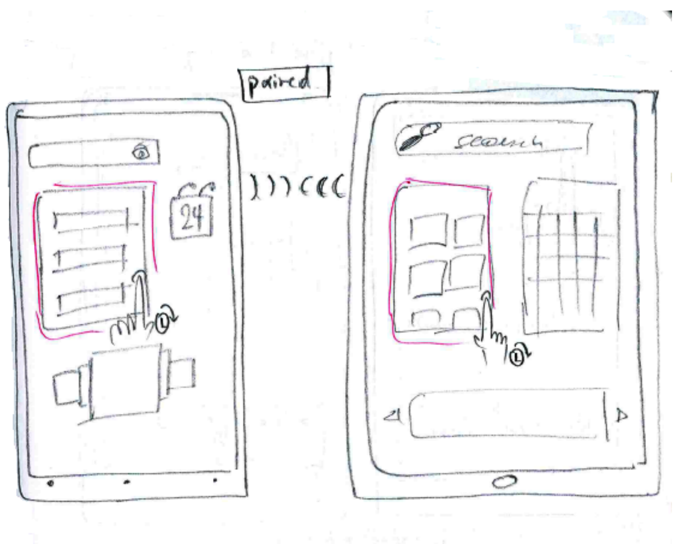


Figure 4.4: Pairing of UI elements accross devices through simultaneous long-tapping

Here we present the most representative digital mockups that show the design for the simulated and on-device modes. In Figure 4.5 we can see a refined version of the main screen in the simulated mode. The user has already defined the URL and the involved devices and roles and can start designing the distribution by dragging and dropping elements from the reference UI on the left to an emulator, here a smart-phone based one, on the right. A *replace with* area on the right side is dedicated to suggesting UI elements that can replace the selected one in order to better match the target device or platform. This feature draws inspiration from the work of Florins and Vanderdonckt [4]. The set of devices and roles highlighted in orange indicate for which device (type) and user roles the design is being realized.

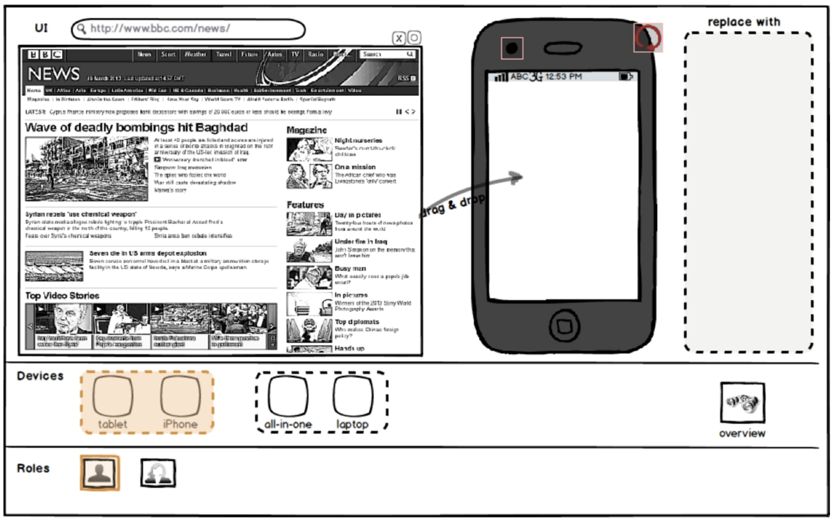


Figure 4.5: Main screen in simulated mode

Figure 4.6 demonstrates the zoom-in view of the specific UI version that is being designed. A toolbar on the left enables the user to further edit the existing design. This toolbar is inspired by the work of Nebeling et al. [17].

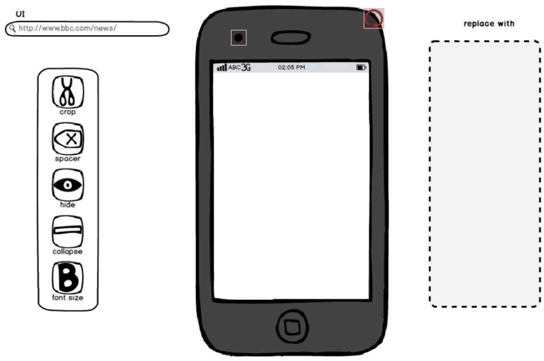


Figure 4.6: Zoom-in specific view for further editing

In the on-device mode the design workspace is spread on a central authoring device and on the target devices. On the left side of Figure 4.7 we present the screen of the central device, where the initial UI and the involved profiles are displayed. On the right we can see the result of the design directly demonstrated on the target device, here on a mobile phone. Further editing is also possible on the target device.

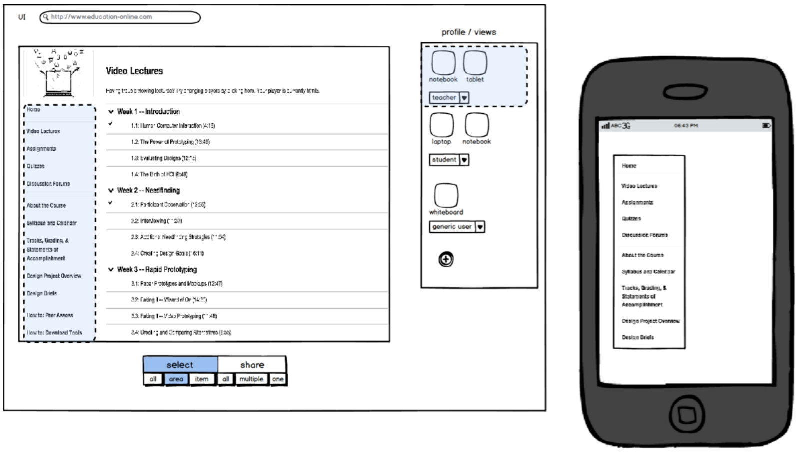


Figure 4.7: Central screen and target device in on-device mode

In Figure 4.8 we present the distributed workspace while designing in the on-device mode. The 'master UI' displayed in the middle is the reference point for this mode. It does not comprise part of the distribution, but is used to handle the definition of the URL and the involved devices and user roles. The design process is distributed on both the master UI and the target devices and consistency is maintained across all devices after changes. In this example we use a central PC executing the master UI and a mobile phone and a laptop as target devices.

Figure 4.8: Distributed Design Workspace in on-device mode

4.2 Concepts

4.2.1 Definitions

The conceptual model behind XDStudio is presented through a diagram in Figure 4.9.

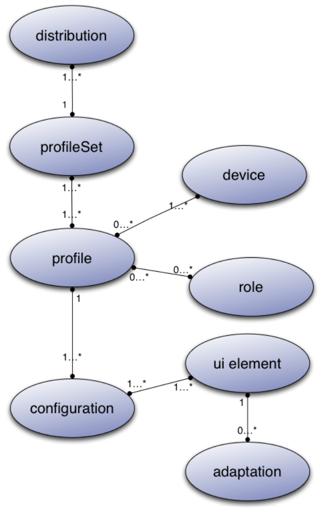


Figure 4.9: Conceptual model of XDStudio: basic notions and their relations

As we have been explaining so far, the distribution of the UI is realized based on what devices (or device types) it will be displayed on, the type of user that will be handling these devices and other context factors. This combination of devices and user roles brings in the idea of the *distribution profile*, which we hereafter mention as *profile*. The notion of the profile makes it easier to define and handle the different versions of the distributed UI.

As displayed in Figure 4.9, the profile is a combined set of at least one device and an arbitrary number of user roles. With device we generally refer to the device type, such as tablet or desktop. Yet, a specific device can be defined as well. The user role is used to identify different types of users that are handling each device. Based on the user role, such as presenter or visitor, the displayed UI on a specific device type can change showing different UI elements and allowing different kinds of interaction with the UI. A set of profiles builds a profileSet, which in other words contains all different profiles that are involved in a scenario. Also for every profile there is a configuration, which describes what is displayed on every UI version and how it has been adapted by the designer. The UI element contains the information regarding the UI elements that are included in every distributed UI view and the adaptation describes how every UI element is adapted for every profile.

All these notions set up a distribution, i.e. an initial UI that is distributed for a specific profileSet consisting of particular profiles, which in turn are made up of device (types) and user roles. For every profile a configuration holds the information about which UI elements of the initial UI are included and how they have been adapted.

The idea is that particular notions are combined to set up a distribution, but are still independent. An already defined profileSet should stay available for future use, in case a new UI needs to be distributed for the same involved profiles. Also, an existing distribution can be adapted to address a new profileSet. This can provide valuable flexibility and save the designer from repeating redundant work.

4.2.2 Application of Concepts

To be more concrete we will show how the conceptual model applies to the classroom scenario presented in 3.2. As displayed in Figure 4.10, in this scenario the UI is distributed in a specific way, which in total is conceptualized with the notion of distribution: collaborative learning. There are four involved profiles:

- the teacher profile, which is represented as [device: tablet, role: teacher]
- the battery group profile represented by the set [device: tablet, role: battery group]
- the wire group profile represented as [device: tablet, role: wire group] and
- the nail group profile depicted as [device: tablet, role: nail group].

These four profiles constitute the profileSet: collaborative learning set. This profileSet could be used in the future by different applications to distribute other UIs on the same set of devices and roles. For every profile there is a corresponding configuration, where the UI elements that are available on each UI view are defined. In this example no adaptation of the UI elements is required.

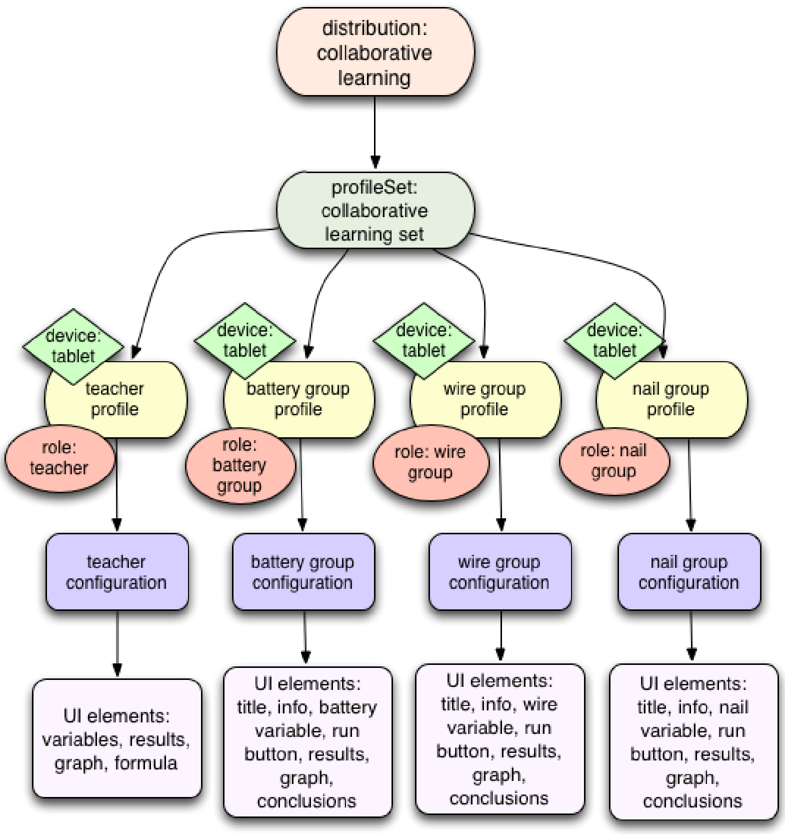


Figure 4.10: Application of Conceptual Model on Classroom Scenario

4.3 System Walkthrough

In this section, we present the current version of XDStudio, walk the reader through the main screens of the UI editor and display its key features. We demonstrate both simulated and on-device modes on the central authoring device and on peripheral target devices. We use the meeting room scenario presented in 3.1 in order to illustrate how XDStudio can support designing the distributed version of the meeting room UI for this scenario.
As we will show in the walkthrough, XDStudio offers two ways to design the distribution. For each profile the designer can either select which elements will be included by dragging and dropping them from the initial UI, which is taken as a reference, or they can select to use a full version of the UI as a starting point for a profile and then tap on specific UI elements in order to remove them.
In Figure 4.11 the main screen of XDStudio in simulated mode is displayed. The current distribution has already been given a name in the welcoming screen and thus in the upper left (1) we can see the annotation 'meeting room scenario', the URL of the website has been loaded (2) and is displayed in the main window (5). Currently the tool is in design mode (3) meaning that the designer can use the displayed UI to distribute it, while by changing to the use mode it would be possible to interact with it, for example by clicking on the previous and next buttons to show the previous or next slide. On the upper right (4) the mode is set to simulated. The rectangular area below (6) is dedicated to creating the involved distribution profiles.

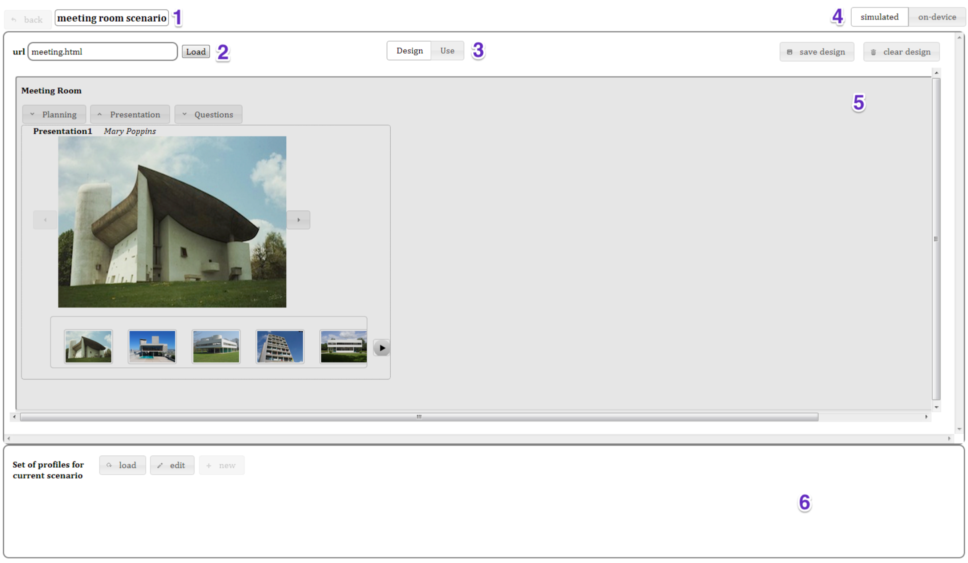


Figure 4.11: Main screen in simulated mode for meeting room UI

The creation of profiles involves the definition of the involved device types. As displayed in Figure 4.12, by clicking on the plus button (1), a device dedicated dialogue opens up, where the user can define the device type, here a mobile device (2), give it a name (3) and add it in the list of involved devices. In the same way the designer can create other devices.

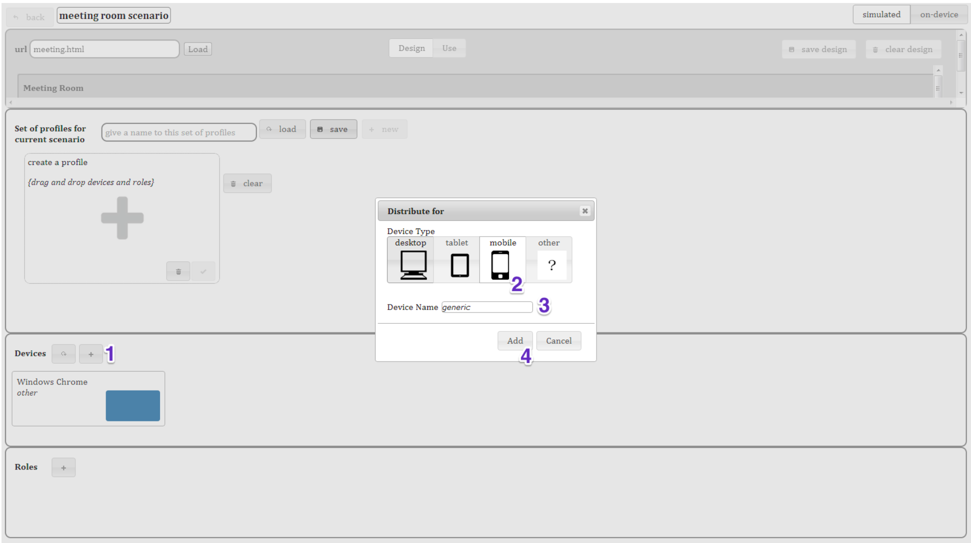


Figure 4.12: Creation of a new device type for the current distribution

The created devices are displayed one next to the other as rectangles in the devices area. In Figure 4.13 we can see that a mobile device, a projector and an All-In-One are already defined. The current implementation supports the device types: ’desktop’, ’tablet’, ’mobile phone’ and ’other’ for any other device type. For this reason the projector and the All-In-One are of type ’other’. The device rectangles are draggable and as displayed in this figure, the All-in-One (1) was dragged and dropped in the profile area (2) and added (3) to the list of existing profiles. The selection can be saved (4) and the user can start designing for the defined profiles.

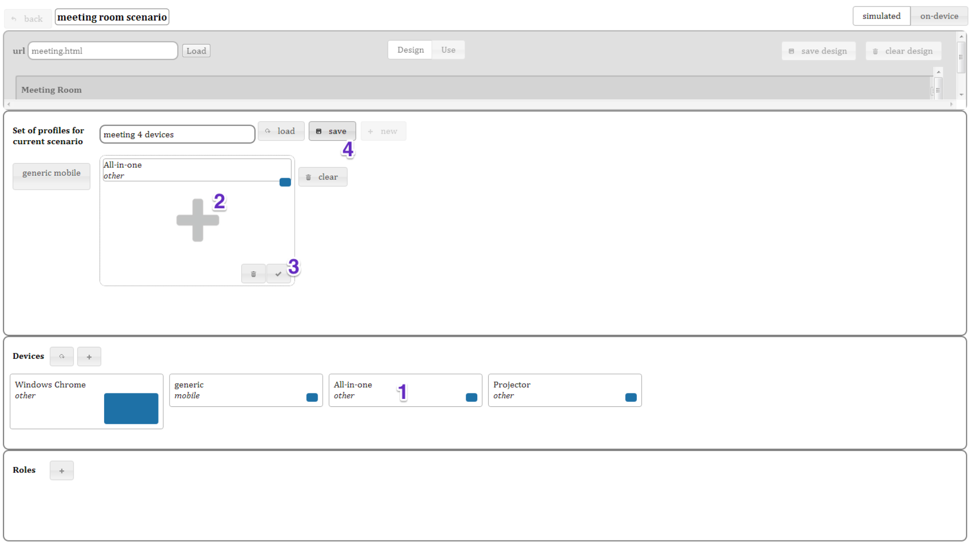


Figure 4.13: Creation of profiles

In Figure 4.14 we can see the profiles list (2), which can be edited (1) at any moment during the design process. This list acts as a set of toggle buttons, enabling the user to select one or more profiles to design for. In this case the projector profile is selected and thus a window is open for designing the projector UI version. The user can select a UI element (3) from the initial UI on the left and drag and drop it on the projector window (4). Once the design is ready for a profile the user should save it (5) in order to be able to reload it later.

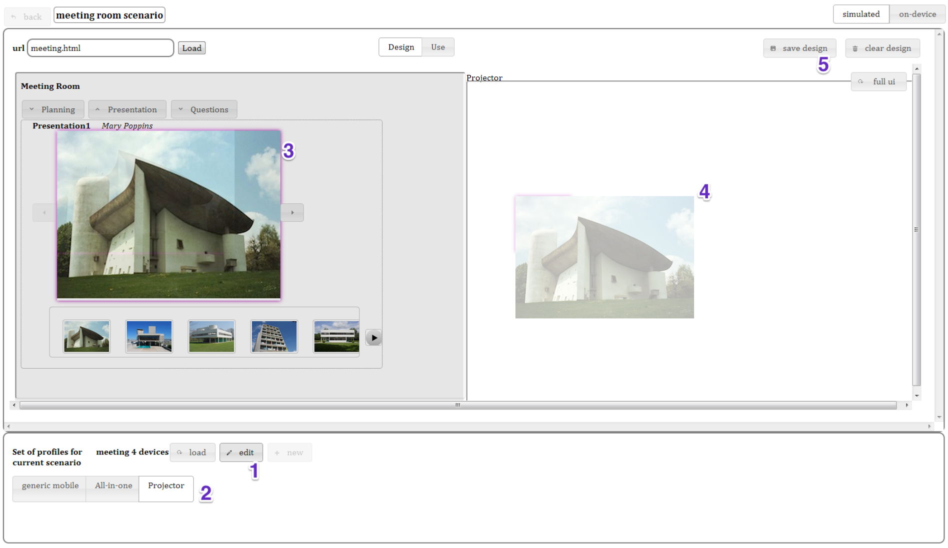


Figure 4.14: Design using drag & drop in simulated mode

Figure 4.15 shows the main screen in the simulated mode when all three profiles are open for design (1). By tapping inside each window (3-5) on top of a UI element a grey overlay pops up (6) enabling the user to remove the element, if it is considered redundant for the design. The screen is already cluttered displaying the reference UI on the left(2), the window for the mobile phone UI version (3), another one for the All-In-One (4) and another one for the projector (5). At this point, the user decides to change to the on-device mode (7) and thus include the target devices in the design.

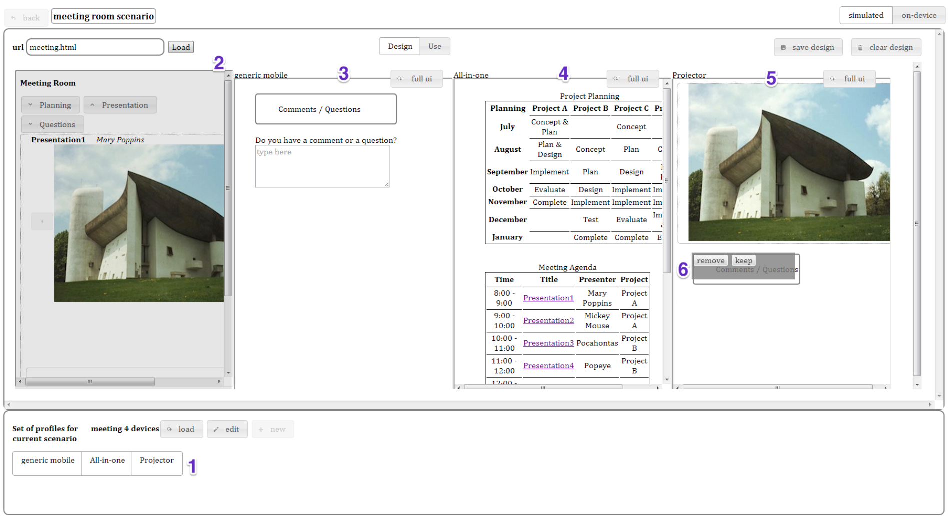


Figure 4.15: Design for multiple profiles in simulated mode - remove redundant elements

The main screen in the on-device mode (1) is displayed in Figure 4.16, showing now the profiles as rectangle areas where UI elements can get dropped.

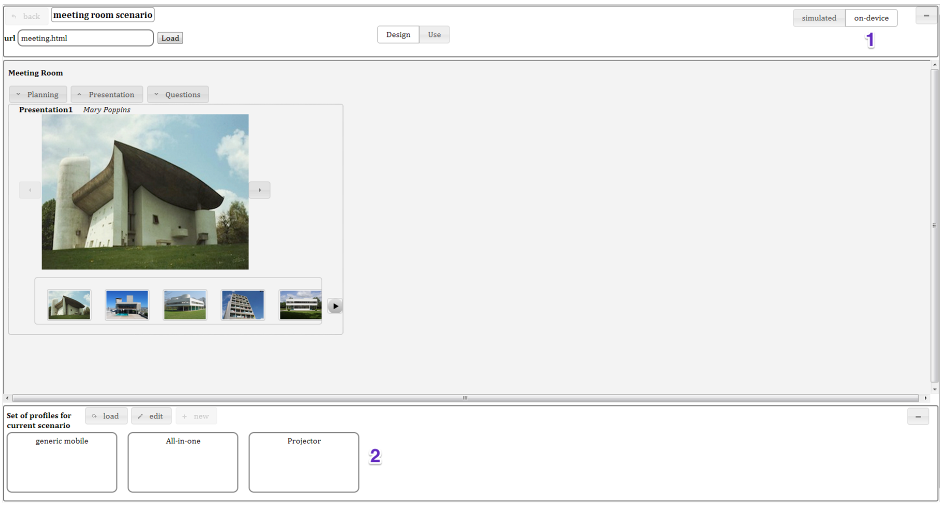


Figure 4.16: Main screen in on-device mode

At this point the peripheral devices, i.e. an All-In-One PC, a projector, a mobile phone and an iPad had joined the design process. The design process is consistent across simulated and on-device mode and thus we can see on the devices the designed UI versions for the existing profiles, as displayed in Figures 4.17, 4.18 and 4.19.

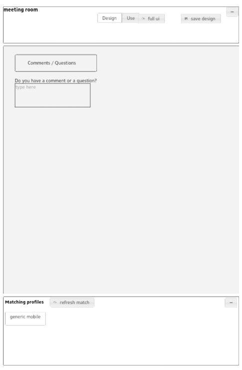


Figure 4.17: Mobile

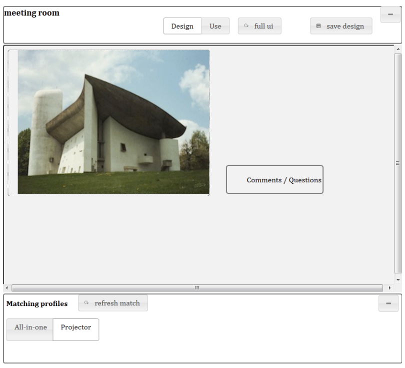


Figure 4.18: Projector

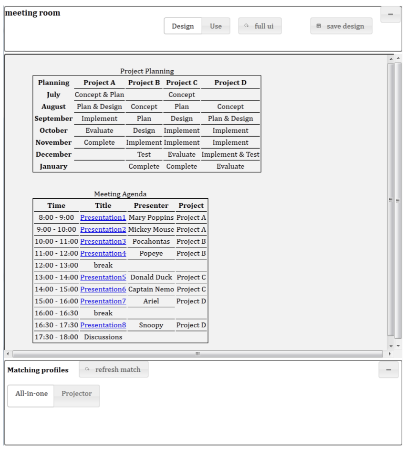


Figure 4.19: All-In-One

Yet, the profile for the tablet which is being handled by the presenter is still missing. In Figure 4.20 we can see how the designer refreshes the list of connected devices (1) and gets the indication for an iPad that had joined the distribution (2). Then the user decides to create the role of the presenter (3) through a pop-up dialogue similar to that of Figure 4.12. The iPad device (4) and presenter role (5) are dragged and dropped in the new profile area and added

(6) to the profiles list. The new list is saved (7) and the design can continue as displayed in Figure 4.21 or on the actual target devices.

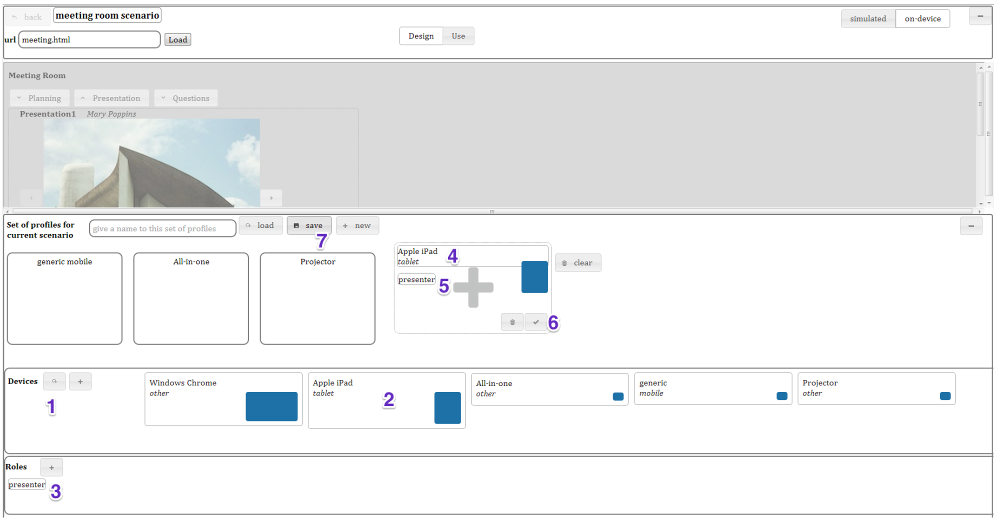


Figure 4.20: Detection of connected iPad and creation of respective profile

Figure 4.21 illustrates how the design is realized in the on-device mode. A UI element, here the whole presentation tab, can get selected and dragged (1) and dropped on the respective profile (2). The result of this design action is instantly displayed on the connected target device. This is displayed in Figure 4.22 (1). On the lower part of this figure, in the matching profiles area only the profile targeting the iPad is displayed. The system is able to match existing profiles with the type of the connected device and demonstrate only the ones that are related. Apart from dragging and dropping UI elements, the user could use the full UI (3) on the tablet to start the design and then remove redundant elements, as displayed before.

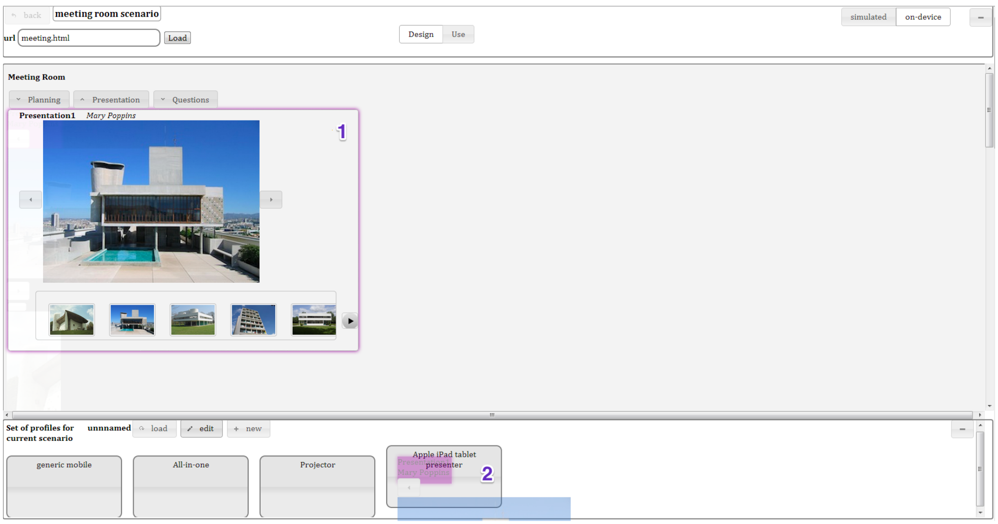


Figure 4.21: Desing in the on-device mode on the master UI

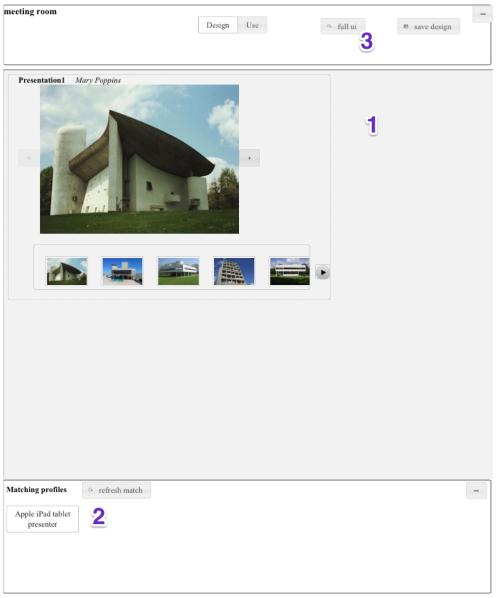


Figure 4.22: Design on the tablet

To interact with the UI the user has to change from design to use mode. As one can see in Figures 4.23 and 4.24, this change is synchronized across all involved devices (1). Once the user clicks on the next button (2) in Figure 4.23 the next slide gets displayed (3) on the iPad and on the projector (2), see Figure 4.24, as well. If the user is not satisfied with the design they can instantly change back to design mode for further editing.

Figure 4.23: Interact with the iPad UI in the on-device mode

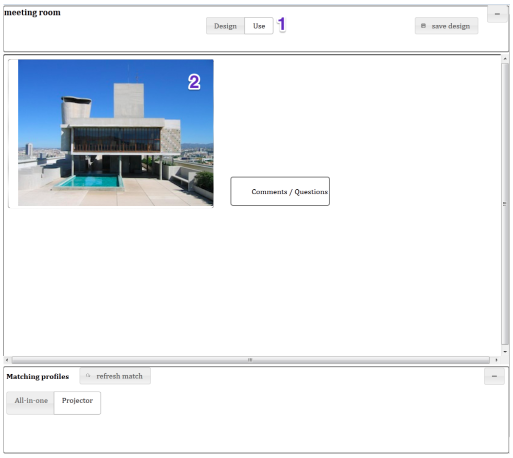


Figure 4.24: Interact with the projector UI in the on-device mode

4.4 Implementation

Here we describe the systems architecture, the technologies involved and further implementation details. XDStudio follows a client-server architecture and builds on top of the MultiMasher framework [6] in terms of the architectural model and the involved technologies.
In Figure 4.25 we can see the main components of the system and their interactions. The server supports the communication across the clients especially in the case where the on-device design mode is active. It also ensures that a consistent state is maintained across the involved devices and that all information is stored in the systems database. On the client-side users selection and firing of events are detected and communicated to the server, while also information pulled or pushed by the server is accordingly displayed.

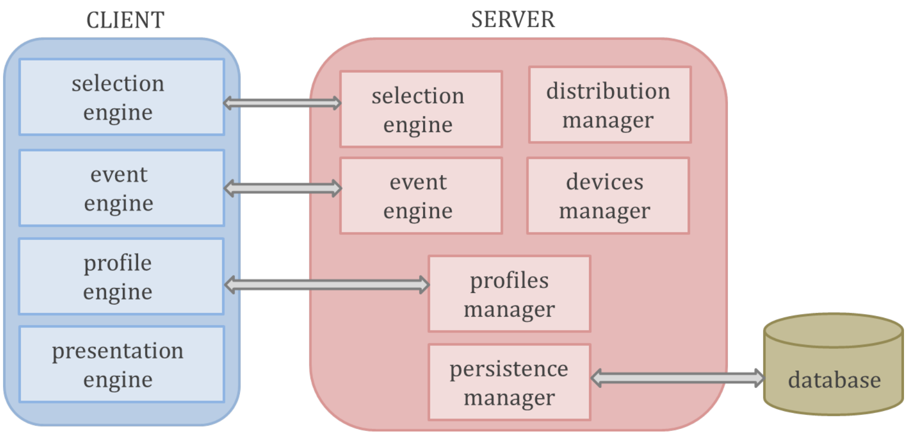


Figure 4.25: System Architecture

The persistence manager stores distributions, profiles, devices and roles in the database and retrieves this information when it needs to be updated. The device manager identifies connected clients and extracts information about the device type, operating system and browser. This information is used in two ways: either to create profiles for the identified device (type) or to display the profiles that match this device in the on-device mode. During the design process, the creation, display and selection of profiles on the client is handled by the profile engine. It gathers information about the created profiles, indicates the profile(s) for which the user is designing at any moment and displays the matching profiles in case the device is a peripheral one (on-device mode). On the server side, the profile manager receives information about created profiles from the client, merges it with existing information in the system, and forwards it to persistent storage through the persistent manager. The profile manager also informs the presentation engine about the currently handled profiles. This is necessary in order to display the design for these profiles on the client-side. The selection of UI elements in order to include them in the design is handled by the selection engine on the client side. This component informs the selection engine on the server-side about selected visible or hidden UI elements. The server-side selection engine merges this information with existing selections and informs the presentation engine about what to display. The overall creation and update of distributions is handled by the distribution manager, which in turn forwards updates to the database via the persistence manager. Event triggering on the client side is detected and replayed if necessary by the event engine, which transmits local DOM events to the server-side

event engine. This one is responsible for propagating these events to the involved devices ensuring a state consistency across the connected devices.
For the server-side implementation we used jQuery[1] on top of the Node js platform[2]. To store and retrieve data we made use of a MySQL Database which communicates with the server through a pure node.js JavaScript Client implementing the MySQL protocol[3]. To push and pull information between the server and the clients we selected Socket.IO[4], a cross-browser transport mechanism for real-time applications. On the server-side this JavaScript library runs for Node.js and on the client-side in the browser. To support real-time connectivity on every browser, it selects the adequate transport mechanism, primarily the WebSocket protocol but also other methods such as Adobe Flash sockets or JSONP polling, and keeps the API unaffected. Asynchronous utilities for the server and the browser are supported by the Async.js module[5].
The client-side of XDStudio is implemented using HMTL5, CSS3, jQuery and jQuery UI[6]. jQuery UI Touch Punch[7] is used to support touch interaction, an important feature when touch-devices such as tablets and smart phones are involved in the design process. The web-based approach of XDStudio makes the tool accessible using an entry point through a browser. On the main screen the user can load a web site, which will be used as the reference UI to get distributed. The display and handling of UIs in the tool is realized through the use of iframes. For the initial UI and for every UI targeting a profile separate iframes are employed. When in design mode, a transparent overlay is attached over the iframe displaying the initial UI. This disables the interaction with the actual website and allows the user to select elements without enabling them. The selected elements are identified based on the position of the cursor on the click event and are attached to the overlay right on top of the initial UI. The selection is supported for elements with an id, thus the closest parent element with an id is picked upon a click. The selected elements can then get dragged and dropped on an overlay on top of the iframe of the required profile (simulated-mode) or in the respective profile area (on-device mode). On the iframes of the involved profiles the UI is initially hidden and only dropped elements and their parents become visible.
In case the user selects to display the full version of the UI, then all UI elements, expect from those that are hidden by default, are displayed. An overlay over the iframe enables the user to select elements in the same way as before and remove them, i.e. hide them.
The selection of elements either to make them visible or to hide them for specific profiles is communicated to the server, which integrates them to previous selections and updates the UI version on the corresponding iframes or on the connected devices displaying the related profiles.
While in the use mode, the overlay is removed and DOM events on UI elements are detected and propagated through the server to other iframes or connected devices. This replaying of an event replicates the interaction across devices. An extension of this mechanism would allow the users to define for which UI elements or profiles they would like to remove the interaction.

[1] http://jquery.com/
[2] http://nodejs.org/
[3] https://github.com/felixge/node-mysql
[4] http://socket.io/
[5] https://github.com/caolan/async
[6] http://jqueryui.com/
[7] http://touchpunch.furf.com/

5 Evaluation

A user study was conducted to evaluate the design and usability of the tool experimenting with the two design modes: simulated and on-device. Our evaluation had multiple goals. We aimed to:

- identify the advantages and limitations of each mode
- spot what kind of distribution scenarios and tasks are better supported using each mode
- compare design- versus run-time.

5.1 Method

5.1.1 Test Organization

The evaluation of the tool was realized through a well-structured user study. The participants carried out the study one after the other. At first, every participant was given a short introduction to the system's goal and main functionalities. Before beginning with the specific tasks, every user was asked to fill a background questionnaire in order to identify the relation of the users to technology, interactive devices and experience in user interface design. Then a short written description of the use case scenarios and the tasks to be performed where given to the participants.
During the trial the participants were asked to use XDStudio in order to design the distributed UIs for two different scenarios. The scenarios used in the user study are based on the application scenarios presented in Chapter 3. For each of them the design task was repeated three times: once having available only the simulated design mode, another using only the on-device design mode and a third one where it was possible to switch from one mode to the other. Thus, in total every user had to perform six tasks. After each task the participants were asked to complete a questionnaire in order the evaluate the provided features of the tool they used, how easy they found the task and how efficient they felt. In the end, users were asked

to fill in a final questionnaire to give an overall grade to every task and to vote for the winner design mode -simulated, on-device and combined- for every scenario. Comments and other features to be supported were also provided by the participants on this last questionnaire.
No learning phase was included in the user study for the participants to get accustomed to the tool. To counterbalance the learning effect, the six different tasks were rotated and all twelve possible combinations were covered by the conducted user study. The permutations of the six tasks are presented in Appendix A.2. Changing the order of the tasks for every participant and assisting them by demonstrating the basic functionalities on the run was thus considered necessary. The study supervisor had mostly the role of an observer, but was also there to guide the participants when necessary.
Apart from the questionnaires, observations and comments were noted down by the study supervisor. These are presented in the results section later in this chapter.

5.1.2 Participants

The user study involved 12 participants (2 female), aged between 24 and 38 (median: 28). All of the users have and use regularly, i.e. minimum once a day, at least two interactive devices, such as a mobile phone and a laptop, and have little or medium experience in designing UIs. On a 1-7 scale (with 1 for novice and 7 for expert UI design skills) the average experience is 2.8 with a standard deviation of 1.1. Five out of the twelve users are PhD students and members of the GlobIS Group, while the other seven are ETHZ Master's level students. All participants were introduced to the tool and used it for the first time during the user study.

5.1.3 Test Case Scenarios and Tasks

Based on the scenarios presented in Chapter 3, the proposed test case scenarios for the user study of XDStudio are also taking place in a meeting room and in a classroom. The written instructions for every scenario, as they were given to the participants, are presented in Appendix A.1. For both scenarios the participants were asked to design the distribution of a user interface containing all necessary UI elements and functionalities.
During the first scenario the participants were asked to distribute the meeting room UI, see Figure 5.1, over an All-In-One PC, a projector, a tablet and a mobile phone.

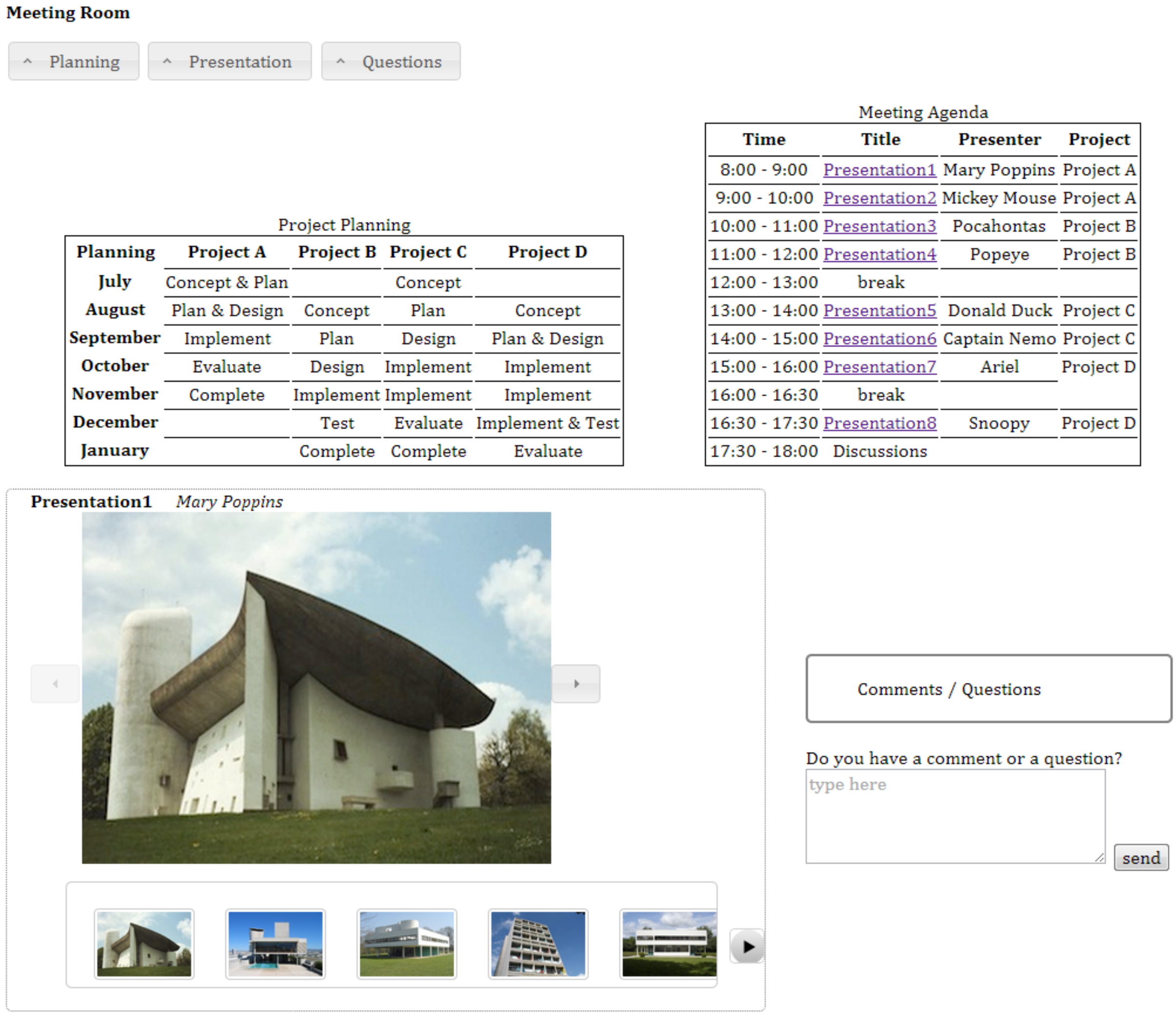


Figure 5.1: Meeting Room UI

On the projector the currently presented slide and comments and questions should be displayed. On the tablet the whole presentation UI component and existing comments and questions should be available. The mobile phone should display existing comments and questions and also provide a text-area for inserting new ones. Finally, the All-In-One should show the meeting agenda and project planning of the group that is attending the meeting. The distributed version of the UI is displayed in Figures 5.2, 5.3, 5.4 and 5.5.

Figure 5.2: Projector

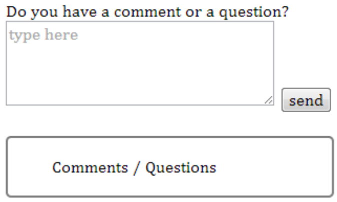


Figure 5.3: Smart phone

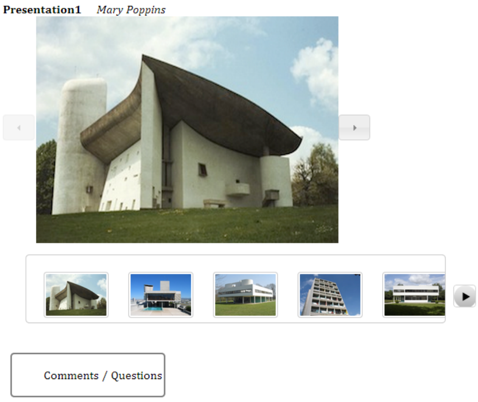
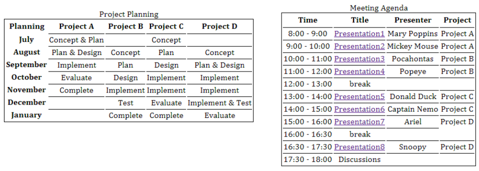


Figure 5.4: Tablet



Project Planning

Planning	Project A	Project B	Project C	Project D
July	Concept & Plan		Concept	
August	Plan & Design	Concept	Plan	Concept
September	Implement	Plan	Design	Plan & Design
October	Evaluate	Design	Implement	Implement
November	Complete	Implement	Implement	Implement
December		Test	Evaluate	Implement & Test
January		Complete	Complete	Evaluate

Meeting Agenda

Time	Title	Presenter	Project
8:00 - 9:00	Presentation1	Mary Poppins	Project A
9:00 - 10:00	Presentation2	Mickey Mouse	Project A
10:00 - 11:00	Presentation3	Pocahontas	Project B
11:00 - 12:00	Presentation4	Popeye	Project B
12:00 - 13:00	break		
13:00 - 14:00	Presentation5	Donald Duck	Project C
14:00 - 15:00	Presentation6	Captain Nemo	Project C
15:00 - 16:00	Presentation7	Ariel	Project D
16:00 - 16:30	break		
16:30 - 17:30	Presentation8	Snoopy	Project D
17:30 - 18:00	Discussions		

Figure 5.5: All-In-One PC

After completing the design task the user was prompted to check whether the distributed UI works as expected. As mentioned before, each participant had to repeat this design task under three different conditions:

- using only the simulated version; in this case the user had available a Windows PC
- working only with the on-device mode, where all target devices are included in the design process; an iPad, an Android phone, an HP TouchSmart, a Windows PC with two screens, one as the central authoring device and the second simulating the projector, were provided to the participants for this task
- having the option to switch at any moment from the simulated to the on-device mode

The classroom scenario revolves around a lecture where the collaborative method presented in section 3.2 is applied to teach students about electromagnetism and to encourage classroom interaction. The initial UI is displayed in Figure 5.6.

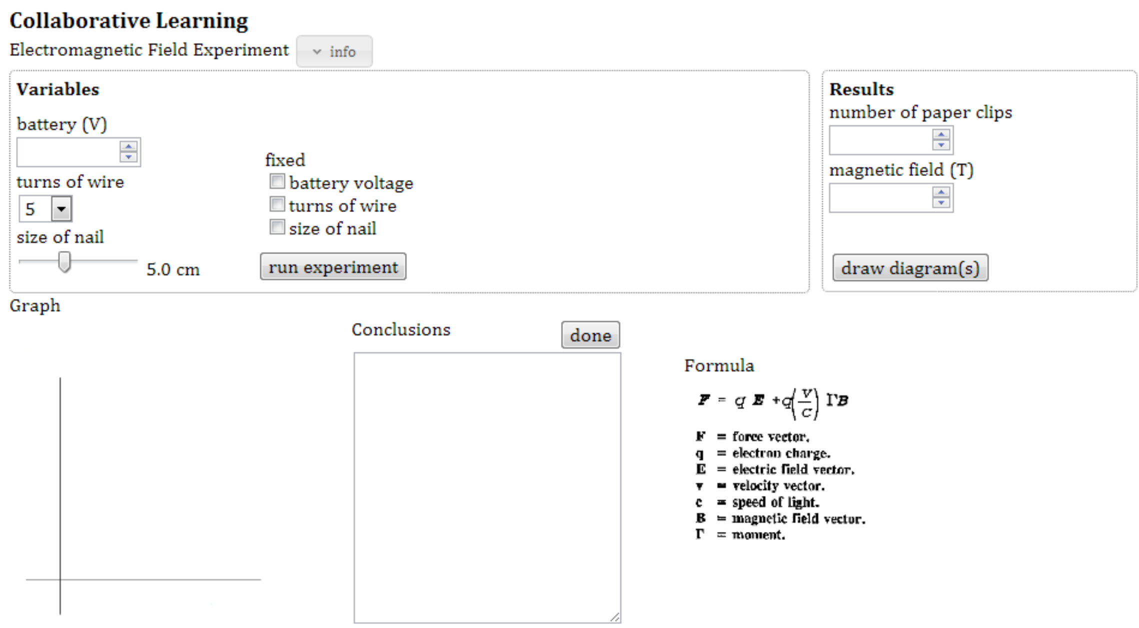


Figure 5.6: Classroom UI

In this scenario both teacher and students are using tablets. The teacher has full access to the application seeing everything apart from the header, the info section and the conclusion text-area. On the other hand, the students can see only one of the three independent variables on their UI version, run the experiment, gets results, draw a diagram for the calculated values and note down comments and conclusions. The distributed version of the classroom UI is presented in Figures 5.7, 5.8. 5.9 and 5.10. In this case again users are asked to design the distribution of an existing fully-functional UI, find out how the result of the design process looks like and check whether it operates correctly. Once more, the design, interaction, and confirmation phases were executed under the three conditions:

- handling the simulated mode alone; the participant was asked to complete the task on a Windows PC
- utilizing the on-device mode only; a Windows PC as the central authoring device and an iPad as the target device participating in the design process were provided to the participant
- alternating any time from one design mode to the other according to the participant's will

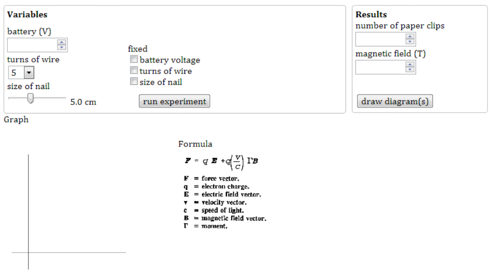


Figure 5.7: Teacher with tablet

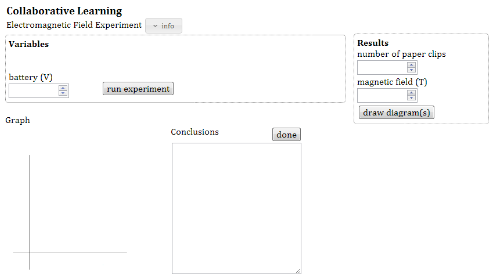


Figure 5.8: Battery group with tablet

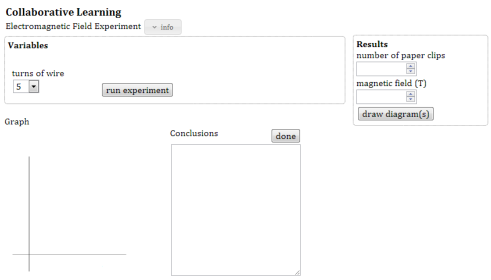


Figure 5.9: Wire Group with tablet

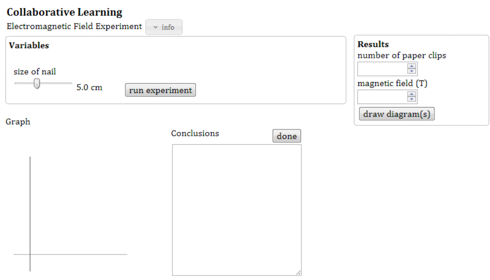


Figure 5.10: Nail group with tablet

As discussed also in 3.3, each of the test cases follows a different approach aiding us to evaluate different aspects of XDStudio. The meeting room case is a device-centric scenario on a complex UI with multiple hidden elements, that are distributed over different devices but are still interconnected. On the other hand, the classroom scenario is a role-centric test case, where a simple UI is slighlty altered for every UI view, while the different UI views are not interconnected. We asked the participants to experiment with the two selected scenarios in order to see how different cases are handled with XDStudio, how the device- and role-factors are perceived and handled and which design mode and tools apply better for each case.

5.2 Results

To complete the aforementioned test cases the participants were expected to load the respective UI on the dedicated area of the tool, create the profiles, design the distributed UI version for each of them and finally interact with them to verify the result. All of the participants completed successfully all six tasks in 1 to 2 hours including the time spent to introduce the tool, fill in the questionnaires and provide feedback. The exact execution times are presented and discussed in the following section 5.2.2.
Users were invited to fill in questionnaires after every task. They were asked to rate the key features of the tool in a scale from 1 to 7 (with 1 standing for the most negative and 7 for the most positive score), evaluate how easy they found each task and how efficient they felt.

Apart from the users ratings, a logging mechanism automatically recorded the time needed to complete every task and the individual times spent using simulated and on-device mode, when both available. These measurements in combination with the ratings for each mode may indicate the preferences of the users for one mode or the other. Through observation and user feedback we also noticed which tasks participants preferred to perform on each mode. In the following sections we present the average of the users ratings and the times and we correlate these numbers with the related user feedback. The different aspects are presented and discussed in turn.

5.2.1 Design Modes

Based on the votes for the preferred design mode in each scenario, the simulated one was rated higher compared to the on-device and the combined one. About half of the users picked the simulated version, but there were also users that voted for the combined or the on-device mode. The respective percentages are displayed in Figure 5.11 for the meeting room scenario and in Figure 5.12 for the classroom scenario. This reveals on one hand that the simulated mode was more solid for supporting the users in the given tasks, but on the other hand that the on-device mode and even more the combination of both modes may have positive effects and be useful for a set of users.

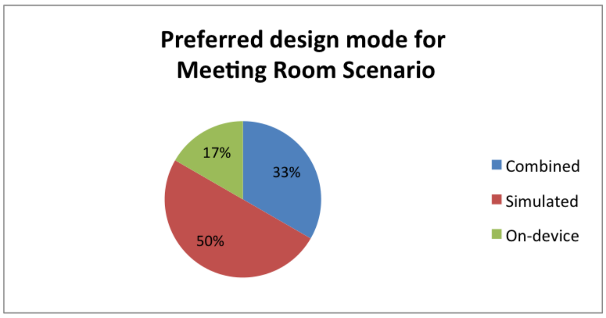


Figure 5.11: Preferred Design Mode for Meeting Room Scenario

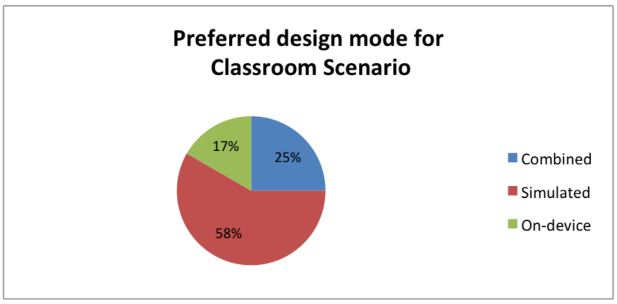


Figure 5.12: Preferred Design Mode for Classroom Scenario

Comparing these two graphs, one could argue about the effect of the scenario on the preferred design mode. We can see that the on-device percentage stays the same in both cases, yet the simulated mode loses in percentage for the device-centric meeting room scenario. Thus, the

option to switch between simulated and on-device mode at any time might be more appreciated for scenarios similar to the meeting room, where many different types of devices are involved and used in combination.
In Figures 5.13 and 5.14 we present the overall score users gave after each task for the available design mode, for how efficient they felt doing the task and for how easy it was for them. The first figure applies to the meeting room scenario, the second one to the classroom scenario.

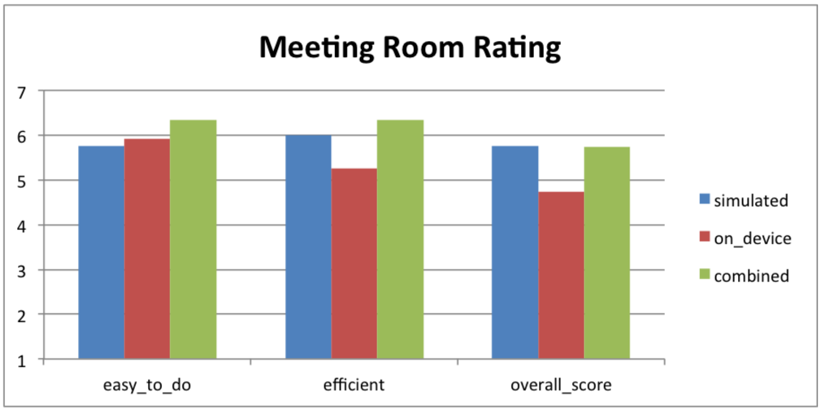


Figure 5.13: Ratings for Meeting Room Scenario

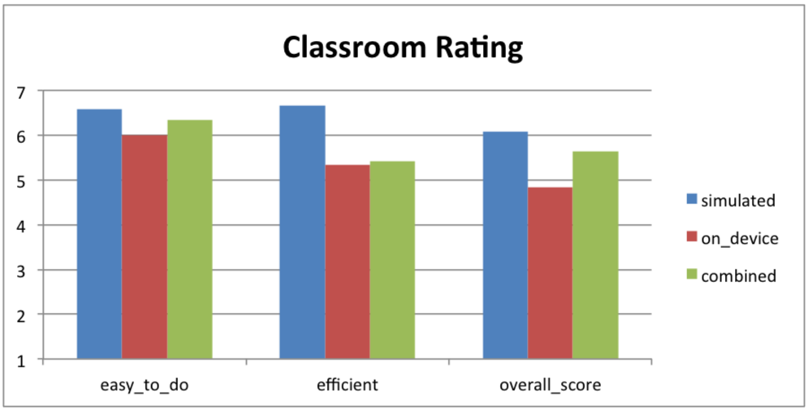


Figure 5.14: Ratings for Classroom Scenario

As for the overall grade that users gave to each task concerning the available design mode, how easy they found the task, and how efficient they felt we notice that:

- For the meeting room the simulated and combined mode have the same overall score but users found it easier to do the task and felt more efficient when both simulated and on-device modes (combined) where available.

- The on-device mode received a relatively lower score. However at the meeting room scenario users still found it quite easy to do the task, giving a slightly higher score compared to the winner simulated mode.

- For the classroom scenario the overall grade of the simulated mode exceeds the other two, leaving the combined in the second place and the on-device last. The same applies for how efficient the users felt under the three conditions and how easy they found completing the task.

The first two observations somewhat contradict the preferred design mode for the meeting room scenario, i.e. the simulated one. As we will also explain later, changes and improvements in the on-device mode as suggested by the users might increase their preference to both the on-device and combined modes. The third finding reveals that in the role-centric scenario, where small adaptations of the initial UI were necessary and only one device type was involved, the simulated mode felt more intuitive and suitable to the users.
In Figure 5.15 users ratings for switching between design modes when both available, are presented.

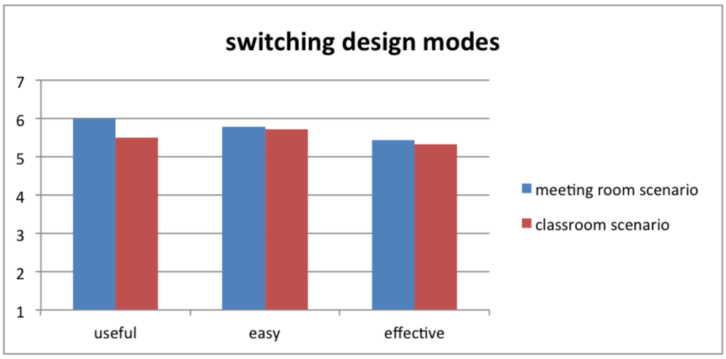


Figure 5.15: Ratings for switching design modes when both available

Although the majority of the users did not vote for the combined modes, the overall score for alternating from one mode to the other when both available, was positive. Especially for the meeting room scenario the feature was considered quite useful (avg: 6.0, sd: 1.9), easy-to-use (avg: 5.7, sd: 1.8) and effective (avg: 5.4, sd: 1.8). Slightly lower scores (avg: 5.5, sd: 2.3 - avg: 5.7, sd: 2.2 - avg: 5.3, sd: 2.3 respectively) were given to the same feature for the classroom scenario, which either way had more supporters in the simulated mode.

5.2.2 Times

To getter a better picture of how the three different design conditions affected the efficiency of the users designing each distribution we present the execution times. In the Figure 5.16 one can see the average time the participants needed to complete each task.

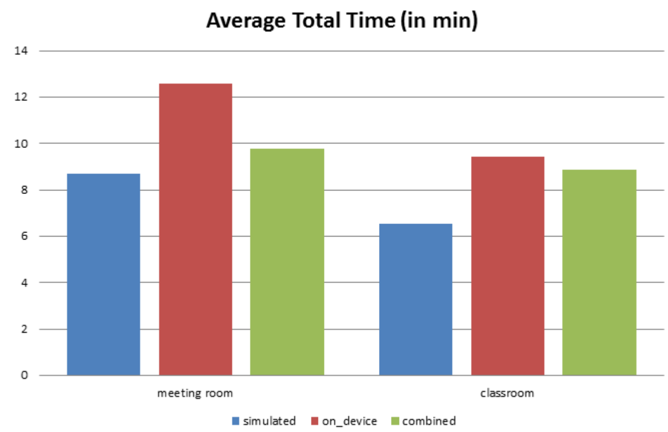


Figure 5.16: Average total times for execution of each task

For the meeting room scenario users needed slightly more time when in the combined mode compared to when in simulated mode. If we consider the time users needed to connect the devices and switch from one to the other we can notice that it is almost the same for both cases. Yet, in absolute times the simulated one performs better. The on-device mode was rather time-consuming taking about 35% more to complete the task compared to the fastest mode, i.e. the simulated.
In the classroom scenario, again the simulated mode was the quickest one, while the combined and on-device modes demanded almost the same amount of time until the completion of the task.
Comparing times between scenarios we can see that the meeting room demanded more time. That was expected since it involves more devices to be handled and a more complex UI, where some of the UI elements to get distributed were hidden. This required of the user to change between the design and use mode of the tool in order to interact with the UI, reveal hidden elements and then select them and assign them to the respective target UI view. On the other hand, the classroom scenario involved only one device and a much simpler UI distribution, where only certain elements were missing from each distributed view.
An interesting observation is that when both modes were available, the total time for the meeting room scenario was slightly longer than when only the simulated mode was used, while for the classroom scenario it increased almost as much as when the on-device mode alone was available.
Figure 5.17 demonstrates the total execution time per task when both design modes were made available to the user. It also displays the partial average times that users spent in the simulated and in the on-device mode.

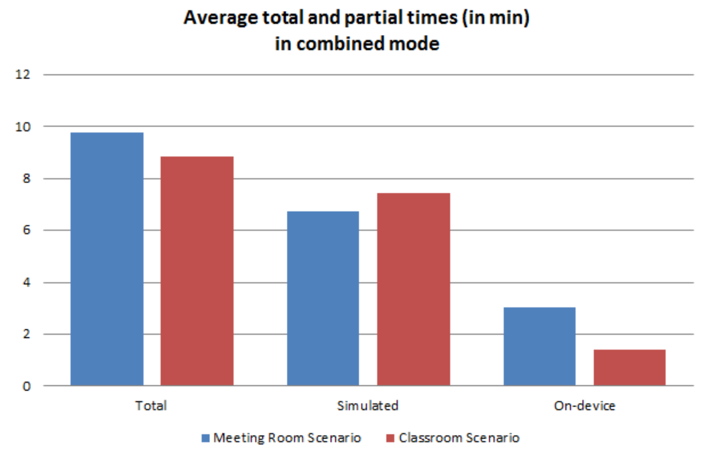


Figure 5.17: Overall and partial times for task execution when both modes available

Through this graph we can see again that the meeting room scenario demanded more time compared to the classroom scenario. Yet, it is interesting that the time spent on the simulated mode is longer for the classroom scenario than for the more time-consuming meeting room. This observation shows once more that the simulated mode was mostly used and appreciated by the participants for the classroom scenario. As for the on-device mode, this was mostly used in the device-centric meeting room scenario compared to the role-centric classroom scenario. Figures 5.18 and 5.19 illustrate the partial times spent in simulated and on-device modes, when the user could switch between them, in relative percentages for each scenario. We can see that users spent in the on-device mode twice as much time when designing for the meeting room as when designing for the classroom.

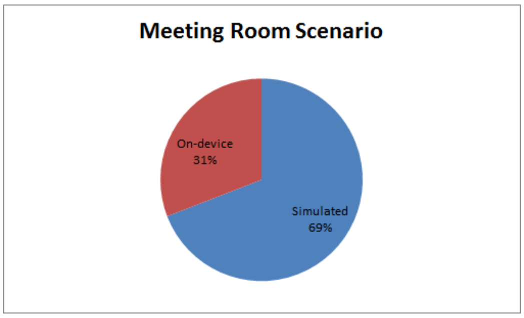


Figure 5.18: Partial Times - Meeting Room

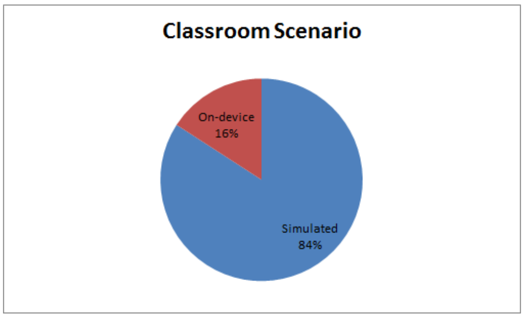


Figure 5.19: Partial Times - Classroom

Based on the results presented in this section, we may argue that in cases where multiple devices are included in the distribution and are available at the design process, and when the initial UI design is more complex, then the combination of the two design modes can prove beneficial. On the other hand, in cases when small changes need to be done on the initial design, i.e. remove certain elements from the initial interface, and only one device is involved, then the simulated version performs better. We relate this observation with users feedback during the classroom scenario, who while using the on-device mode were not sure which profile was selected on the iPad and preferred to go on the simulated version for having a quicker and better overview. What is more, almost all of them stated that performing the whole design on a relatively small-screen device, such as the iPad, was getting troublesome, especially when it handled removing small-sized elements. This leaves open an area of future work for XDStudio, where the adaptation of the tool for different types of devices could be investigated.

5.2.3 Main Features and Design Tools

In this section we present the users feedback, both scores and comments, regarding the key features and design tools of XDStudio.

One of the main properties of the tool is the ability to switch between design and use mode, which as explained before lets the users select whether they want to distribute or interact with the interface. The scores for this feature are displayed in Figure 5.20.

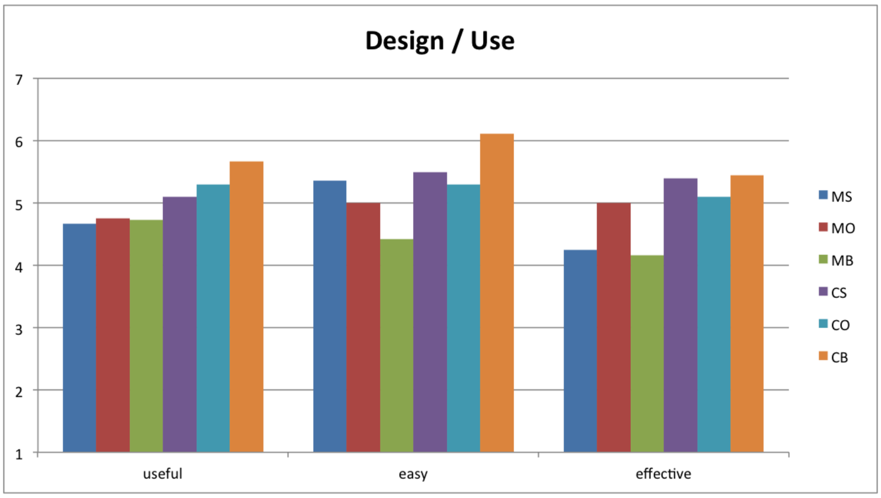


Figure 5.20: Average Rating for Use/Design feature

This feature obtained moderate ratings, especially for the meeting room scenario. There the user had to go into the use mode to interact with the UI and reveal hidden elements and then return to design mode and assign them to the respective profile. This demanded an extra effort that most users stated they would preferably avoid. Better scores were achieved for the classroom scenario, where the switch between design and use was only performed to check the final distributed UI version. In general, the user study revealed that most of the participants would prefer to have this functionality implicit in the system.

As for the creation of devices, roles and selection of profiles we have noticed that users ratings did not considerably change based on the task. Thus, we present an overall average value for each feature in Figures 5.21, 5.22 and 5.23. We should however mention that four out of twelve participants stated that they prefer to create profiles in the on-device mode, where information about connected devices saves times and effort, since they can avoid manual creation of custom devices. Also, role creation was rated only for the classroom scenario, since this feature was not used for the meeting room test case.

We can see that the users have a strong positive opinion regarding these three features. They gave at least a 6.0 as an average for finding these characteristics helpful, easy to use and

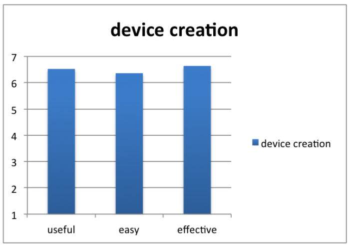


Figure 5.21: Device Creation Average Rating

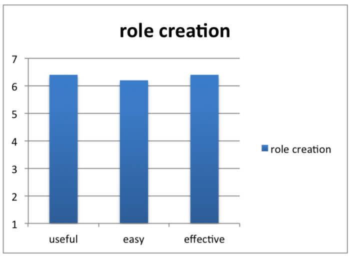


Figure 5.22: Role Creation Average Rating

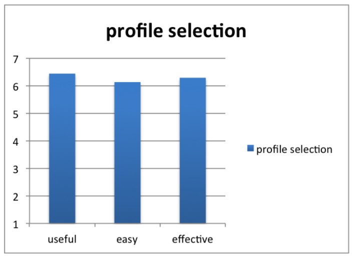


Figure 5.23: Profile Selection Average Rating

effective. A suggestion about the profiles selection was made by three out of twelve users. They commented that they would like to be able to change the order of the profiles, and align the profile button underneath the corresponding window in the simulated mode.

Concerning the design tools of XDStudio, we present the scores for the element selection and drag and drop features in all six tasks. The full/empty UI property is presented only for the classroom scenario since it was hardly used in the meeting room case. In Figures 5.24 and 5.25 the scores for element selection and drag and drop are displayed.

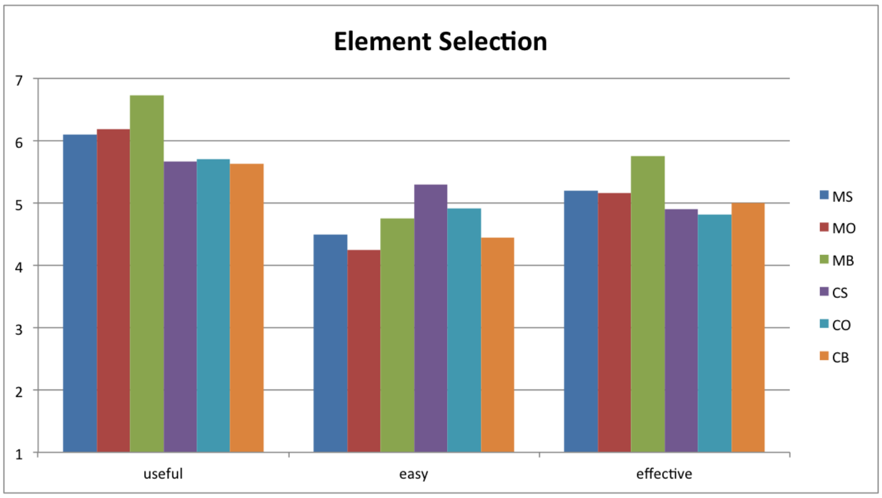


Figure 5.24: Average Rating for Element Selection

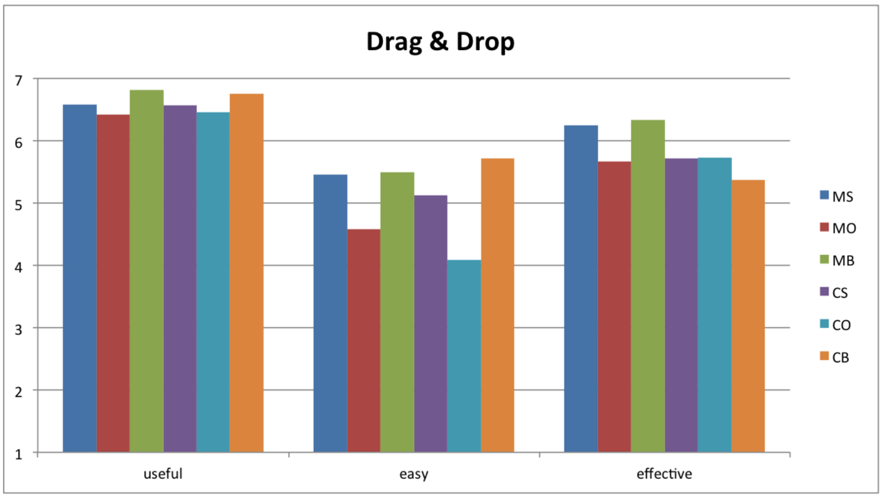


Figure 5.25: Average Rating for Drag & Drop feature

For the meeting room scenario the element selection and drag and drop were considered very useful. In this case users found it more efficient to design that way compared to starting from the full UI version, where many hidden elements had to be discovered through toggling visibility in the use mode, and removed. Also, both features were considered quite effective. Yet, participants expressed a lower satisfaction regarding the ease of use of these tools. As stated by nine out of twelve users, multiple selection or selection by area and dragging and dropping of multiple elements could make these features easier to use and the overall design process faster.

For the classroom scenario, element selection was not considered that useful, since most people worked with the full UI version. Yet, both this feature and drag and drop received good ratings for their effectiveness, as they were used for corrections when particular elements from the full UI version had mistakenly been removed.

Regarding the drag and drop specifically, one can notice that in the on-device mode in both scenarios users had some difficulties. Participants expressed their desire for better feedback upon drag and drop of elements in this design mode and possibly better support for large-sized UI components.

Figure 5.26 demonstrates the ratings for the full/empty UI feature, which was proven significantly useful in the classroom scenario. Every participant turned to this tool for performing the design, since using the full version of the interface and further removing needless elements was faster and more efficient than dragging and dropping the required ones. Here we would like to mention that some of the participants expressed the uncertainty about whether full UI refers to the visible UI or to the whole HTML page.

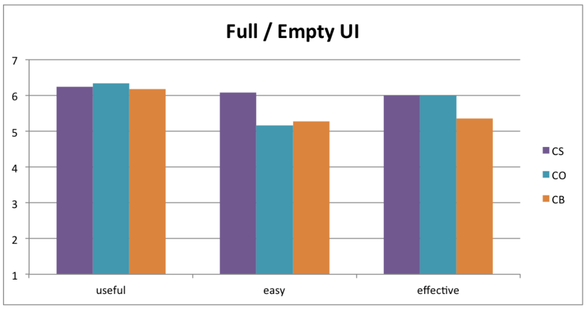


Figure 5.26: Average Rating for Full/Empty UI feature in the Classroom Scenario

5.3 Final Remarks

Observing the users, we noticed that when having the option to alternate between simulated and on-device, eight out of twelve performed the design in at least one of the scenarios using the simulated mode and switched to on-device mostly to check how it looks like on-the-run. Yet, six out of twelve participants turned to the on-device mode in order to do partial or the whole design in at least one of the scenarios. According to those users, we identified some weak points in the design and current implementation of the on-device mode. Four out of six stated that they would prefer to:

1. have better feedback when dragging and dropping elements on the profiles area
2. be able to select the currently handled profile from the master device as well
3. have an indication of the currently handled profile on the master device
4. get an indication about which elements are already selected for a profile in the profiles area on the master device, for instance through thumbnails.

Also, one user stated that they would prefer to do the design using the current version of the simulated mode, but also see the changes on the connected target devices. Other users, stated that they would have rather worked with the on-device mode more, if the aforementioned features were also supported. This remains as future work to be implemented and investigated.
Finally, considering the nature of the scenarios one should notice that there is a great emphasis on the design of a precisely defined distribution and less on checking whether the distributed version of the UI fulfils its scope. For this reason, all users concentrated on performing the design tasks and only shortly tested the distributed interface on-the-run. To thoroughly investigate how users would interact with the tool to perform a less precisely defined distribution and debugging the design, the user study should be enriched with additional tasks more concerned with testing and debugging. We leave this area open for discussion and further investigation and believe it will lead to interesting findings for this tool and for improvement.

6 Conclusion

This thesis has presented XDStudio, a UI editor for distributing web-based user interfaces in multi-device environments. The proposed tool offers two interconnected design modes: a simulated mode, where the design and testing takes place in one central authoring device through simulation of the target devices and an on-device mode, where the target devices actively participate in the design and testing process. The web-based approach enables the tool to support cross-platform design and instant checking of the distributed user interface. The basic idea behind the project is to design a simple UI editor for both experts and non-experts users and identify the benefits and limitations of the two suggested design modes. A user study revealed both strong and weak traits of the current design and implementation and motivates future work. In this last Chapter, we discuss limitations to be handled and open issues to be investigated.

6.1 Limitations

XDStudio has been implemented in the framework of this thesis as a proof of concept tool for investigating the effectiveness of the proposed UI editor. The pilot tool enabled us to successfully conduct a user study and look into the strengths and weaknesses of the available design modes. Yet, some of the design ideas and proposed features are not supported by the current version of the editor. Here we discuss the main limitations of XDStudio based on the existing design and implementation, as they were revealed through the user evaluation:

- Users frequently expressed their desire to select multiple UI elements at a time and be able to assign them to multiple profiles as well. The single element selection was considered useful, but not sufficient for the design process. Selecting only elements with an id is a current technical limitation, that restricts the freedom of the designer. Extending the element selection is expected to increase the user satisfaction and effectiveness of the design process.

- When designing in the on-device mode, most users were not sure about whether UI elements were effectively assigned to a profile. More reassuring feedback is necessary to make the design process more solid especially in this design mode. A combined parallel version of the two modes was also expressed as a wish by the users.
- Apart from deciding which UI elements are available on each profile, further editing should be possible. Adaptation of the selected UI elements in terms of size or actual position would give more freedom to the designer and enable them to reach a better result. Combining this work with the CrowdAdapt framework [17] could provide this extra feature to the designer.
- Suggestion tips for replacing UI elements according to the target device or platform is expected to increase the quality of the final distributed UI. This feature could be supported using different strategies, such as the graceful degradation and progressive enhancement [4]. In Figures 4.5 and 4.6 a 'replace with' area is suggested for providing this additional functionality.
- Indication of interconnected elements both in a specific device UI and across connected devices is one of the features that could make the distribution of a UI easier and more intuitive saving time during the debugging process. Also, removing existing interaction between UI elements and enabling it when necessary is one element that the tool currently does not support, but is considered necessary to support more scenarios and user needs.
- The need to switch between design and use mode should be minimized and ideally made implicit in the system. According to the users, the change between the two modes should only be done for testing the distributed UI.
- Propagation of DOM events shall be enriched to support a greater variety of events and ensure the state consistency across devices under any condition.
- A greater selection of device types for the user to define would make the creation of profiles more precise and would better support the matching of profiles on the target devices, while designing in the on-device mode.
- Loading and further editing existing distributions or existing sets of profiles is one of the main features that should be implemented for this tool to be more flexible and allow designers to save time and effort. For now all information is persistently saved in the database. Replaying it in the tool would make this information valuable for future designs and required adaptations.

6.2 Future Work

As already discussed, the current limitations remain to be handled in a future version of the tool and further investigated through dedicated user studies.
Other issues that stay open as future work are:

- Evaluation of the tool for other scenarios, including these where the distributed UI version is not clearly pre-defined. Through such a user study one could investigate how the tool supports the user in deciding what to include in every view of the distributed UI. This is expected to increase the need for instant checking and debugging during the design process. Interesting findings could be revealed regarding how the two design modes, simulated and on-device, support the user in such a case.

- Further evaluation of the tool for other types of tasks could also lead to interesting findings and help re-design the DUI editor.
- More detailed comparisons and evaluations against other multi-device tools is an area of further research for identifying more precisely the contribution of XDStudio in the field.
- Other ways of interacting with the tool, such as with gestures or voice, should be investigated. Such interaction methods could improve the context awareness of XDStudio and could make the design process more intuitive. Yet, this should be tested via a respective user study.
- Involving multiple existing user interfaces in the design would be another interesting area of experimentation. One would then be able to see how users mix UI elements from different websites to build a single distributed UI. Defining interactions across UI elements of different UIs would be another challenge to handle. An approach, such as the one presented by the MultiMasher framework [6], could be used in this direction.
- Support for device matching on deployment of the DUI and adaptation in case the expected set of devices (types) are not available should be explored. Future work to support this extra feature can be based on an extension of the Context Matcher and the Adaptation Rules at the core of XCML, presented by Nebeling et al. [15].
- Participation of multiple users in the design process of a DUI is another area that remains open to investigate. Collaborative editing introduces interesting design issues to address and to explore with. As a crowd-sourcing tool, CrowdAdapt [17] handles collaborative editing and could thus be seen as a starting point in an attempt to support multiple users at design-time.

Bibliography

[1] L. Cardelli. Building user interfaces by direct manipulation. In *Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, UIST '88, pages 152–166, 1988. ISBN 0-89791-283-7.

[2] D. Dearman and J. S. Pierce. It's on my other computer!: computing with multiple devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 767–776, 2008. ISBN 978-1-60558-011-1.

[3] P. Dillenbourg and F. Hong. The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning*, 3, 2008.

[4] M. Florins and J. Vanderdonckt. Graceful Degradation of User Interfaces as a Design Method for Multiplatform Systems. In *In IUI 04: Proceedings of the 9th international conference on Intelligent user interface*, pages 13–16. ACM Press, 2004.

[5] G. Ghiani, F. Paternò, and C. Santoro. Push and pull of web user interfaces in multi-device environments. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, 2012.

[6] M. Husmann, M. Nebeling, and M. C. Norrie. MultiMasher: A Visual Tool for Multi-Device Mashups. In *Proceeding of 13th Intl. Conf. on Web Engineering (ICWE 2013 Workshops)*, 2013.

[7] H.-C. Jetter, M. Zllner, J. Gerken, and H. Reiterer. Design and Implementation of Post-WIMP Distributed User Interfaces with ZOIL. *International Journal of Human-Computer Interaction*, 28(11):737–747, 2012. doi: 10.1080/10447318.2012.715539.

[8] S. K. Kane, A. K. Karlson, B. R. Meyers, P. Johns, A. Jacobs, and G. Smith. Exploring cross-device web use on PCs and mobile devices. In *Human-Computer Interaction–INTERACT 2009*, pages 722–735. Springer, 2009.

[9] J. Lin and J. A. Landay. Employing Patterns and Layers for Early-Stage Design and Prototyping of Cross-Device User Interfaces, 2008.

[10] J. Meskens, J. Vermeulen, K. Luyten, and K. Coninx. Gummy for multi-platform user interface designs: Shape me, multiply me, fix me, use me. In *In Proceedings of AVI08*, 2008.

[11] J. Meskens, K. Luyten, and K. Coninx. Jelly: A multi-device design environment for managing consistency across devices. In *In Proc. AVI10*, pages 289–298. ACM, 2010.

[12] J. Meskens, K. Luyten, and K. Coninx. D-Macs: building multi-device user interfaces by demonstrating, sharing and replaying design actions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 129–138, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5. doi: 10.1145/1866029.1866051.

[13] B. Moggridge. *Designing Interactions*. The MIT Press, 2006.

[14] B. Myers, S. E. Hudson, and R. Pausch. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 7(1):3–28, Mar. 2000. ISSN 1073-0516.

[15] M. Nebeling, M. Grossniklaus, S. Leone, and M. C. Norrie. XCML: providing context-aware language extensions for the specification of multi-device web applications. *World Wide Web*, 15(4):447–481, July 2012. ISSN 1386-145X.

[16] M. Nebeling, M. Speicher, and M. Norrie. W3touch: metrics-based web page adaptation for touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2311–2320, 2013. ISBN 978-1-4503-1899-0.

[17] M. Nebeling, M. Speicher, and M. C. Norrie. CrowdAdapt: enabling crowdsourced web page adaptation for individual viewing conditions and preferences. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '13, pages 23–32, 2013.

Appendices

A
Test Cases

A.1 Scenario Cards

Meeting Room Scenario

The group members of a lab are having a meeting where they discuss about current and future projects, while everyone also presents an overview of their work and future plans. The meeting takes place in a room equipped with a projector and an All-In-One PC whereas the presenter uses a tablet and every group member their smart phone. During the meeting all these devices are collaborating to support the overall process.

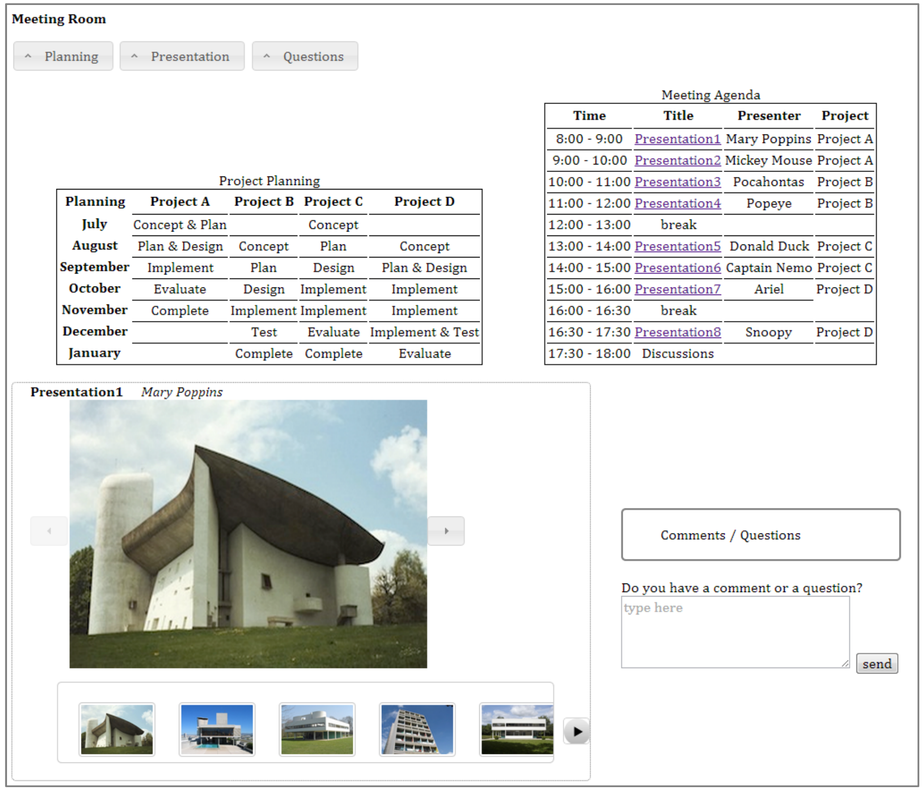
Your task

is to design the distributed version of this user interface according to the Device-based UI Distribution table below. Design the distribution for this scenario and test whether it works as expected.

Device-based UI Distribution

Projector	Current slide & Comments / Questions
All-in-one	Project planning & Meeting Agenda
Tablet	Presentation & Comments / Questions
Smart phone	Insertion of comment/question & Comments / Questions

Full version User Interface



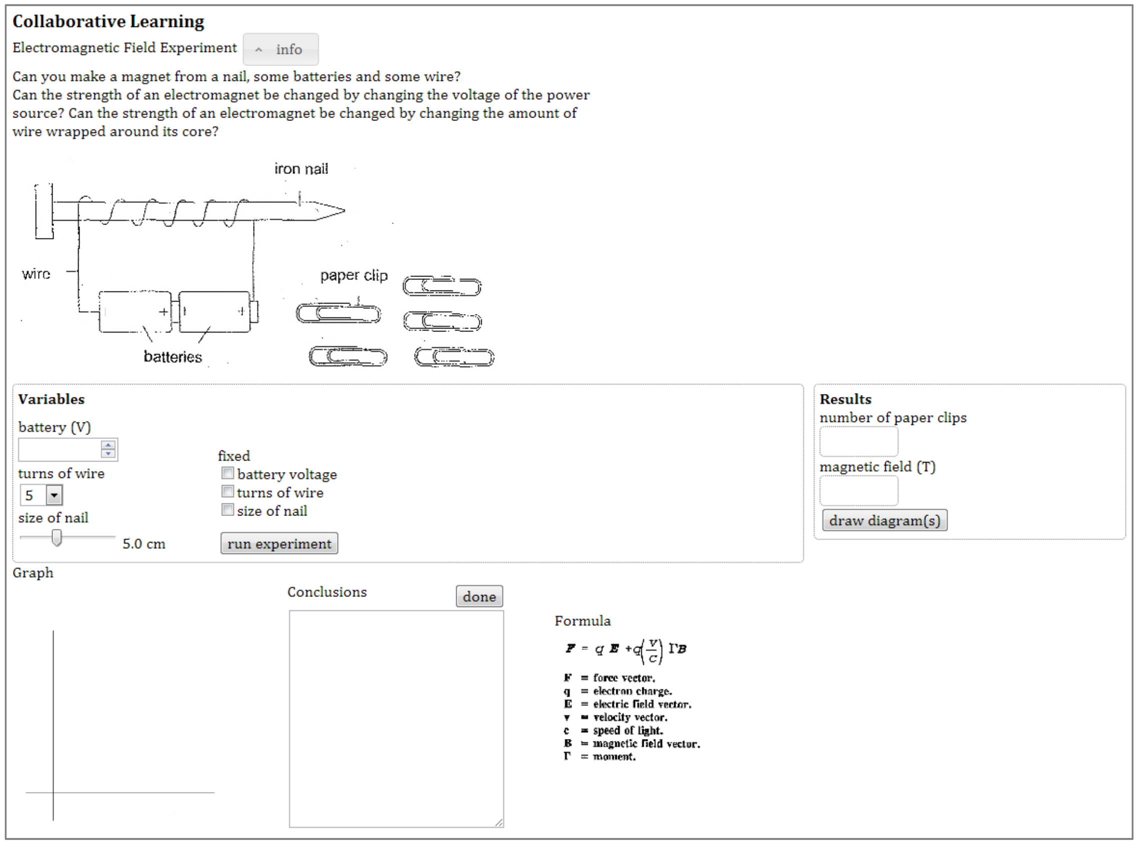
Classroom Scenario

A teacher uses a collaborative learning method to foster interaction across students while they learn about electromagnetism. The class is split into three groups and each group can affect only one of the three independent variables of an experiment. Through the experiment and discussion students are asked to come to a common conclusion about how all three variables affect the result of the experiment.

Your task

is to design the distributed version of this user interface according to the Role-based UI Distribution table below and considering that both students and teacher are using tablets. Design the distribution for this scenario and test whether it works as expected.

Full version User Interface



Role-based UI Distribution

Teacher	Variables (whole component) Results (whole component) Graph and Formula
Battery group	Title & Info Battery Variable & Run button Results (whole component) Graph & Conclusions
Wire group	Title & Info Wire Variable & Run button Results (whole tab) Graph & Conclusions
Nail group	Title & Info Nail Variable & Run button Results (whole tab) Graph & Conclusions

A.2 Task Rotation

User	Tasks in order of execution					
1	MS	MO	MB	CS	CO	CB
2	MO	MB	MS	CO	CB	CS
3	CS	CB	CO	MS	MB	MO
4	MB	MS	MO	CB	CS	CO
5	CB	CO	CS	MB	MO	MS
6	MS	MB	MO	CS	CB	CO
7	CO	CS	CB	MO	MS	MB
8	CS	CO	CB	MS	MO	MB
9	MO	MS	MB	CO	CS	CB
10	CB	CS	CO	MB	MS	MO
11	CO	CB	CS	MO	MB	MS
12	MB	MO	MS	CB	CO	CS

M: Meeting Room
C: Classroom

S: Simulated
O: On-device
B: Both

Figure A.1: Execution Order of tasks - rotation to counterbalance learning effect

B User Study Questionnaires

B.1 Background Questionnaire

Questionnaire

Gender: Female ☐ Male ☐

Date of Birth: _______________

What kind of electronic devices do you own?

☐ Mobile Phone ☐ Tablet ☐ Laptop ☐ Desktop ☐ Other: ____________________

How often do you use them?

Check the respective box

	Once a month	Once a week	Several times a week	Once a day	Several times a day
Mobile Phone	☐	☐	☐	☐	☐
Tablet	☐	☐	☐	☐	☐
Laptop	☐	☐	☐	☐	☐
Desktop	☐	☐	☐	☐	☐
Other: _______________	☐	☐	☐	☐	☐

What do you use them for?

Check the respective box

	Communication	Work	Entertainment	News	Other (please fill in)
Mobile Phone	☐	☐	☐	☐	
Tablet	☐	☐	☐	☐	
Laptop	☐	☐	☐	☐	
Desktop	☐	☐	☐	☐	
Other: _______________	☐	☐	☐	☐	

When do you use multiple devices at the same time? Why?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

What experience do you have in designing user interfaces?

Novice ☐☐☐☐☐☐☐ Expert

B.2 Post-Task Questionnaire

Post-task Questionnaire

Please rate the following statements from 1-*strongly disagree to* 7-*strongly agree*:

It was easy to do the task	1 ☐☐☐☐☐☐☐ 7
I felt efficient doing this task	1 ☐☐☐☐☐☐☐ 7

Fill from 1-*strongly disagree* to 7-*stronly agree*

Leave blank where not applicable or where you do not have an opinion

	Useful	Easy to Use	Effective
Device creation	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Role creation	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Profile selection	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Element selection	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Drag'n'drop distribution	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Full UI / Empty UI	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Remove element	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Clear Design	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Change between Design/Use	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Change between Simulated/On-device	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7

B.3 Overall Questionnaire

Overall Questions

How would you overall rate the design modes for the two scenarios?

Fill from 1-*worst* to 7-*best*

	Meeting Room	Classroom
Simulated	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
On-device	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7
Combined	1 ☐☐☐☐☐☐☐ 7	1 ☐☐☐☐☐☐☐ 7

Which mode did you prefer for every scenario?

Check the respective box

	Meeting Room	Classroom
Simulated	☐	☐
On-device	☐	☐
Combined	☐	☐

What other features would you like the tool to support?

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

Other comments

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………