JVLC

Journal of Visual Language and Computing

Volume 2019, Number 2

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Run-time adaptation of modalities of interaction and context of use in web mobile apps

Danilo Camargo Bueno, Luciana A. M. Zaina

[a]Department of Computer Science, Federal University of São Carlos - UFSCAR, Sorocaba (SP), BRA

ARTICLE INFO

Article History:
Submitted 04.29.2019
Revised 08.01.2019
Second Revision 10.09.2019
Accepted 10.09.2019

Keywords:
Dynamic Adaptation
Mobile web app
Context of use
Modalities of interaction
Front-end Framework

ABSTRACT

Several approaches have emerged to support the development of mobile web applications (apps). Front-end frameworks (FeF) have emerged to support developers in the construction of responsive mobile web apps. However, these frameworks do not provide resources to handle easily variables of the context of use and to deal with different modalities of interaction. Considering this gap, we proposed the HyMobWeb, an approach that assists developers in working with these aspects. By grasping the popularity of the FeFs, HyMobWeb proposes a flexible and reusable approach based on FeF structure. It works with a hybrid approach that treats the adaptation in two phases. The static one, performed in the development time, allows developers to implement the resources of adaptation. The dynamic one performs the adaptation during the run-time. In this article, we present the HyMobWeb dynamic adaptation approach. As end-users are the receivers of the dynamic adaptation our evaluation was carried out in the perspective of this audience. The results showed that the adaptations regarding the context of use were the ones that presented more impact on the user experience. We concluded that the HyMobWeb dynamic adaptation provides ways to enhance user interaction in mobile web apps when compared with RWD resources.

1. Introduction

Recently, developers have been facing issues on how to build web applications that are suitable to interact in small mobile devices [15, 16]. Some of these issues come up from the limitations that some browsers presented in supporting applications (apps) of different mobile platforms [1, 15]. Frequently, the development of Web applications require an extra effort of developers. Considering a base web version of the app, they have to customize the web versions to use resources (e.g. sensors, screen size) of the mobile devices [18].

The interaction in mobile devices create a new demand for the developers. They are forced to deal with the context in which the application will be used [7, 15]. Context of use emcompasses a set of elements as such as users preferences, device capabilities and environmental. These elements together provide the information about the context of use [12, 7].

Besides of questions about the user experience, by treating the aspects relevants to the context of user interaction the developers can enhance user performance on the apps [10, 15, 25]. Non-adaptive user interface for mobile devices can cause some frustration that directly affects the performance and usability of the users. The context of use can also affect the user preferences for different modalities of interaction [13].

The adaptation to different modalities of interaction can become suitable in situations in which the users have some discomfort caused by the luminosity, background noises, and other environmental issues. By providing different modalities of interaction developers can improve the usability in mobile apps [10].

Although the adaptation of mobile apps is not a new topic, there are some gaps to be explored regarding the tools and techniques to support the adaptations [1, 15, 20, 13]. Responsive Web Design (RWD) is an approach proposed by Ethan Marcotte that is widely used by Web developers. It main principles are flexible layouts with relative dimensions, flexible images and videos content through dynamic

danilocbueno@ifsp.edu.br (D.C. Bueno); lzaina@ufscar.br (L.A.M. Zaina)
ORCID(s): 0000-0002-7359-9937 (D.C. Bueno); 0000-0002-1736-544X (L.A.M. Zaina)

DOI reference number: 10-18293/JVLC2019-N2-011

resizing; and adaptation based on media queries[1] [16, 11].

Although RWD is widely used, it presents some limitations regarding the context of use. The detection in the context changes are restricted to the adaptation considering the screen resolution and device orientation. Additionally, the support to code the adaptations is merely provided by showing/hiding elements and changing some visual attributes [16, 17, 12].

From the increasing of popularity of RWD many Front-end Frameworks (FeF) came up to aid developers in the construction of web mobile applications. FeF combines the concepts of RWD to the User Interface Design Patterns (UIDP). UIDPs are descriptions of best practices or some sort of heuristics to fulfill design problems [23]. They are reusable solutions (e.g. menus, buttons, lists, and so on) that can be applied in different domains. A set of components built in HTML (Hypertext Markup Language), CSS and the JavaScript programming language composes the core of FeFs.

Bootstrap, Zurb Foundation, Pure, Materialize, KickStart[2] are examples of FeFs. Among them, Bootstrap has become the most popular one to support developers and designers work. Its popularity is a consequence of the framework maturity (precursor among the other FeFs) [19, 22].

However, these FeFs concerns strictly in handling features linked with RWD (e.g. different screen sizes) [12]. FeFs do not provide resources to implement adaptations of different input modalities or even different contexts of use [17]. As a consequence, developers have to employ more effort on the implementation of such features. Additionally, there is no pattern to support the designing of different interaction modalities as well as to treat the context of use.

Based on the RWD and FeF restrictions, we proposed the HyMobWeb approach. It provides resources to develop a hybrid adaptation on user interfaces of web mobile apps. A hybrid approach works with static and dynamic strategies of adaptation. HyMobWeb enables developers to indicates in code time (i.e. static strategy of adaptation) which parts of the web mobile user interface can be adapted during the run-time (i.e. dynamic strategy of adaptation). Static and dynamic adaptations are implemented from an extension of FeF. This approach has the advantage that it can be applied to different FeFs as they have similar structures. From HyMobWeb approach, developers can define different modalities of interaction considering the aspects of context of use. The static adaptation of HyMobWeb was previously presented in [5].

In this article, our goal is to present the HyMobWeb dynamic adaptation. To explore the HyMobWeb proposal, we extended the Bootstrap FeF and applied it in the development of a mobile web app. With the participation of 14 volunteers, we evaluated the dynamic adaptation from the perspective of end-users. To explore the potential of the proposal, we developed two versions of apps and conducted a comparison of them. The first one was implemented by using the basic RWD resources of Bootstrap. In the second, we evolved the first one by applying the resources available in the extension. We highlight that dynamic adaptation approach, and its evaluation was not presented in other work.

This article is organized as follow: related work is presented and discussed in Section 2; as both adaptations are interconnected, we briefly give an overview of HyMobWeb approach and of static adaptation in Sections 3 and 4, respectively; Section 5 presents the dynamic adaptation approach in details and Section 6 its evaluation; finally the discussion about the proposal and the final remarks are discussed in Sections 7 and 8, respectively.

2. Related Work

Our related work concerns on discussing of mobile web adaptations in different perspectives as follow.

Semantic Transformer [21] promotes the adaptation of user interface from web desktop to mobile web approach. The adaptation mechanism modifieds the user inteface structure to get one with low cost to be performed in a mobile device. Small Screen Device (SSD) Browser [1] also presents a desktop-mobile Web adaptation method based on visual and structural arrangement of interface elements. It identifies the position of elements and the relationship among the elements of the document and thus provides the mobile web user interface. Besides, the method allows users to set their own adaptations. However, a particular browser has to be adopted to use the proposal.

Tree Adapt [2] provides the adaptation by rearranging the interface elements to keep only the content which can fit into mobile devices screen. As a result users can visualize different contents by clicking on a title which triggers the block's expansion. W3Touch [18] takes the injection of JavaScript code to track touchscreen events and collected data from the user interaction (interaction tracking). By combining the data of user interaction to adaptation rules W3Touch automatically detects and suggests web page components. These adaptation rules are defined in accordance to the mobile device features. Yigitbas et al. [25] presents the CoBAUI, a component-based development framework for building adaptive interfaces. The work is based on the Angular - framework for the development of advanced Web applications. The developer create the component using the angular architecture and write the code that will perform the adaptation according to the previously defined rules.

A Web interface adaptation approach by combining the modalities of voice and graphic is proposed by [15]. The approach follows a set of rules to select the suitable adaptation to a given device. During the application development phase, the interface elements can be marked by a CSS class. Considering these marks and the adaptation rules, a multimodal interface is automatically generated. However, this approach demands for a browser extension which are not available on mobile browsers. To make the proposal feasibility, the authors developed a new mobile browser which has to

[1] *media query* is a CSS (Cascading Style Sheets) resource from which user interface can be adapted according to the features of devices.

[2] Links: getbootstrap.com, foundation.zurb.com, purecss.io, materializecss.com, getkickstart.com

be installed in the devices. AdaM approach [20] adapt interfaces in real-time collaborative environments promoting the rearragement of the interface elements. It works based on the analysis of information from the context of use, such as the user's device, functions, preferences, and access rights.

Although the works above discuss the topics of use interfaces adaptation, they present some gaps. Semantic Transformer [21], Tree Adapt [2], W3Touch [18] and AdaM approach [20] present different ways of manipulating a Web page's content so that it can adapt to the size of the mobile device. However, they concentrate on working in the content adaptation and not in the interface elements adaptation to different modalities of interaction.

SSD Browser and Manca proposal [1, 15] introduce barriers to their adoption so these proposals demand for the installation of new mobile browsers. CoBAUI [25] handles only variable of context of use, not including different modalities of interaction. Besides, as far as we know there were no works that explore the adaptation by applying the features of FeFs. And also we could not found works which provide resources to deal with the adaptation to different modalities of interaction and to the context of use.

3. HyMobWeb approach

HyMobWeb is a hybrid approach which supports the adaptation of the web mobile apps concerning on variables of the context of use in combination with the different modalities of interaction. The hybrid strategy blends static and dynamic approaches for interface adaptation. On the one hand, static adaptation is implemented by developers in coding time. On another hand, in dynamic strategy the adaptation in the code happens during the run-time [7]. By taking the advantages of user interface design patterns available in FeFs, HyMobWeb interconnects both strategies of adaptation without requesting the use of new web browsers. Additionally, HyMobWeb extends the structure of FeF to provide developers with resources to work on the adaptation of modalities of interaction and context of use. By using the same structure of the FeFs HyMobWeb approach becomes ease to be implemented in different FeFs.

Figure 1 illustrates the approach. The core of HyMobWeb is a Domain Specific Language (DSL). This DSL provides a set of codes which aids developers, during code time, to define what will be adapt considering the context of use and the modalities of interaction. The step (A) in Figure 1 shows how the static adaptation works. Based on the application constructed from a FeF (1a), the developers use the HyMobWeb DSL (2.1a) to indicate in which interface elements the adaptations will undergo (3a) during the run-time. The markings are introduced in the code of the interface elements.

Through an adaptation engine (Figure 1 - step B), HyMobWeb performs the dynamic adaptation by parsing the source code of the web mobile apps and thus modifying the elements which were previously marked in the development time (step A). In the dynamic adaptation, the engine checks each marking that was inserted during the coding time (i.e. from static adaptation (1b)), interprets such codes and delegates the execution to multimodal (2.1b) and context (2.2b) handlers. The *Multimodal Handler* manages the adaptations of the interaction modalities of the elements. The *Context Handler* analyzes continuously the changes that occur in the variables of user context, and then performs the adaptation. Finally, the adaptation engine returns the code adapted to the application (4b).

As mentioned in the Introduction section, the static adaptation of HyMobWeb and its evaluation were already presented in [5]. As the dynamic adaptation are interconnected to static, we will discuss briefly the static one in next section. Sequentially, we will present the dynamic one which is the focus of this work.

4. Static adaptation

The main component of static adaptation is the HyMobWeb DSL. It allows developers to set which elements of context of use and modalities of interaction will be take into account in a run-time adaptation. The approach considers three categories of variables of context of use based on the proposal of [21]: *user* (i.e. preferences, goals and tasks, physical position), *technology* (i.e. screen resolution, connectivity, browser, battery), and *environment* (location, luminosity, noise level). Our approach proposed two distinct ways for the implementation of the code of the adaptation.

In the first one, the developer uses CSS language following the grammar presented in Definition 1. The `@context` points to the *Context Handler* that a context variable should be treated; the `<aspect>` describes which variables of the context will be considered; the `<state>` informs which is the value that will triggers the adaptation; and the `{...}` defines the CSS code that will be applied to one or more elements to provide the adaptation. Some examples are *@context (user-activity: walking){ ... }*, *@context (battery-level: low){ ... }*, *@context (noise-level: high){ ... }*. They are variables of context of use that when changes to that state will cause the adaptation. They represent the conditions of the user walking, the device's battery at a low level and a high noise level in the environment, respectively.

Definition 1. *grammar to use in CSS code*

@context *(<aspect> : <state>) { ... }* (1)

The second way allows developers to configure which interface elements will suffers the adaptation. To do this, developers make a linking between HTML elements and CSS class following the grammar presented in Definition 2. The `<object-behavior>` represents the behavior that the object should take; the `aspect` reports the available context aspects (battery-level, luminosity, user-activity); the `state` describes the values that trigger the execution of the behavior. This proposal can be applied to a variety of contexts and states. HyMobWeb provides a list of pre-defined behaviors as such as *visible*, *hidden*, *bigger*, *darken*. For instance, some potential classes can be *.visible-on-user-activity-walking*, *.increase-on-luminosity-low* and *.hidden-on-noise-level-low*.

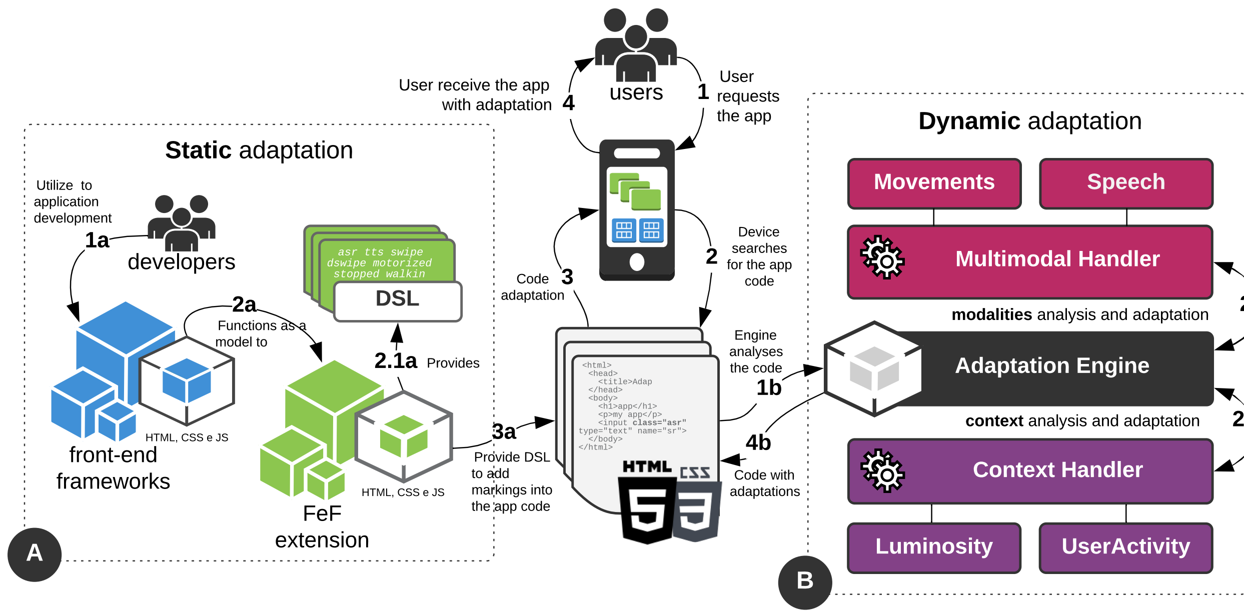

Figure 1: HyMobWeb approach.

Definition 2. *grammar to use in HTML code*

$$.\langle object\text{-}behavior\rangle\textbf{-on-}\langle aspect\rangle\textbf{-}\langle state\rangle \quad (2)$$

Following the same structure of grammar above, HyMobWeb offers the possibility of using new interaction modalities. In the `<aspect>`, developers report on which aspect the modalities of interaction will be available (i.e. interface element, screen, movements, device). Some examples are *.listen-on-element-focus* (activate automatic speech recognition when the element receive focus), *.speak-on-user-activity-walking* (activate speech synthesizer when the user is walking), *.hidden-on-movements-swipeleft* (hide the element when the swipe to the left movement is recognized), and *.vibrate-on-luminosity-low* (vibrate the device when the ambient light level is low).

HyMobWeb DSL is generic enough to be applied to different aspects. However, technological features of devices and browsers' features have to be taken into account during the implementation of the grammar.

To implement an instance of our approach, we extended the *Bootstrap* FeF. We added the modalities of *Speech* and *Movements* providing to developers ways to work with automatic speech recognition, text-to-speech, swipe movements, double swipe, double tap and pinch. Regarding the context of use, the extension supported developers on the analysis of characteristics of user and environment using the contexts of *User Activity* and *Luminosity*. These consider the user's physical state (i.e. stopped, walking or motorized) or the level of luminosity (i.e. low, medium or high), respectively.

As mentioned previously, we already carried out the evaluation of static strategy of HyMobWeb approach with 19 developers. The findings showed that the developers were able to use the static adaptation of HyMobWeb and also demonstrated the developers could perceive the utility of the approach. The details of static adaptation implementation and evaluation can be found in [5].

5. Dynamic Adaptation

Dynamic adaptation (see Figure 1-B) is performed based on the settings of variables of context and modalities of interaction. These settings are done during the coding phase (i.e. static adaptation). The core of HyMobWeb dynamic strategy is the *Adaptation Engine*. The adaptation engine is composed by a set of HTML, CSS and Javascript codes that are installed on the web mobile application. They execute the parsing on the code and perform the adaptations. *Multimodal* and *Context Handlers* (see Figure 1 - 2.1b and 2.2b respectively) are responsible to the treatment of the variables and to the changings in the code in accordance with the results of code analysis. HyMobWeb adaptation engine (see Figure 2) is conceived as a based-component architecture which makes easy its extension. The operation of the *Context Handler* and *Modality Handler* components will be explained individually.

5.1. Context Handler

First, the *Context Handler* calls `loadContexts()` method that loads a list of all context aspects which are available (i.e. context variables that are implemented). We implemented these aspects by using regular expression. After loading the context aspects, the `analyzePresentation()` method performs the parsing into the CSS code of the application to search for matchings of code chunks and the regular expressions. Figure 3 shows two examples of regular expressions (read from the left to the right). The first one refers to the checking of

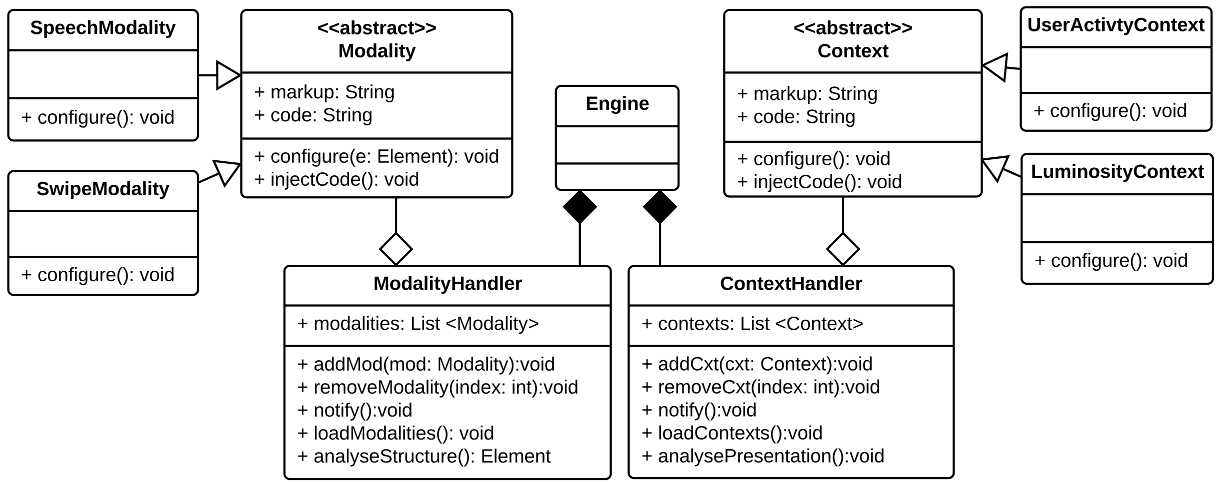

Figure 2: HyMobWeb architecture.

```
/\@context\s*\(\s*luminosity\s*:\s*((high|low|medium))\s*\)/
/\@context\s*\(\s*user-activity\s*:\s*((stopped|motorized|walking))\s*\)/g
```

Figure 3: Regular expression available in the Context Handler.

luminosity aspect and the second to users activity regarding their movement.

To illustrate the *Context Handler* operation, we present an example (see Figure 4). *Context Handler* parses the code apps searching for the chunks that match to the regular expressions (1). Thus, this handler finds the code of the file "main.css" (2) and then triggers the *configure()* method. This method works as context listener (3) by checking the changing in the context of use. In this example, the *devicelight* event is responsible for observing any changes of the light level. At each change, the respective luminosity value is stored and the *analyzeContext()* method is invoked. This method checks whether the information in the current context matches to the information represented in the code. The function used by *analyzeContext()* is shown in (4) that checks whether the current context values match those declared in code (2). In this way, when the context status declared in the code (luminosity: low (2)) is reached, the handler proceeds with the injection of code (i.e. *injectionCode()* method) into the application (5).

To implement *Luminosity* aspect, we took the Ambient Light Sensor API. This API defines a sensor interface to monitor the ambient light level or brightness of the environment where the device is being used. The current specification is reported as draft status and is supported by the browsers Mozilla Firefox[3] and Microsoft Edge[4] [14, 9]. Despite limitations on browser support, the approach works with progressive enhancement. Such strategy aims to reach the least capable devices first by providing a favorable experience for devices with a limited amount of available resources.

[3]https://www.mozilla.org/firefox
[4]https://www.microsoft.com/en-us/windows/microsoft-edge

From this, one or more layers of enhancements are added depending on the specific capabilities of each browser [8]. In this way, the approach will provide different interact ways for those who have the characteristics available and will maintain a suitable experience for those who do not have them.

5.2. Multimodal Handler

Similar to the *Context Handler*, the operation of this handler starts by loading the list of modalities (`loadModalities()` method). The handler works based on the parsing of HTML code. It looks for markings in the code that identify the use of modalities.

Looking at Figure 5, we can see an example of how *Multimodal Handler* works. By performing the `analyzeStructure()` the handler verifies the HTML code of the application (2) and finds the *listen* identification. From this, the `configure()` method is called to set the Speech mode (3) by executing the CSS code injection, modifying the HTML code, and configuring the speech API. In this example, the voice modality is setting as data input that is represented by the microphone icon (4).

The *Speech* aspect is implemented by using Speech Recognition and Speech Synthesis (components of Speech API Specification). They allow the inclusion of speech recognition and synthesis in web applications by using script language. Currently, Mozilla Firefox[5] support partially this API (requires specific browser settings) and Google Chrome[6] has total support [6, 9].

The operation of the *Movements* modality is similar to *Speech* one. The main difference is that to *Movements* as-

[5]https://www.mozilla.org/firefox
[6]https://www.google.com/chrome

Figure 4: Example of dynamic adaptation of *Luminosity* aspect.

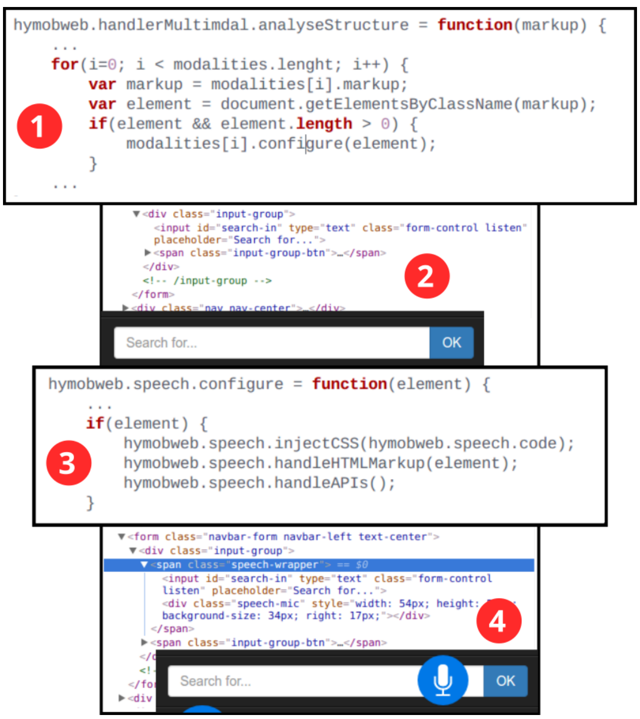

Figure 5: Example of dynamic adaptation of *Speech* aspect.

pect it is not necessary to insert code in the application, because the marking and the visual of the element in question are not changed. The modality, in this case, will only be responsible to modifying the behavior of the element. The implementation was accomplished using the Touch Events API that is supported in Mozilla Firefox, Google Chrome, and Microsoft Edge browsers[4, 9].

6. Evaluation of the dynamic adaptation

As the dynamic adaptation has a direct impact on the use of the apps, our evaluation focused on exploring the HyMobWeb approach in the perspective of end-users. The data and results presented in this article have not published before. The evaluation in the perspective of developers (i.e. evaluation of static adaptation) can be found in [5].

To examine the dynamic adaptation, we carried out a controlled experiment with the objective of verifying end-users feedback. This evaluation is relevant, since end-users are those who will use the resources provided by dynamic adaptation. We followed the Wohlin et al [24] guidelines to organize the planning, execution and analysis phases of this study.

6.1. Planning

Initially, we defined the research question (RQ): *How is the experience of end-users regarding the features provided from HyMobWeb dynamic adaptation?*. To answer our RQ, we took a comparison method. We developed two versions of a music player mobile web app. Both versions of the app provided features to browse among the albums available in the library, visualize the album's songs, play and pause a song, and turn the volume up or down.

The two versions were built by a developer with two years of experience in web development. First, the developer implemented a music player version only using the RWD resources available in *Bootstrap* FeF. We named it of *Base* version. This version had restrictions about the use of different modalities of interaction and adaptation to the context of use. For the second version, the developer extended the *Base* one by applying the resources of the HyMobWeb DSL (i.e. resources available in FeF extension - see Figure 2-A). The details about DSL can be found in Section 4. The developer added the features of adaptation modalities of interaction and context of use by applying the static adaptation. This second version was named *HyMobWeb*.

We elaborated four tasks (see Table 6.1) to guide the participants interaction during the study. For each task, we had different treatments to the two versions (i.e. the *Base* and *HyMobWeb* apps). A tutorial was developed to assist the end-users to accomplish the tasks.

Self-Assessment Manikin (SAM) instrument was adopted to collect the end-users feedback. SAM is a pictograph evaluation method to measure emotional responses of users after some sort of stimulus. Three dimensions are considered by this technique: pleasure (if the participant had a positive

Table 1
List of tasks - aspects of modalities of interaction (a) and context of use (b)

Task	Aspect	Base	HyMobWeb
(T1) Browse through the available albums until found a specific one	Movements (a)	Touch	Swipe
(T2) Search for a particular album through a keyword	Speech (a)	Typing	Speak
(T3) Read a text in the mobile app in a low-light environment	Luminosity (b)	Not available	Available
(T4) Select a song while walking	UserActivity (b)	Not available	Available

Table 2
Results: participants' interaction (ID), which version the participant has interacted first (APP) - Base (BA) or HyMobWeb (HW), and SAM indexes values - pleasure index (PI), arousal index (AI) and dominance index (DI).

		(T1) Movements						**(T2) Speech**					
		Base			HyMobWeb			Base			HyMobWeb		
ID	*APP*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*
1	BA	4	4	4	8	9	8	3	4	4	9	9	9
2	HW	7	6	7	9	9	9	6	7	8	8	9	8
3	BA	5	3	5	8	8	9	8	6	8	9	8	8
4	BA	8	8	8	9	8	9	4	5	6	9	5	9
5	BA	8	8	8	9	9	9	6	6	5	9	9	9
6	BA	8	7	7	9	9	9	7	5	6	9	7	7
7	HW	8	8	9	9	9	9	9	7	9	9	9	7
8	HW	5	5	5	7	7	9	5	5	6	8	8	8
9	HW	8	8	9	5	6	4	7	8	7	9	9	9
10	BA	6	6	7	8	9	9	5	7	6	8	9	8
11	HW	9	7	9	9	9	9	9	9	9	7	9	6
12	BA	5	5	6	9	9	9	5	6	5	9	9	9
13	HW	7	8	5	8	8	9	9	9	8	6	7	5
14	BA	7	6	6	8	8	8	9	9	8	5	2	6
		(T3) Luminosity						**(T4) UserActivity**					
		Base			HyMobWeb			Base			HyMobWeb		
ID	*APP*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*	*PI*	*AI*	*DI*
1	BA	5	4	5	8	9	9	3	2	5	7	8	9
2	HW	6	7	8	9	9	9	5	6	6	8	8	8
3	BA	5	4	8	9	8	8	5	6	4	9	8	9
4	BA	5	5	5	9	9	9	6	8	9	9	9	9
5	BA	6	6	5	9	9	9	8	8	8	9	9	9
6	BA	7	5	6	9	7	7	3	3	2	9	8	7
7	HW	7	7	9	9	9	7	7	7	9	9	9	8
8	HW	5	5	6	8	8	8	6	6	6	9	9	9
9	HW	7	8	7	9	9	9	9	8	8	6	8	6
10	BA	5	7	6	8	9	8	6	7	7	9	8	7
11	HW	7	7	9	9	9	9	8	6	9	9	9	9
12	BA	4	5	5	9	9	9	7	7	7	5	5	6
13	HW	5	2	4	9	9	9	7	6	3	9	7	9
14	BA	5	6	5	9	9	8	3	5	5	9	9	9

or negative reaction), arousal (body stimulation level from an event or object) and dominance (feeling in control of the situation or controlled by it). The user chooses a value on a scale of one to nine on each dimension, using images, to represent their emotions after each task [3]. Besides of SAM, a post-experiment questionnaire was applied to collect the participants' profile that includes questions about their age, education level, the frequency of Internet access and use of device resources, and their perceptions about the *Base* and *HyMobWeb* versions.

A pilot test was running with two individuals. This test aimed at checking whether the study instructions and procedures were sufficiently clear to the participants. We observed some difficulties of the individuals during the pilot test, and hence, we refined the tutorial.

6.2. Execution

A total of 14 participants took part in the study voluntarily and were selected by convenience. They were students, teachers and collaborators from the Federal Institute of São Paulo (IFSP), Itapetininga, Brazil. The study was carried out in the Informatics lab. All the volunteers signed the consent form in which the participants agreed about the use of their data for academic purposes.

All the participants were used mobile devices (smartphones) daily. Most of them aged 16 to 18 years old (75%), attended high school (91.7%), and accessed Internet through mobile devices frequently (95.4% access daily). To avoid problems during the experiment execution, two smartphones were provided to the participants during the study (one ASUS ZenFone 5 and one Xiaomi Redmi 4).

An initial explanation about the study conduction, its goal and evaluation process were done. The tasks were presented in a textual way on the mobile device screen. All the interactions of the participants were recorded by using the Ace Screen Recorder application[7] that was installed into the smartphones.

All participants used both app versions (i.e. *Base* and *HyMobWeb*). However, we performed a balance in the order in which the versions were used. From this balance, we avoid the introduction of biases in the results which could be caused by the order the versions were used. Hence some participants interacted first with the *Base* version and others with the *HyMobWeb* one.

After finishing each task, the participant answered the SAM questionnaire. At the end of all the tasks, the users responded the post-questionnaire composed of seven questions about their frequency of use of mobile devices and four non-mandatory questions about their perception about the both app versions.

6.3. Analysis

The SAM and post-questionnaire answers and the individual recordings of participants interaction were analyzed. To answer our RQ (*How is the experience of end-users regarding the features provided from HyMobWeb dynamic adaptation?*), we analyzed the videos contained the participants' interactions tasks and the SAM answers for each one. Additionally, our results were complemented by the responses of open-questions found in the post-questionnaire. Table 2 presents the results per participants for each task. We also associated the aspect (see Table 6.1) observed for each task.

[7]https://play.google.com/store/apps/details?id=com.dev47apps.screenrecorder

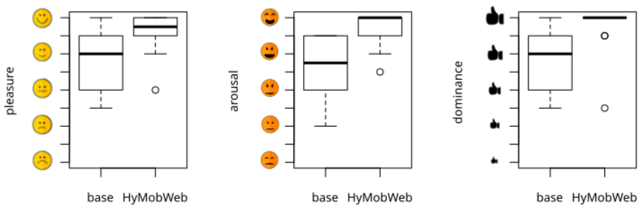

Figure 6: Boxplots for the pleasure, arousal and dominance of the Movements aspect (T1).

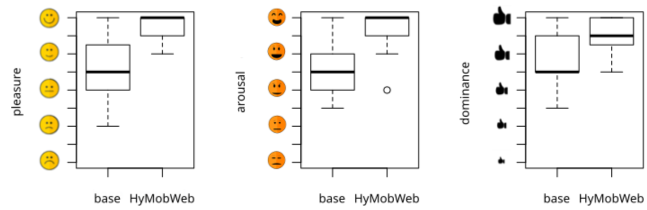

Figure 7: Boxplots for the pleasure, arousal and dominance for the Speech aspect (T2).

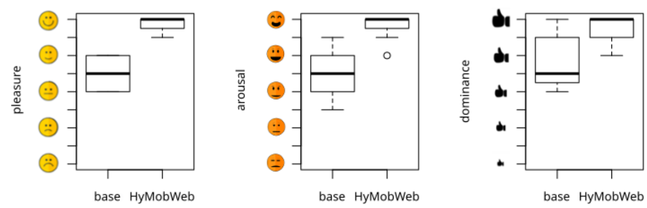

Figure 8: Boxplots for the pleasure, arousal and dominance indexes for the Luminosity aspect (T3).

6.4. Threats of Validity

The validity of our study was treated by taking into account some strategies considering the levels of validity treatment: internal (i), external (ii), construction (iii) and completion (iv) [24].

Internal threat (i) refers to the tiredness of the participants. To mitigate this, we prepared a set of tasks that could be accomplished in 20 minutes. External threat (ii) refers to the sample. The participants of our study represented typical users of mobile devices. For them, *smartphones* are devices that take part in their daily so they could naturally perform the tasks.

Construct threat was mitigated in two ways (iii). First, the participants' interaction was randomly divided into two groups. While one group had their first interaction by using the *Base* version the other group started its interaction with *HyMobWeb*. Afterwards, the groups changed the order of interaction. Besides, the tasks were shuffled, so the album requested to be located in the *Base* version was in a different position than the same one on *HyMobWeb* version.

We handled the threat of conclusion (iv) by crossing the data collected from the perspective of the participants (i.e. from using SAM) and from the analysis of the videos. This proceeds avoided our conclusions were driven by only one source of data.

6.5. Findings

Our results will be discussed for each aspect of modality of interaction or context of use related to the tasks (see Table 6.1). We took the descriptive statistics to support our considerations about the findings. Additionally, the results showed in Table 2 give the background to our discussions.

First, considering the *Movements* aspect (see Figure 6) related to the first task (T1 - see Table 6.1). The results show that SAM indexes have their values concentrated at the highest levels for the *HyMobWeb* version. The three dimensions (i.e. PI, AI and CI) revealed similar results for the *Base* version. This version presents a measure of central tendency equal to or less than 7. On the other hand, *HyMobWeb* version presents values of 8.5 for the indexes of pleasure (PI), and 9 for arousal (AI) and dominance (DI). In dominance index (DI), the concentration of the values is even more perceptive. In this dimension, only two users stand outside the maximum SAM scale.This result suggests the participants had a higher level of control while using the swipe mode than by using the touch interaction. However, some outliers was found (see Figure 6). In the post-questionnaire, we could found the explanation from a user statement: *"I was afraid to move in the albums on too fast and end up missing that one I wanted"*. By examining the videos, we could see the participants struggled in performing some touch interaction. As the *Base* version used the RWD resources, some icons were automatically reduced in their size. Additionally, we noticed that the users attempted to perform the swipe movement in the album gallery even when the instructions explicitly oriented to take the touch interaction.

Figure 7 shows the results for the *Speech* aspect related to the second task (T2 - see Table 6.1). We see that the pleasure index values (PI) are greater in the *HyMobWeb* version that in the *Base* one. For the voice recognition resource to work, the internet connection should be available. Consequently, the problems caused by some interruptions in the Internet signal that happened may have impacted on the results of the arousal index (AI). Nevertheless, besides the speech mode, the participants had the typing mode available even while Internet access fails. Taking into account the post-questionnaire answers, we see that most of the participants stated they liked to keep the interaction by speech mode for being more practical. Some participants reported that they choose voice mode only in the situation that typing mode is not suitable, for instance, when they are driving a car.

The highest levels in all the SAM indexes were obtained by *Luminosity* aspect, which is linked to the third task (T3 - see Table 6.1). In Figure 8, we see that whereas the median remains at the highest level of the scale (i.e. 9) in the three indexes for the *HyMobWeb* version, for the *Base* one the value is 6. Some participants reported that *"...when the luminosity changes automatically the reading becomes more comfortable..."; "...this change avoids I squint my eyes to see better.."*. Besides, by observing the videos that contained the users' interaction, we could notice a positive facial expression as soon as the luminosity automatically changed (see Figure 9).

Finally, in the *UserActivity* aspect linked to the last task

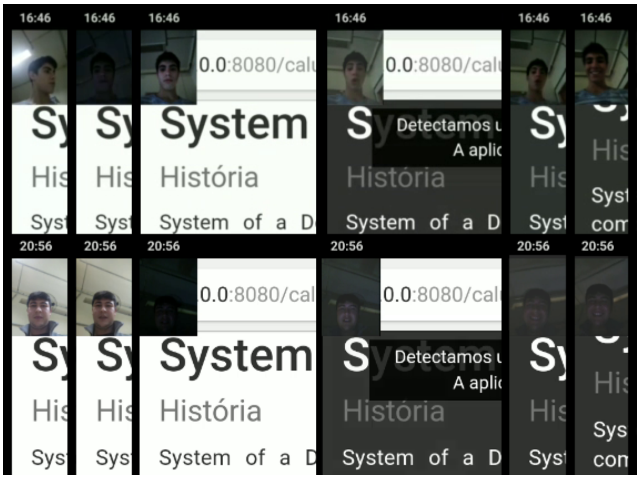

Figure 9: Positive facial expression - reaction when the Luminosity aspect worked.

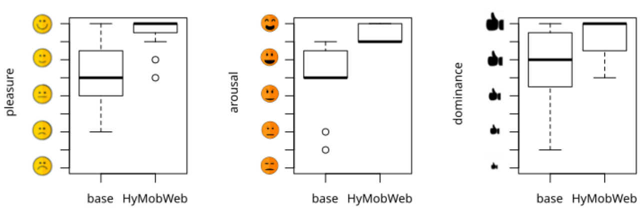

Figure 10: Boxplots for the pleasure, arousal and dominance for the UserActivity aspect (T4).

(T4 - see Table 6.1), we see again that the *HyMobWeb* version got the highest levels in the SAM indexes (see Figure 8). Google API[8] provided the resources to catch the user's movements (e.g. if they are standing, walking or moving). However, this API takes a few seconds for the recognition of the user activity. Consequently, the delays provoked by the API can be the cause of some outliers (see Figure 8). Although this affected the participants' feedback regarding the SAM indexes negatively, we could get some positive responses from the post-questionnaire as such: *"I liked this resource...while I am walking or running I could see the most important information in the same position on the screen...the things do not move from one position to another..";"...the recognition of my moving improves my ability for interacting with the app..."*.

Retaking the RQ (*How is the experience of end-users regarding the features provided from HyMobWeb dynamic adaptation?*), we can respond that the resources of *HyMobWeb* dynamic adaptation could provide a good experience to end-users.

7. Discussion

Our findings revealed that the three SAM dimensions (i.e. pleasure, arousal and dominance) presented high values for the *HyMobWeb* version. We could notice that the use of different modalities of interaction made the user interaction easier in some tasks. However, the aspects of the context of use were the ones that presented more impact on the user experience. Taking into account all the aspects we considered in this study, we can state that the Luminosity and User Activity were those that most have influence into the pleasure index values.

Different solutions for working with dynamic adaptation in the context of mobile web apps can be found in the literature [21, 1, 18, 2, 15, 20]. However, our approach presents new contributions. First, our approach concerns on providing resources for performing the app adaptation based on an extension of FeFs features and RWD techniques. As FeFs and RWD are widely known by software developers [17, 18] they will spend less time in learning how to apply the adaptation resources. Consequently, their learning curve for the implementation of the adaption resources will be reduced. Besides, HyMobWeb proposes a DSL and a structure to implement the apps adaptation that is flexible and easily extensible to different FeFs [13]. Given the flexibility of HyMobWeb, new handlers can be inserted into the approach in addition to Context and Modalities handlers. This flexibility allows that other resources could be implemented or customized to attend the developers' demands.

One of the advantages of HyMobWeb dynamic adaptation is that the values of context of use could be caught without using intelligent agents or external plugins. Differently, the works of [1] and [15] that request the use of additional resources. User actions can be monitored only using resources of HTML, CSS and Javascript that are native lightweight technologies for web development. Although such mechanisms have already been used in other approaches [2], in HyMobWeb proposal, these consider the standard structure of FeF. By following these structures, our approach becomes reusable for different apps.

Additionally, HyMobWeb provides ways to promote two different adaptations in the same app. One focused on the modality of interaction and another on variables of the context of use. The adaptation of modes of interaction and context of use had already been addressed in other [15, 18, 25]. However, as far as we know, our approach is the first that brings these two possibilities together.

8. Final remarks and future works

HyMobWeb is an approach that provides resources to perform adaptation on web mobile application. The core of the proposal is to allow developers to dealing with different modalities of interaction and with context of use through hybrid adaptation. During the coding time (i.e. static adaptation), developers can identify in the code which parts can be adapted during the run-time (i.e. dynamic adaptation). By extending the *Bootstrap* FeF, we provided concrete ways to use the HyMobWeb proposal. The evaluation of static adaptation was already done in another work.

In this work, we focused on presenting the dynamic adaptation of the approach and its evaluation. The dynamic adaptation was built from an extensible architecture based on han-

[8] https://developers.google.com/android/reference/packages

dler components which give the flexibility to adding other kinds of user interaction variables. The implementation of variables of context of use could be used in combination with different modalities of interaction.

The evaluation from the end-users perspective showed that the adaptations provided by HyMobWeb can potentially improve the user experience in mobile web apps. Some limitations of our proposal are: HyMobWeb does not provide ways of end-users set their preferences and we did not explore the complexity of handling various contextual aspects and different modalities of interaction together.

As future works, we intend to explore other categories and elements of interaction to identify their problems and hence to propose solutions in the scope of HyMobWeb. We also intend to run testings to observe the performance of our approach in different settings of browsers and Internet conditions.

9. Acknowledgement

The authors thank the grant 2013/25572-7, São Paulo Research Foundation (FAPESP).

References

[1] Ahmadi, H., Kong, J., 2012. User-centric adaptation of web information for small screens. J. Vis. Lang. Comput. 23, 13–28. URL: http://dx.doi.org/10.1016/j.jvlc.2011.09.002, doi:10.1016/j.jvlc.2011.09.002.

[2] Anam, R., Ho, C.K., Lim, T.Y., 2014. Tree Adapt: Web Content Adaptation for Mobile Devices. International Journal of Information Technology and Computer Science 6, 1–13. URL: http://www.mecs-press.org/ijitcs/ijitcs-v6-n9/v6n9-1.html, doi:10.5815/ijitcs.2014.09.01.

[3] Bradley, M.M., Lang, P.J., 1994. Measuring emotion: The self-assessment manikin and the semantic differential. Journal of Behavior Therapy and Experimental Psychiatry 25, 49 – 59. URL: http://www.sciencedirect.com/science/article/pii/0005791694900639, doi:https://doi.org/10.1016/0005-7916(94)90063-9.

[4] Brubeck, M., Moon, S., Barstow, A., Schepers, D., 2013 (accessed 13 March 2016). Touch Events. W3C Recommendation. W3C. http://www.w3.org/TR/2013/REC-touch-events-20131010/.

[5] Bueno, D.C., Zaina, L.M., 2017. Hymobweb: A hybrid adaptation of context-sensitive web interfaces with multimodality support in mobile devices. SBC Journal on Interactive Systems 8, 20–34.

[6] Burnett, D., Shuang, Z.W., 2010 (accessed 13 April 2016). Speech Synthesis Markup Language (SSML) Version 1.1. W3C Recommendation. W3C. Http://www.w3.org/TR/2010/REC-speech-synthesis11-20100907/.

[7] Cirilo, C.E., Prado, A.F., de Souza, W.L., Zaina, L.M., 2010. A hybrid approach for adapting web graphical user interfaces to multiple devices using information retrieved from context., in: The 16th International Conference on Distributed Multimedia Systems, pp. 168–173.

[8] Desruelle, H., Blomme, D., Gielen, F., 2011. Adaptive Mobile Web Applications Through Fine-Grained Progressive Enhancement. The Third International Conference on Adaptive and Self-Adaptive Systems and Applications , 51–56URL: http://www.thinkmind.org/index.php?view=article{&}articleid=adaptive{_}2011{_}3{_}20{_}50030.

[9] Deveria, A., 2017 (accessed 01 April 2017). Can I use... Support tables for HTML5, CSS3, etc. URL: https://caniuse.com/.

[10] Dumas, B., Solórzano, M., Signer, B., 2013. Design guidelines for adaptive multimodal mobile input solutions, in: Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services, ACM, New York, NY, USA. pp. 285–294. URL: http://doi.acm.org/10.1145/2493190.2493227, doi:10.1145/2493190.2493227.

[11] Fabri, D.W., Krempser, T., Filgueiras, L.V.L., 2013. Estudo de responsive web design aplicado a um sistema de pesquisa de opinião na área médica, in: Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems, Brazilian Computer Society, Porto Alegre, Brazil, Brazil. pp. 264–267. URL: http://dl.acm.org/citation.cfm?id=2577101.2577159.

[12] Ghiani, G., Manca, M., Paternò, F., Porta, C., 2014. Beyond responsive design: Context-dependent multimodal augmentation of web applications, in: Proceedings of the 11th International Conference on Mobile Web Information Systems, Springer International Publishing, Cham. pp. 71–85. doi:10.1007/978-3-319-10359-4_6.

[13] Iqbal, M.W., Ahmad, N., Shahzad, S.K., Feroz, I., Mian, N.A., 2018. Towards adaptive user interfaces for mobile-phone in smart world. International Journal of Advanced Computer Science and Applications 9. URL: http://dx.doi.org/10.14569/IJACSA.2018.091177, doi:10.14569/IJACSA.2018.091177.

[14] Kostiainen, A., 2016 (accessed 13 March 2016). Ambient light sensor. https://www.w3.org/TR/2016/WD-ambient-light-20160830/.

[15] Manca, M., Paternò, F., Santoro, C., Spano, L.D., 2013. Generation of multi-device adaptive multimodal web applications, in: Proceedings of the 10th International Conference on Mobile Web Information Systems - Volume 8093, Springer International Publishing, New York, NY, USA. pp. 218–232. URL: http://dx.doi.org/10.1007/978-3-642-40276-0_17, doi:10.1007/978-3-642-40276-0_17.

[16] Marcotte, E., 2017. Responsive web design: A book apart n4. A Book Apart, Eyrolles.

[17] Nebeling, M., Norrie, M.C., 2014. Beyond responsive design: Adaptation to touch and multitouch, in: Web Engineering, Springer International Publishing, Cham. pp. 380–389. doi:10.1007/978-3-319-08245-5_23.

[18] Nebeling, M., Speicher, M., Norrie, M., 2013. W3touch: Metrics-based web page adaptation for touch, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA. pp. 2311–2320. URL: http://doi.acm.org/10.1145/2470654.2481319, doi:10.1145/2470654.2481319.

[19] Otto, M., Thornton, J., et al., 2015 (accessed 20 April 2015). Bootstrap· the world's most popular mobile-first and responsive front-end framework. URL: https://getbootstrap.com/.

[20] Park, S., Gebhardt, C., Rädle, R., Feit, A.M., Vrzakova, H., Dayama, N.R., Yeo, H.S., Klokmose, C.N., Quigley, A., Oulasvirta, A., Hilliges, O., 2018. Adam: Adapting multi-user interfaces for collaborative environments in real-time, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA. pp. 184:1–184:14. URL: http://doi.acm.org/10.1145/3173574.3173758, doi:10.1145/3173574.3173758.

[21] Paternò, F., Zichittella, G., 2010. Desktop-to-mobile web adaptation through customizable two-dimensional semantic redesign, in: Proceedings of the Third International Conference on Human-centred Software Engineering, Springer-Verlag, Berlin, Heidelberg. pp. 79–94. URL: http://dl.acm.org/citation.cfm?id=1939212.1939221.

[22] Spurlock, J., 2013. Bootstrap: Responsive Web Development. O'Reilly Media.

[23] Toxboe, A., 2016 (accessed 15 April 2016). UI-Patterns. URL: http://ui-patterns.com/.

[24] Wohlin, C., Runeson, P., Hst, M., Ohlsson, M.C., Regnell, B., Wessln, A., 2012. Experimentation in Software Engineering. Springer Publishing Company, Incorporated.

[25] Yigitbas, E., Josifovska, K., Jovanovikj, I., Kalinci, F., Anjorin, A., Engels, G., 2019. Component-based development of adaptive user interfaces, in: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM, New York, NY, USA. pp. 13:1–13:7. URL: http://doi.acm.org/10.1145/3319499.3328229, doi:10.1145/3319499.3328229.