Enhancing Collaborative Sketching Activities with Context-aware Adaptation Guidelines

Vivian Genaro Motti, Ugo Sangiorgi, Jean Vanderdonckt
Louvain Interaction Laboratory - Universite catholique de Louvain
Place des Doyens 1 - 1348 - Louvain la Neuve - Belgium
vivian.genaromotti,ugo.sangiorgi,jean.vanderdonckt@uclouvain.be

ABSTRACT

Designing interactive systems for multiple contexts of use becomes a burden when the end user interaction takes place in distinct scenarios whose specific characteristics and constraints vary and must be carefully considered. Stakeholders face then two main challenges: they are not aware of *what* among several context information is significantly relevant to consider, or *how* to appropriately adapt the user interfaces according to the information considered. Furthermore, stakeholders cannot simply rely on existing UI editors once they usually do not provide enough support for adaptation. Thus, adaptation is often ignored, resulting in user interfaces that are only suitable for static and conventional contexts of use. To support the design of user interfaces that are properly adapted to their target context, this paper proposes a novel methodology to enhance sketching activities by proposing to the end user context-aware adaptation guidelines. This work aims at raising awareness about context-aware adaptation since the early stages of the UI design.

Keywords

Collaborative sketching; Context-aware adaptation; Collaborative design; UI Prototyping.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Prototyping

1. INTRODUCTION

Users currently interact from contexts that vary concerning the user's profiles (e.g. their preferences and impairments), their platforms (e.g. devices and connections) and environments (e.g. the brightness and the noise level). Due to such variety, for stakeholders it is a challenge to carefully consider the characteristics and constraints of these contexts during all phases of the user interface development. Thus, adaptation is often ignored, either because stakeholders are not aware of relevant characteristics and constraints of the context, or because they are aware but unable to correctly address related issues to implement and to provide efficient adaptation. Consequently, stakeholders tend to adopt a "one-size-fits-all" approach, providing UIs that are suitable for a conventional and static context of use. By doing so, accessibility issues may raise, hindering or even preventing the end user to accomplish her goals.

Due to the inherent challenges in properly considering many different contexts of use and in efficiently providing adaptation since the early phases of the development process, in this work we propose a novel approach that support adaptation within user interface editors, aiding stakeholders to consider context-awareness during all the development phases, starting from low fidelity prototypes. Such approach not only raises awareness for contextual differences but also facilitates the adoption of adaptation guidelines. The approach proposed consists in enhancing the electronic sketching and prototyping activities according to the envisaged context(s) of use.

Multi-platform sketching is defined as the activity of drawing with an electronic stylus in different devices while still having the same system running on those different devices [15]. GAMBIT has been developed as a multi-platform collaborative tool for user interface design that enables users to sketch and to prototype UI's on different devices. To augment the sketching activities by guiding stakeholders through the development of adapted UI's according to the target context, three main dimensions of context have been considered:

- *Multi-platform*: ensures more suitable results by enabling the sketching, prototyping and testing of the UI's to be performed on the target device. GAMBIT has been built with HTML5 and JavaScript being capable of running on any device with browsing capabilities, through a browser or embedded into a native application;
- *Multi-user*: since the user profiles vary in many aspects, e.g. preferences, expertise levels (novice, intermediate, advanced) or impairments (visual, cognitive, motor), also distinct requirements and needs must be considered for the UI;
- *Multi-environment*: for varied constraints and characteristics, different adaptation techniques are applicable. For instance taking into account the brighteness and the stability levels in order to provide users appropriate UI elements.

This paper is organized as follows: the next section motivates this work, presents and discusses related works; then Gambit is described, followed by the novel proposed approach to support context-aware adaptation guidelines for sketching.

2. MOTIVATION AND RELATED WORK

User Interface design is inherently a complex interdisciplinary activity, often involving designers, programmers, users and stakeholders. UI designers often use sketched prototypes since not all the UI specifications are known at design time, but rather evolve significantly over time when design sessions are organized. De-

WebMedia'13, November 5–8, 2013, Salvador, Brazil.

ACM 978-1-4503-2559-2/13/11 ...$15.00.
http://dx.doi.org/10.1145/2526188.2526220.

Tool	Feature	Fidelity	Portability	Collaboration	Export	License	Guidance
Balsamiq [1]	Sketching	Low	Web or client version for Mac, Windows, Linux	Sharing in almost real time via DropBox	As PNG or XML	Payed	
JustInMind [3]	Prototyping		Runs on Windows, and Mac	Publish and Share with UserNote	As HTML, to Microsoft Word or Open Office	Payed	Emulators to adapt to mobile web apps.
Maqetta [4]	Sketching	Low & high	Web version run across browsers (except IE), (no support for tablets)	Asynchronous	As a zip file	Open Source	Widgets libraries for mobile apps. (Dojo)
SketchFlow [5]	Prototyping	Low & high	Web or client version for Windows		As XAML, or to Microsoft Work	Payed	

Table 1: Analysis of existing UI editors concerning their: features, fidelity levels, portability, collaboration, export formats, license and guidance.

signers and usability professionals can find usability problems at an early stage of design, before substantial resources are invested in flawed designs.

Several tools that support the UI design, sketching and prototyping are available. MAQETTA, for example, is a visual authoring tool for designing HTML5-based UI's, it adopts a WYSIWYG approach. Because MAQETTA is also a web-based application, no plugins or downloads are required. Although it permits the design of UI mockups, context-aware adaptation is not integrated as a feature [4]. Besides MAQETTA, there are also other graphical editors to support activities of sketching and prototyping UI's in varied fidelity levels. Four popular tools that support UI sketching and prototyping have been analyzed for reference (Table 1):

- **BALSAMIQ** [1] to quickly sketch interfaces and to communicate design ideas (balsamiq.com);
- **JUSTINMIND** [3] a platform for defining prototypes for web and mobile applications (www.justinmind.com);
- **MAQETTA** [4] a web-based visual authoring of HTML5 user interfaces, WYSIWYG (maqetta.org);
- **SKETCHFLOW** [6] an editor to create interactive prototypes (www.microsoft.com/silverlight/sketchflow).

Such tools effectively support UI prototyping and sketching, however they do not focus on context-aware adaptation. By using those, designers tend either to ignore context or to manually implement various different UI versions.

For stakeholders considering the significant differences in current contexts of use is challenging to implement interactive systems. *User profiles* vary according to their age, expertise levels, preferences and needs. The *platforms* have different screen dimensions, interaction modalities, and capabilities. The *environments* have different brightness, noise and stability levels. These differences impact the requirements for GUI's, and if certain characteristics are not properly considered the interaction may be hindered or even prevented. However, for designers, it is not scalable to take each context detail into account so that many different versions of the same system are implemented. Currently, to generate context-aware UIs, designers must be aware of the target context, and also properly know how to apply adaptation, for instance by following standard guidelines [7] or well-known patterns [9], [5]. There exist tools to support the UI design and electronic sketching, however they do not provide support to consider context-awareness and design guidelines. Thus, the resulting UI's are not always suitable to their actual contexts, unless several versions of the same UI have been implemented, which among other problems, considerably delay in their time to market. The main shortcomings of such tools regarding context-aware adaptation of UI's are:

Firstly, such tools are already not adapted, being implemented to run in a conventional context of use of an able-bodied user in a Desktop PC in a stable environment (i.e. office). Although practitioners may be more familiar working in their traditional workspace, because the design occurs outside the target context, the results may be inaccurate. For instance, by designing mobile applications in a smartphone, a more realistic result can be achieved, specially concerning screen dimensions and performance constraints.

Second, contextual information is not supported, or it is but limited e.g. to screen sizes. Stakeholders are not aware of *which* information to consider and *how* to consider. No support is provided concerning usability guidelines and patterns.

Third, by being inflexible such tools force users to sketch and to prototype one version of the same application dedicated to each context. When various contexts must be covered it becomes not scalable to properly ensure adaptation.

Finally, although users with different profiles work often in large teams, such tools not always support collaboration, thus either all stakeholders must be co-located or they cannot simultaneously access and edit the project, thus pipelining their tasks. By matching different skills and expertise in a synchronous project collaboration, better results can be obtained.

Besides this, when the context is considered just during late development phases, if the resulting UI's are not satisfactory, significant re-work may be needed. When usability tests and experiments are conducted during later stages, results may be incorrect once a high fidelity prototype is already available, discouraging users to negatively evaluate the resulting UI's.

3. ENHANCING SKETCHING WITH ADAPTATION

Sketching is a powerful tool for the UI design. For [11], the presence of ambiguity in early stages of design broads the spectrum of considered solutions and tends to result in a higher quality design. Fostering creativity is important since design is essentially a problem of *wicked nature*, i.e. the process of solving it is identical with the process of understanding it [14]. In wicked problems, in initial phases the designer does not have a clear understanding of what to produce and has only a vague goal in mind, thus to support the design with an interactive system, it is important that the system does not get "in the way" between the designer and the solution.

GAMBIT offers a flexible setup where users can choose their preferred devices for interacting – i.e. devices that are more suitable for sketching input can be used for pen interaction, while large displays can be used for visualization (Figure 1 middle). By observing UI design sessions on IT companies in Belgium, a cycle of 3 main activities has been identified:

- **Sketching:** one or more participants produce sketches to express the ideas.
- **Sharing:** the participants often share the drawings using a big sheet of paper and post-its. The sheets are arranged as a storyboard on a wall for discussion.
- **Discussing:** the participants refine the sketches based on what their discussion results.

GAMBIT provides a large (infinite) workspace where users can add images and sketch. Parts of the workspace can be shared among different platforms, composing a virtual meeting room. The tool is an essential part of a research on sketching user interfaces, its usage in current design practices also considers the diversity of contexts of use. To do so, users can test the interfaces on the same device in which it is intended to be used. GAMBIT's requirements are presented in [15].

3.1 Methodology

By informing the designer about possible design flaws related to the context of use at the sketching and the prototyping phases could aid to anticipate design flaws and potential usability issues before the interface is ready.

To enhance the sketching activity as it is currently supported towards the effective consideration of context-awareness, we propose to augment it by integrating adaptation guidelines. This methodology involves three main steps (Figure 1):

1) **Describing the Context**. by means of an XML description of the target context, users provide the characteristics the user, the platform and the environment (Figure 1 top).
2) **Sketching the UI's**. the UI's are sketched in the user preferred device(s) (Figure 1 middle). Since not all the devices have the same resolution or performance, each device is suitable for each basic activity (sketching, sharing and discussing) [15, 16]. The UI elements are recognized according to a standard library.
3) **Adapting the UI Sketches**. based on the GUI elements recognized, and in the target context that has been parsed, relevant guidelines are searched, retrieved and and suggested to the end user (Figure 1 bottom).

3.2 Design Decisions

The contexts can be expressed by means of a descriptive XML document, however if the user sketch in the target device, its specification can be automatically verified and retrieved. Other information, as the user profile and the target environment, must be either explicitly expressed by the user, or a conventional scenario of interaction can be adopted.

Algorithms can automatically recognize the UI elements involved in the sketch. Previous works, as UsiSketch [10], for instance, discuss and present algorithms that automatically recognize sketch elements for GUI's. An alternative solution consists in enabling the end user to tag the sketched elements. This approach may reach more accurate results, however it delays the original task. An hybrid approach is also suitable, e.g. enabling the user to associate gestures to the sketch to indicate whether a rectangle represents an image or a button.

Based on the elements recognized in the UI, and the context provided, the system can then automatically search for and present

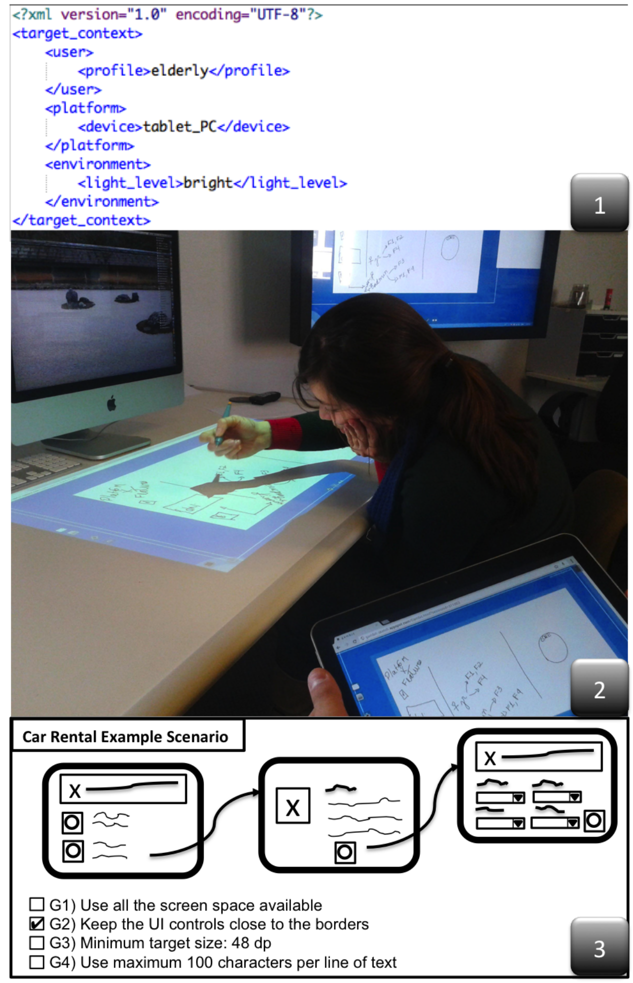

Figure 1: The 3 main interaction steps for the enhanced sketching tool: first the end user provides a description of the target context, then while sketching, the tool automatically recognizes the interface layout and its UI elements, proposing relevant adaptation techniques (as guidelines).

relevant guidelines for the end user. The UI guidelines and patterns can be retrieved from (i) a pre-defined catalog as reports [13], (ii) an online repository (e.g. the W3C accessibility guidelines [7]), (iii) a local repository integrating many sources, or (iv) third-party cloud-based web services.

The UI guidelines cover up to 3 GUI's dimensions: contents, presentation and navigation. The *contents* include the different formats, as images, texts, videos and elements that support the user tasks. Based on the context, such contents can be removed, resized, replaced, etc. The *presentation* concerns the layout of GUI's, i.e. the contents distribution can follow a certain alignment, balance, or specific re-arrangements. And the *navigation* in an interactive system is structured by a task tree (e.g. with CTT [2] or HTA). Although a unique task tree enables the same goals to be achieved in different contexts of use, many groupings and hierarchies are possible to better suit tasks in different scenarios; e.g. concerning the screen dimensions, if the user interacts from a small screen (smart phone), few tasks must be presented at a time.

These dimensions cover 3 abstraction levels (the properties of the UI elements, their layout in the UI and the arrangement among

UI's). Because Gambit also captures the navigation, all 3 levels can be supported (semantic annotations may be needed to aid the selection of relevant guidelines though).

A semi-automatic approach can be applied to suggest changes for the sketches, i.e. once the guidelines' list is presented, the user is in charge of analyzing it, judging whether it is indeed appropriate, modifying the UI, and changing the guideline's status (as checked). It is also important to capture the relevance of the guideline, aiming at improving the methodology performance. Given that design sessions often concern a specific application domain, the suggested guidelines must cover a generic purpose, following specific constraints of context of use and UI elements, but still being domain-independent.

3.3 Case Study

Our case study scenario consists in designing graphical user interfaces for a car rental system, targeting a tablet platform, an elderly user, and an environment with bright light level.

Table 2: The UI guidelines matching context and UI sketch elements.

Context	Content	Presentation	Navigation
User	G1: prefer drag and pinch gestures	G2: provide visual feedback	G3: keep distance between targets
Plat.	G4: use maximum 100 characters per line of text	G5: use all screen space available	G6: keep UI targets at the borders with 8mm at least
Env.	G7: prefer dark tones	G8: ensure contrast	

In our study 3 GUI's are needed to accomplish a user task: first the user accesses a list of cars, then one car is selected and further information about it is presented, finally the dates and location for pickup and delivery are selected. Each car is described by an image and a text. To complete the rental, the user has a confirm button after the car description. By scrolling the UI, the user navigates through the list of cars. Figure 1 bottom shows the UI's and a set of 4 guidelines presented to the user, concerning the tablet platform, and the characteristics of the sketches. Table 2 complements this set, based on the context (user, platform and environment) and on the UI's (content, presentation and navigation) [12], [8].

4. FINAL REMARKS

Although many tools support UI design, they do not integrate UI guidelines. In the current technological landscape the contexts of use can vary significantly. However designers are unaware of *which* and context information to consider and *how* to do it. By integrating UI guidelines in UI sketching tools, stakeholders can be aware of relevant guidelines, generating enhanced UI's. This paper proposes enhanced sketching by means of a novel approach, illustrated with a case study.

5. ACKNOWLEGDEMENTS

Projects: Serenoa (E.C. FP7 (FP7-ICT-2009-5)) and QualIHM (Region Wallonne, Dir. generale operationnelle de l'Economie, de l'Emploi et de la Recherche DGO6).

6. REFERENCES

[1] Balsamiq Mockups: Rapid Wireframing Tool, 2012. `www.balsamiq.com`.

[2] Concur Task Trees (CTT), 2012. `http://www.w3.org/2012/02/ctt/`.

[3] JustInMind: a platform to define web and mobile apps., 2012. `http://www.justinmind.com/`.

[4] Maqetta: Visual Authoring of HTML5 User Interfaces in the browser., 2012. `http://dojofoundation.org/projects/maqetta`.

[5] My UI Pattern Browser, 2012. `myuipatterns.clevercherry.com`.

[6] SketchFlow: Expression Studio Ultimate, 2012. `http://www.microsoft.com/silverlight/sketchflow/`.

[7] Web Content Accessibility Guidelines (WCAG) 2.0, 2012. `http://www.w3.org/TR/WCAG/`.

[8] Android Tablet Guidelines, 2013. `http://developer.android.com/distribute/googleplay/quality/tablet.html`.

[9] Yahoo! Design Pattern Library, 2013. `http://developer.yahoo.com/ypatterns/`.

[10] Coyette, A., Schimke, S., Vanderdonckt, J., and Vielhauer, C. Trainable sketch recognizer for graphical user interface design. In *INTERACT (1)* (2007), 124–135.

[11] Goel, V. "Ill-Structured Representations" for Ill-Structured Problems. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, vol. 14, Lawrence Erlbaum (1992), 130–135.

[12] Kobayashi, M., Hiyama, A., Miura, T., Asakawa, C., Hirose, M., and Ifukube, T. Elderly user evaluation of mobile touchscreen interactions. In *Human-Computer Interaction–INTERACT 2011*. Springer, 2011, 83–99.

[13] Motti, V. G., and Vanderdonckt, J. A Computational Framework for Context-aware Adaptation of User Interfaces. In *Proc. of the Research Challenges in Information Sciences Conference - RCIS '13* (2013), 12.

[14] Rittel, H. Dilemmas in a general theory of planning. *Policy sciences 4*, 2 (1973), 155–169.

[15] Sangiorgi, U. B., Beuvens, F., and Vanderdonckt, J. User Interface Design by Collaborative Sketching. In *Proc. of the Designing Interactive Systems, DIS '12*, ACM Press (Newcastle, U.K., 2012), 378.

[16] Sangiorgi, U. B., Motti, V. G., Beuvens, F., and Vanderdonckt, J. Assessing Lag Perception in Electronic Sketching. In *Nordic Conference on Human-Computer Interaction* (Copenhagen, DK, 2012).