Enhancing Collaborative Sketching with Adaptation Guidelines

Vivian Genaro Motti, Ugo Braga Sangiorgi, Jean Vanderdonckt
Louvain Interaction Laboratory, Université catholique de Louvain
Place des Doyens, 1 - B-1348 Louvain-la-Neuve (Belgium)
first.last@uclouvain.be

ABSTRACT
Designing interactive systems for multiple contexts of use becomes a burden when several interaction scenarios, their characteristics and constraints, must be considered. Stakeholders face then two main challenges: they are not aware of *what* is significantly relevant to consider from contextual information, or *how* to appropriately adapt the user interfaces based on this information considered. Moreover, stakeholders cannot count on existing UI editors once they usually do not provide support for context-aware 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 suggesting to the end user adaptation guidelines. This work aims at raising awareness about context-aware adaptation since the early stages of the UI design.

Author Keywords
Collaborative sketching; Context-aware adaptation;
Collaborative design; Prototyping.

ACM Classification Keywords
H.5.2 User Interfaces: Prototyping

INTRODUCTION
The contexts of use in which users interact nowadays significantly vary in terms of user's profiles, platforms and environments. Due to such variety, it is challenging for stakeholders to properly consider the specificities of dynamic contexts of use while developing user interfaces. As such, usually adaptation is 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 and thus to provide efficient adaptations. As a result, stakeholders often adopt a "one-size-fits-all" approach, providing UIs that are suitable for one single conventional context of use, with such an approach, accessibility issues often raise, hindering or even preventing the user interaction.

Given that it is a challenge for developers to be aware of possible adaptation issues and that adaptation must be considered since the initial stages of the development process, this work proposes the support to context-aware adaptation within user interface editors, aiding stakeholders during all the development phases, but specially starting from low fidelity prototypes. This work also aims at raising awareness for context peculiarities and to the importance of adopting adaptation patterns and guidelines. To do so, a novel methodology for electronic sketching is proposed, i.e. electronic sketching activities must be enhanced based on the target context of use.

In previous works, *multi-platform sketching* has been defined as the activity of drawing with an electronic stylus in different devices while still having the same system running on those different devices [12]. GAMBIT has been developed as a multi-platform collaborative tool for user interface design. GAMBIT enables to sketch UI's and to simulate them on different devices. Now we propose to augment sketching activities by guiding stakeholders through the development of adapted UI's based on the target context. Three main dimensions of context have been defined:

- Multi-platform: sketches can be done on the target device itself, enabling prototyping and testing the UI's on the very device it is intended to run. 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's profiles vary in different aspects, as expertise level (novice, intermediate, advanced) and impairments (visual, cognitive, motor), also different user interface requirements and needs must be considered;
- Multi-environment: for varied constraints and characteristics, different adaptation techniques must be applied. For instance taking into account light and stability levels in order to provide users appropriate interactors.

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

MOTIVATION AND RELATED WORK

The fact that the contexts of use significantly vary challenges stakeholders while implementing interactive systems. The profiles of users vary according to their age, expertise levels, preferences and needs. The platforms have different screen dimensions, interaction modalities, performances and capabilities. The environments have different light, noise and stability levels. Such heterogeneity impacts the requirements for defining GUI's for interactive systems, and the interaction may be even prevented if certain characteristics are not appropriately taken into account. Although some UI's are already adapted to a given context, it is not scalable for designers to take each contextual detail into account in a way that several different versions of the same system are implemented.

To support context-awareness during sketching activities we propose the adaptation guidance to be incorporated into the UI design, first by identifying the context of use, then by suggesting related guidelines that can be adopted by the end user.

The existing tools that support electronic sketching are usually not focused on context-aware adaptation. As such, designers either ignore contextual information or explore different alternatives in a very limited fashion. Nowadays, to generate UIs that are usable in different contexts, designers must be aware of the target context, and then properly know how to apply adaptation, by following standard guidelines or well-known patterns. There exist already many tools available for UI design and electronic sketching, however they currently do not support context-awareness and guidelines. As such, the resulting UI's are not appropriately adapted to their context, or several versions of the same UI must be implemented, causing delay in the time to market, among other problems.

Concerning the UI design, sketching and prototyping, there are several tools available in the market. MAQETTA, for example, is a visual authoring tool for designing HTML5-based user interfaces, 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, a series of alternative graphical editors have been published to support activities of sketching and prototyping UI's in varied fidelity levels. Four popular tools were selected as references; they are listed and briefly described below.

The most popular tools that are currently used for supporting UI Sketching and prototyping are:

- **BALSAMIQ** [1] for sketching interfaces rapidly, and communicate design ideas (www.balsamiq.com);
- **JUSTINMIND** [3] a platform for defining prototypes for web and mobile applications (www.justinmind.com);
- **MAQETTA** [4] a visual authoring of HTML5 user interfaces in the browser, open source, WYSIWYG (maqetta.org);
- **SKETCHFLOW** [5] a UI prototyping tool to create interactive prototypes[1].

[1]http://www.microsoft.com/silverlight/sketchflow/

The main shortcomings of such tools regarding context-aware adaptation of user interfaces, can be summarized as:

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 (e.g. 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 may be achieved, specially concerning constraints as screen dimensions and performance.

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 work usually in 'large' teams, with different profiles, such tools not always support collaboration, thus either all stakeholders must be co-located or they cannot simultaneously access and edit the project, resulting in a pipeline of tasks. By matching different skills and expertise simultaneously collaborating within a project, 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.

ENHANCING SKETCHING WITH ADAPTATION

Sketching is considered to be a powerful tool for designing UI's. As the findings of [8] point out, the presence of ambiguity in early stages of design broads the spectrum of considered solutions and tends to deliver a design of higher quality. Fostering creativity is considered to be 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 [11]. 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, therefore in order to support the design activity with an interactive system, it is important that the system does not get "in the way" between the designer and the solution to be developed.

GAMBIT, as Figure 1 illustrates, offers users a flexible setup where they 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. By observing UI design sessions on belgian IT companies, three main sequential phases have been identified:

1. Sketching: one or more participants produce sketches to express ideas.

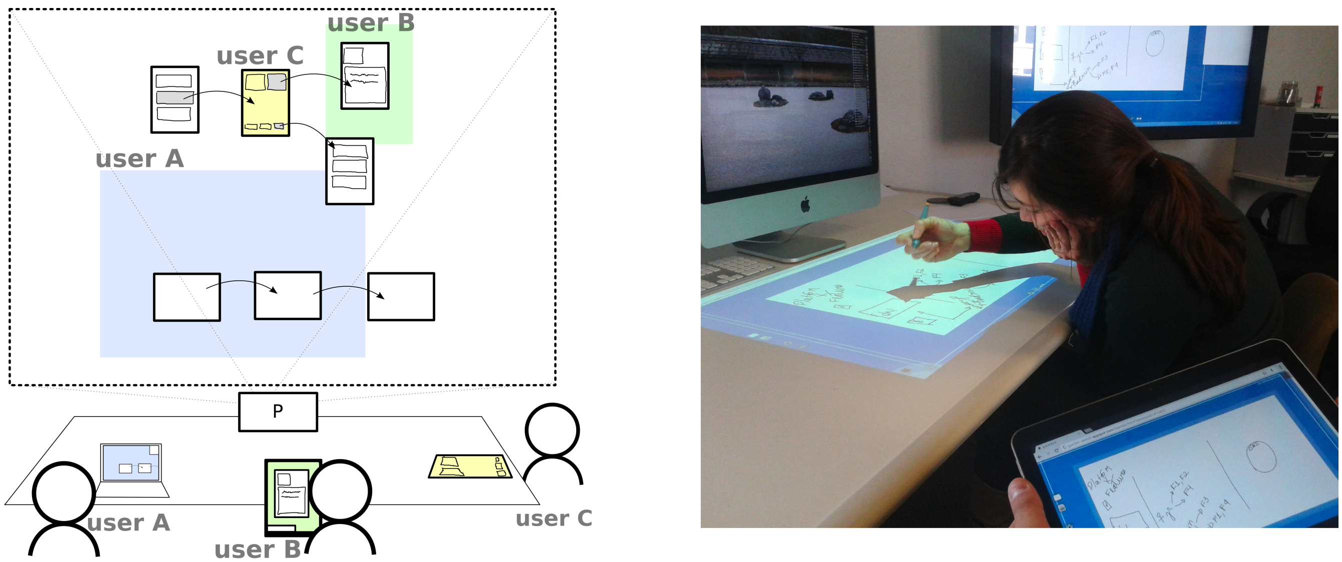

Figure 1. GAMBIT, a distributed sketching system. Left: users A, B and C collaborate in sketching activities, using respectively a laptop, a tablet and an interactive table. Right: 3 synchronized platforms in use for sketching.

2. Sharing: the participants normally share the drawings using a big sheet of paper and use post-its. The sheets are arranged as a storyboard on a wall for discussion.
3. Discussing: the participants refined the sketches based on what was discussed and learned on the discussion.

GAMBIT provides a large (infinite) workspace where users can add pictures and sketch. Parts of the workspace can be shared among different platforms, composing a virtual meeting room. The tool is a fundamental part of a research on sketching user interfaces, its usage in current design practices must also take into account the diversity of contexts of use. In this sense, a designer can test an interface for a system on the same device the system is intended to be used. GAMBIT's basic requirements list was presented in [12].

Aiming to enhance the sketching activity currently supported towards effective consideration of context-awareness, we propose to augment it by integrating adaptation guidelines. To do so, first the end user must provide the description of the target context, then while sketching the tool will recognize the layout of the UI elements and also propose relevant adaptation techniques according to the UI definitions.

Methodology and Design Decisions

For the context of use, it can be provided to the system by means of an XML-based document, containing data about the: user, platform and environment. Once parsered such information will be used later on to search for corresponding guidelines. For the sketch recognition, some algorithms have already been investigated in previous works. UsiSketch [7], for instance, presents an algorithm that automatically recognizes sketches for graphical user interfaces. And concerning the guidelines, an online repository as W3C guidelines [6] can be accessed, a local repository integrating different sources, or third-party web services on the cloud could be connected to the system to provide the corresponding list (based on the target context and the UI elements).

For the UI guidelines, three distinct dimensions of (graphical) user interfaces can be considered, namely:

- *Navigation*: an interactive system is usually structured by means of a task tree (being CTT [2] and HTA often employed). Although one single task tree can enable the same goals to be achieved in different contexts of use, there are several groupings and hierarchies that can be adopted to better accommodate tasks according to different scenarios. For example concerning the screen dimension of the user, if it is a small screen (smartphone) a reduced number of tasks must be presented at a time.
- *Presentation*: the layout of graphical user interfaces can be presented in different ways, the distribution of the contents can follow a certain alignment, balance, or specific certain re-molding actions.
- *Contents*: several formats of contents can be included in the UI composition. Usually images, texts, videos and UI interactors combined provide a task for the user. Depending on the context, such contents can be: removed, re-sized, replaced, etc.

Such dimensions provide an overview about possible adaptations, for a more complete list of techniques other references can be accessed (e.g. [10]).

According to the elements recognized in the UI, and the context provided, the system can then automatically search for and present relevant guidelines for the end user. A semi-automatic approach can be applied to suggest changes for the sketched UI, i.e. once the guidelines' list is presented, the end user is responsible for analyzing it, judging whether it is valid, modifying the UI, and changing the guideline's status (e.g. as checked).

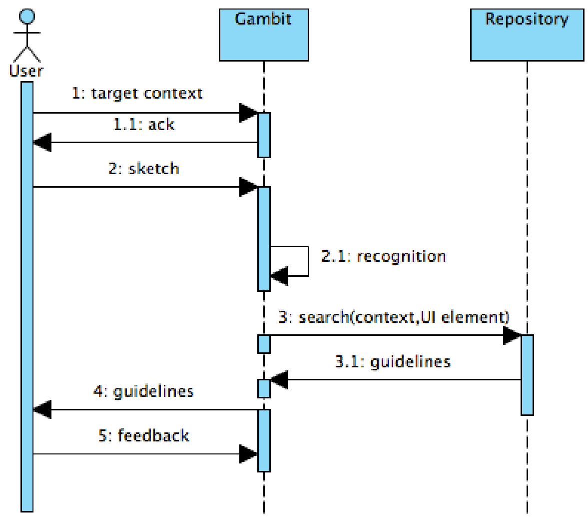

Figure 2. Sequence Diagram

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. Specific requirements of the given methodology include:

- Context as an input: users must provide information about the target context of use
- UI Sketching: users design the UI defining its layout
- Pattern Recognition: the system recognizes UI elements
- Guidelines Suggestions: based on the element previously recognized the system search for, retrieve and present guidelines that match both context of use and UI element
- User Feedback: the user provides his/her feedback about the system suggestion, accepting, rejecting, evaluating

Figure 2 structures such steps as a sequence diagram. The target context (message 1) is provided once, in the start of the design session, sketching (2), recognition (2.1) and guidelines (3 and 4) are repeated until the sketch is concluded.

Scenario

The case study consists in designing graphical user interfaces for elderly users. It has already been proved that older adults perform better and prefer touch screen interfaces. Actually, many studies conducted in this domain, provided valuable insight concerning guidelines for interaction, e.g. [9]. In our study 3 GUI's are needed to accomplish a user task: the user is searching for a movie. First 3 main categories are presented (e.g. drama, sci-fi, comedy), when one of those is selected a list of movies is presented. Each movie contains an image and a short description. For further information, the user needs to click in a link available below the description. To navigate through the list of movies, the GUI can be scrolled. Figure 3 illustrates this scenario. Observe that the scroll bar has been detected as a potential issue for the UI. So the guideline points it to the user, and suggests a new dimension or a new element for the user, improving accessibility.

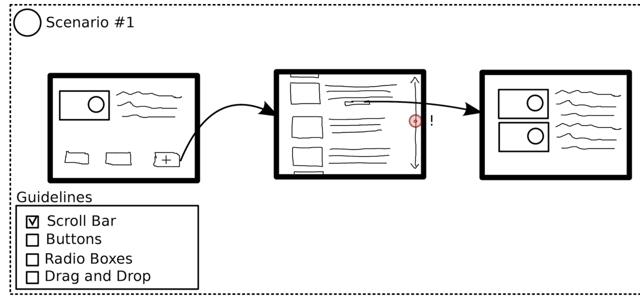

Figure 3. Sketching GUI's for elderly users (storyboard of 3 UI's).

CONCLUSION

Many editors support UI design, but they often exclude UI guidelines. Due to the technological and contextual evolution, designers are unaware of *which* and *how* context information to consider. Thus, most existing UI's ignore adaptation and target one context of use, while contexts vary and evolve. We claim that by incorporating UI guidelines in UI editors, stakeholders can enhance sketching, by being aware of relevant principles, verifying and improving UI's. This paper proposes enhanced sketching activities, by means of a novel approach, and illustrates the proposal with a case study.

ACKNOWLEDGEMENTS

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

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. SketchFlow: Expression Studio Ultimate, 2012. **http://www.microsoft.com/silverlight/sketchflow/**.
6. Web Content Accessibility Guidelines (WCAG) 2.0, 2018. **http://www.w3.org/TR/WCAG/**.
7. Coyette, A., Schimke, S., Vanderdonckt, J., and Vielhauer, C. Trainable sketch recognizer for graphical user interface design. In *INTERACT (1)* (2007), 124–135.
8. 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.
9. 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.
10. Motti, V. G., and Vanderdonckt, J. A Computational Framework for Context-aware Adaptation of User Interfaces. In *Proceedings of the Research Challenges in Information Sciences Conference on - RCIS '13* (2013), 12.
11. Rittel, H. Dilemmas in a general theory of planning. *Policy sciences 4*, 2 (1973), 155–169.
12. Sangiorgi, U. B., Beuvens, F., and Vanderdonckt, J. User Interface Design by Collaborative Sketching. In *Proceedings of the Designing Interactive Systems Conference - DIS '12*, ACM Press (Newcastle Upon Tyne, United Kingdom, 2012), 378.