

# GAMBIT: Addressing Multi-platform Collaborative Sketching with HTML5

**Ugo Braga Sangiorgi**

Louvain Interaction Laboratory, Université  
catholique de Louvain  
Pl. Place des Doyens, 1 - B-1348  
Louvain-la-Neuve (Belgium)  
ugo.sangiorgi@uclouvain.be

**Jean Vanderdonckt**

Louvain Interaction Laboratory, Université  
catholique de Louvain  
Pl. Place des Doyens, 1 - B-1348  
Louvain-la-Neuve (Belgium)  
jean.vanderdonckt@uclouvain.be

## ABSTRACT

Prototypes are essential tools for design activities since they allow designers to realize and evaluate ideas in early stages of the development. Sketching is a primary tool for constructing prototypes of interactive systems and has been used in developing low-fidelity prototypes for a long time. The computational support for sketching has been receiving a recurrence of interest in the last 45 years and again nowadays within the mobile web context, where there are diverse devices to be considered.

The tool presented on this paper was built with HTML5 and Javascript in order to run on any device with browsing capabilities, for the main purpose of aiding an investigation on addressing issues of multi-platform collaborative sketching.

## Author Keywords

Electronic sketching; Multi-platform systems; Collaborative design; Prototyping.

## ACM Classification Keywords

H.5.2 User Interfaces: Prototyping

## INTRODUCTION

A prototype is a working model built to develop and test design ideas. In User Interface design, prototypes are essential tools for fostering discussion regarding both interface and interaction with stakeholders. Techniques such as Paper Prototyping [17] became very popular due to its' low cost and efficiency, since they are made using sketches for quick generation of prototypes.

Lately, there has been a recurrence of interest on supporting electronic sketching (i.e. sketching at electronic devices and interactive surfaces), and despite sketching *recognition* to be fairly well addressed on the literature, the adoption of electronic sketching as a *design tool* is still a challenge [7].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EICS'12, June 25–26, 2012, Copenhagen, Denmark.

Copyright 2012 ACM 978-1-4503-1168-7/12/06...\$10.00.

The current popularization of touch screen devices and the multi-platform capabilities made possible by using HTML5 might pose new opportunities for researchers to explore, for instance, how designers use sketching to prototype interfaces for a target device by producing and testing them on the device itself.

When designing, people draw things in different ways, which allows them to also perceive the problem in new ways. People engage in a sort of *conversation* with their sketches in a tight cycle of drawing, understanding, and interpreting [15]. When prototyping by using the target device, other elements come into play in that conversation. Nowadays, there are many devices available for designers to sketch upon [9], with different screen sizes, weight and processing capabilities; this is a fact to be addressed in current sketching research.

We introduce GAMBIT, a multi-platform collaborative tool for User Interface design that allows the sketching and simulation of UI's on many different devices.

The tool is an essential part of a research on sketching, whose goal is to investigate electronic sketching usage in current UI design practices taking into account the multi-platform context for producing and validating interactive prototypes. The system's core concepts are:

1. Sketch-based - electronic sketching is supported as the main mode of interaction, it is used to quickly put ideas on an external medium, where they can be discussed, improved and stored for further reference [4];
2. Multi-platform - for it allows users to sketch using the device of their preference, and also allows the prototyping and testing of systems on the very device it is intended to run. The system was built with HTML5 and Javascript in order to run on any device with browsing capabilities, through a browser or embedded into a native application;
3. Collaborative - for it focus on group sessions, allowing not only designers to sketch and discuss together, but also to include end users in the process.

This paper is organized as follows: in the next section we motivate the system construction together with the related works. Section 3 presents the requirements gathered in order to support design sessions. Section 4 presents the system's architecture and we conclude on Section 5.

## SKETCHING IN USER INTERFACE DESIGN

Sketching is considered to be a powerful tool for doing design. As the findings of [5] point out, the presence of ambiguity in early stages of design broadens the spectrum of solutions that are considered and tends to deliver a design of higher quality.

As for the fundamentals of sketching, we could mention Van der Lugt's work [18] who conducted an experiment to analyze the functions of sketching in design, in which participants produced individual sketches and then presented them for the group for discussion. Three primary sketching functions were identified:

- F1** Sketching stimulates a re-interpretive cycle in the individual designer's idea generation process: Schon and Wiggins [14] describe design as a cyclic process of sketching, interpreting and taking the sketches further.
- F2** Sketching stimulates the designers to re-interpret each other's ideas: when sketching to also discuss (as opposed to sketch for self-interpretation), the designer invites others to interpret her drawings as well. The function of inviting re-interpretation is especially relevant for the idea generation process, as re-interpretation leads to novel directions for generating ideas [18].
- F3** Sketching stimulates the use of earlier ideas by enhancing their accessibility: Since it is externalized, sketching also facilitate archiving and retrieval of design information.

UI design by sketching is recognized for several proved virtues such as, but not limited to: maintaining an informal representation to foster creativity [3, 11, 10], complementarity between paper and pencil and software [1, 18], capability to take one design idea at a time and work it out in details or consider alternative designs at a time (i.e. lateral transformation [10]), ability to reveal as much usability problems as if it was a real UI [6].

In order to support sketching into UI design, we needed to analyze the process in which UI design is included. Currently, the development life cycle of interactive applications consists of a sophisticated process that does not always proceed linearly in a predefined way. The tools available for UI development are usually not focused on UI **design**, in which designers usually explore different alternatives but in UI **modeling** as a final product, where designers must attend to formal standards and notations.

There are many tools available for both modeling and design, however practitioners are currently forced to choose formal and flexible tools. Whichever they choose, they lose the advantages of the other, with attendant loss of productivity and sometimes of traceability and quality.

As the study reported in [2] mentions, designers desire an intelligent whiteboard because it would not require hard mental operations while sketching during meetings and design sessions.

However, electronic sketching is still behind the classical sketching in paper, since the tool in use becomes too evident

[19]. Perhaps until the gap between displays and paper are minimized, (for instance with paper-like displays [16]), this distance will continue high, hindering the designer's *conversation*.

A great care must be taken to support the designer's reflection when making design software that employs sketch recognition, for instance. If the system interprets drawings too aggressively or at the wrong time, it may prevent the designer from seeing alternative meanings.

Calico [10] and DENIM [11] are good examples of "vanishing tools" since they keep out of the way between the designers and the problem at hand, and this can be useful especially during early design stages.

Therefore, fostering creativity is specially 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 [12]. In wicked problems, the designer does not have a clear understanding of what to produce and has only a vague goal in mind in the beginning.

However, electronic sketching has some important advantages over classical 'pen and paper' approach. While sketches are useful to facilitate discussions on the conceptual level, computer prototypes are useful for discussing operational and interaction issues [6]. Thus, raw sketches and interactive prototypes are complementary.

One important issue with currently sketch-based systems for prototyping of user interfaces is that they are *single-platform*, since they are usually made to be used on Desktop computers [11, 10], even though the prototypes are targeted at multiple devices [8].

A designer could sketch and test interfaces for many platforms using just a single platform such as a large sketching device (e.g. Wacom, TabletPC). However, the main benefit of sketching as a prototyping technique is to allow us to 'see as' and 'see that' [15]. That benefit is hindered since only the size of the target device is being considered, while there are other significant factors such as weight, screen resolution, brightness and interaction modes (e.g. multi-touch, WIMP).

We argue that a more complete prototyping system would allow sketching and simulation on the target device, enriching both designers' and users' experience with an interactive prototype, allowing them finally to have a richer *conversation* with the working design at hand.

## REQUIREMENTS

We have observed design sessions conducted in two companies related to user interface development. The people involved on those sessions were designers, project managers, programmers and frequently stakeholders. In overall, in these companies the design sessions are usually done around a central topic, about which people discuss in order to produce some artifact, usually a report with a list of requirements, wireframes and some session log of the decisions made around the interaction. It is important to note that this report is not produced on site but after the meeting, for what

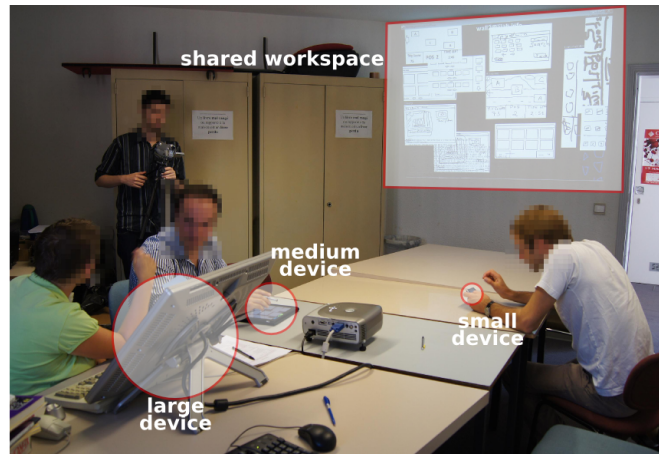
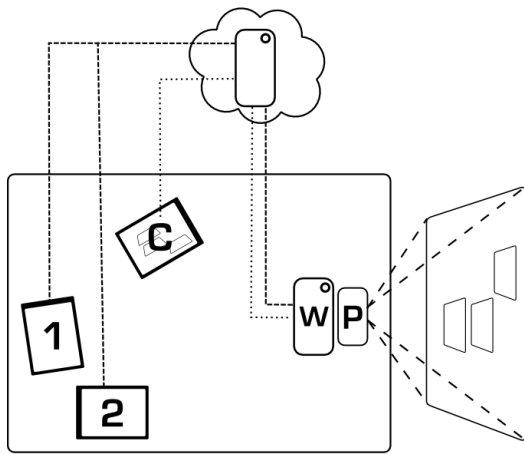


Figure 1. Physical setup of GAMBIT.

people usually take pictures for remembering and registering what was discussed. Nevertheless, the design sessions most often proceeded with three distinct phases:

1. Mental model construction and concepts: the mediator leads the task, asking the participants the essential elements of the tasks.
2. Scenario construction: the participants are usually divided into groups to focus on one scenario each. They usually do it using a big sheet of paper and use post-its. After each group agrees on its own scenario, the sheets are arranged as a storyboard on a wall for discussion.
3. Interface prototyping: the participants' sketches the user interface based on what was discussed and learned on the scenarios discussion.

Based on Van der Lugt's work [18] and on the observation of the design session, a preliminary list of requirements for a system to support collaborative sketching was constructed as follows:

- R1** Support sketch production and visualization on different devices;
- R2** Support session storage and retrieval;
- R3** Support private/public production of sketches;
- R4** Provide a broad view of the drawings (like papers arranged on a wall);
- R5** Provide a fine view of a drawing;
- R6** Support the UI design with different level of fidelities;

### GAMBIT SYSTEM

The tool support for the investigation is the GAMBIT system, a distributed software environment designed to be physically deployed around a table, with tablets and a projector. It is multi-platform since it is essentially an embedded website, which might be used through a browser or through a native mobile application (i.e. a 'wrapper' application).

The system is currently developed as depicted on Figure 1: the many input devices (1 and 2 in the figure) can be tablets, mobile devices, large graphical tablets, etc. They are used by designers to sketch and submit drawings to the device representing the wall (W) showing the sketches as if they are real sheets of papers organized onto a real wall.

The wall is projected using a common projector (P) and can be controlled using a tablet, called 'control tablet' (C). The roles of the devices are interchangeable – a user might request the wall's control at any time, organizing and grouping the sketches. Since GAMBIT is a web-based system operating through a browser, the wall (W) might be a full-screen browser window opened on a desktop computer, a projection or a large interactive display.

Figure 1 (left) shows the deploy scheme of the system, with designers using different devices each around a projector in the middle. In the right part the wall shows the sketches being organized with the control tablet.

Figure 1 (right) shows a picture taken during a preliminary study (not to be described on this paper) with designers from one software development company. That experiment showed some indications regarding designers' preferences of device types for each one of the requirements, and it is one of the expected outcomes of the tool.

The system was developed in HTML5 in order to centralize the code for different platforms. In this sense, the system can run on any device with a browser. The sketch interface of the system is showed on Figure 2, with a drawing area that uses HTML5 <canvas> element and Javascript routines to capture the mouse/pen/touch events.

The left part shows a toolbar that can be used to switch from sketching to control functionalities. Figure 3 shows the wall with the sketches arranged like sheets of paper that can be dragged and grouped. The black background is intentionally put in order to make only the "sheets" to be projected on the wall, so as to mimic the physical storyboard mentioned during the interviews. The wall is the main part of the system, since the design session progresses around it.

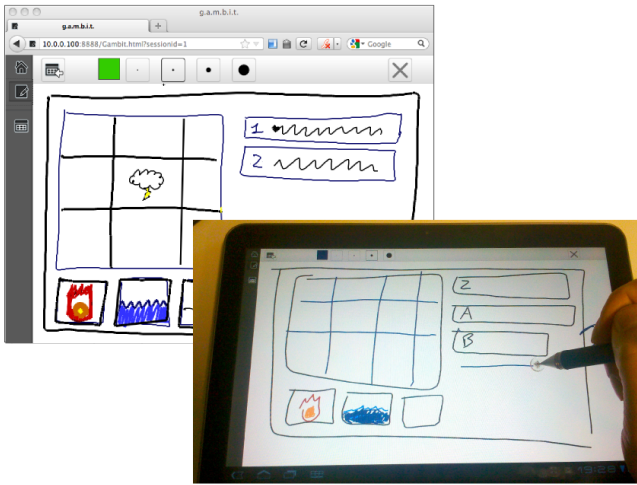


Figure 2. GAMBIT interface for sketch production respectively on a desktop and a tablet.

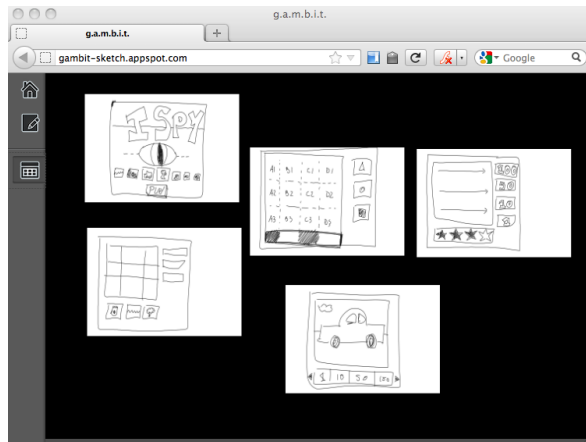


Figure 3. Interface for overview or "Wall sharing" on big displays

GAMBIT was logically designed as depicted in Figure 4, with many HTML5 clients running on tablets, smartphones and/or desktops, while being managed by a cloud-based web server. The various components of this software architecture are further explained in the next sub-sections.

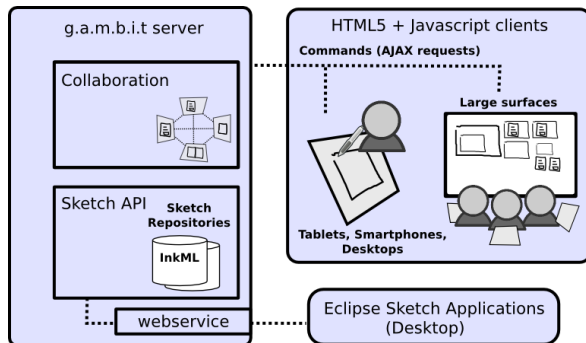


Figure 4. The logical components of GAMBIT

**HTML5 + Javascript clients** The clients run the same application frontend through a browser or a wrapping applica-

tion that only displays the website (without browser controls). The communication with the server is made through asynchronous requests via Javascript (AJAX). The designers might choose the device that better suit their needs, for instance, a designer might want a light device such as a tablet for a face-to-face meeting with a client in order to start a design, and a large interactive surface for online meetings or solo work.

**GAMBIT server** The server is responsible for managing the users, their collaborative activities and their sketches. Also, it relies on a sketch recognition API that is originally part of Eclipse Sketch project [13]— an Eclipse project created to add sketching capabilities to meta model editors built with Eclipse.

**Sketch (recognition) API** This component processes the sketch separately from the rest of the system, in a way referred in the literature as 'lazy or postponed recognition', which means that the sketches are not actively recognized and replaced by high-fidelity versions of what the algorithm interpreted the sketch to be. This is important in order to maintain the original look of the sketch, without stopping the designer's creativity flow and *conversation* with the sketch [7, 15]. The sketches are stored in InkML format (<http://w3.org/TR/InkML>).

GAMBIT is original with respect to the state of the art in that it supports user interface design by combining the following preminent features: multiple stakeholders, multiple input devices, multiple output devices, multiple levels of fidelity, multiple ways on multiple computing platforms, thus supporting multiple configurations in a flexible way. Regarding the requirements, its current state is outlined below:

**R1 Support drawing sharing, visualization and consequently discussion:** The wall device acts as a sharing repository of sketches, aiding the discussion around a design. It is possible to send sketches to the wall, organize them, put them side-by-side for comparison, etc.

**R2 Support session storage and retrieval:** Sessions storage are supported, and can be loaded, saved and continued. History support is also planned.

**R3 Support private/public production of sketches:** Each input device is able to produce live sketches or to produce a sketch separately for later publication on the wall. Sub-group collaboration of two or more participants to produce a sketch is planned, but yet to be supported.

**R4 Provide a broad view of the drawings:** the wall was designed to serve exactly as a physical wall with 'projected sheets of paper', which are the images and sketches.

**R5 Provide a fine view of a drawing:** the input device can serve as a fine view of any sketch, and they can be re-drawn and sent once again to any other device.

**R6 Support the UI Design with different levels of fidelity:** For the moment, only low fidelity is supported.

## CONCLUSION

We introduced GAMBIT (Gatherings and Meetings with Beamers and Interactive Tablets) as a system for investigating electronic sketching on a multi-platform collaborative context. It is a system aimed at constructing prototypes and will evolve to be used at design sessions in companies with a low cost of deployment, which is expected to ease the process of performing experiments.

The tool is a fundamental part of a research on sketching, whose goal is to advance the state of the art in electronic sketching, and its usage in current design practices taking into account the diverse multi-platform context.

By producing and using the prototypes directly on the target device, we expect the designers to have a richer *conversation* with the working design at hand. Experiments will be conducted comparing prototypes produced with tools such as DAMASK [8] (a single-platform tool for multi-platform prototyping) and our tool.

The system currently supports drawing and sharing in many platforms, and will evolve to allow the construction of interactive prototypes in a way similar to DENIM [11], but also including post-WIMP functionalities, with interactive regions for touch/click events and flow connections between them.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the ITEA2-Call3-2008026 UsiXML (User Interface extensible Markup Language) European project and its support by Région Wallonne, Direction générale opérationnelle de l'Économie, de l'Emploi et de la Recherche (DGO6).

## REFERENCES

1. Bailey, B. P., and Konstan, J. A. Are informal tools better?: comparing DEMAIS, pencil and paper, and authorware for early multimedia design. In *CHI'03: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (New York, NY, USA, 2003), 313–320.
2. Cherubini, M., Venolia, G., DeLine, R., and Ko, A. J. Let's Go to the Whiteboard: How and Why Software Developers Use Drawings. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (2007), 557.
3. Coyette, A., and Kieffer, S. Multi-fidelity Prototyping of User Interfaces. *Ifip International Federation For Information Processing* (2007), 150–164.
4. Craft, B., and Cairns, P. Sketching sketching: outlines of a collaborative design method. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, British Computer Society (2009), 65–72.
5. 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.
6. Johansson, M. A case study of how user interface sketches, scenarios and computer prototypes structure stakeholder meetings. *People and Computers XXI - HCI... but not as we know it: Proceedings of HCI '07* (2007).
7. Johnson, G., Gross, M. D., Hong, J., and Yi-Luen Do, E. Computational Support for Sketching in Design: A Review. *Foundations and Trends® in Human-Computer Interaction* 2, 1 (2008), 1–93.
8. Lin, J., Landay, J. A. J., Berkeley, U. C., and L, J. A. Damask: A tool for early-stage design and prototyping of multi-device user interfaces. In *In Proceedings of The 8th International Conference on Distributed Multimedia Systems (2002 International Workshop on Visual Computing)*, In In Proceedings of The 8th International Conference on Distributed Multimedia Systems (2002 International Workshop on Visual Computing (2002), 573–580.
9. MacLean, S., Tausky, D., Labahn, G., Lank, E., and Marzouk, M. Is the iPad useful for sketch input? A comparison with the Tablet PC. *EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling* (2011).
10. Mangano, N., Baker, A., and van der Hoek, A. Calico: a prototype sketching tool for modeling in early design. In *MiSE '08: Proceedings of the 2008 international workshop on Models in software engineering*, ACM (New York, NY, USA, 2008), 63–68.
11. Newman, M., Lin, J., Hong, J., and Landay, J. DENIM: An informal web site design tool inspired by observations of practice. *Human-Computer Interaction* 18, 3 (2003), 259–324.
12. Rittel, H. Dilemmas in a general theory of planning. *Policy sciences* 4, 2 (1973), 155–169.
13. Sangiorgi, U., and Barbosa, S. D. J. SKETCH: Modeling Using Freehand Drawing in Eclipse Graphical Editors. In *FlexiTools: Workshop on Flexible Modeling Tools at the 32nd ACM/IEEE ICSE Intl. Conf. on Software Engineering*, (Cape Town, South Africa, 2010).
14. Schön, D. A. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1983.
15. Schon, D. A., and Wiggins, G. Kinds of seeing and their functions in designing. *Design Studies* 13, 2 (1992), 135–156.
16. Shah, J., and Brown, R. M. Towards electronic paper displays made from microbial cellulose. *Applied microbiology and biotechnology* 66, 4 (Jan. 2005), 352–5.
17. Snyder, C. *Paper prototyping: the fast and easy way to design and refine user interfaces*. Morgan Kaufmann, 2003.
18. van der Lugt, R. Functions of sketching in design idea generation meetings. *Proceedings of the fourth conference on Creativity cognition - CC '02* (2002), 72–79.
19. Weiser, M. The computer for the 21st century. *Scientific American* (1991).