Interactive Prototyping of Tabletop and Surface Applications

Tulio de Souza Alcantara
University of Calgary
Calgary, Canada
tuliosouza@gmail.com

Jennifer Ferreira
University of Calgary
Calgary, Canada
jen.ferreira@ucalgary.ca

Frank Maurer
University of Calgary
Calgary, Canada
frank.maurer@ucalgary.ca

ABSTRACT

Physically large touch-based devices, such as tabletops, afford numerous innovative interaction possibilities; however, for application development on these devices to be successful, users must be presented with interactions they find natural and easy to learn. User-centered design advocates the use of prototyping to help designers create software that is a better fit with user needs and yet, due to time pressures or inappropriate tool support, prototyping may be considered too costly to do. To address these concerns, we designed ProtoActive, a tool for designing and evaluating multi-touch applications on large surfaces via sketch-based prototypes. Our tool allows designers to define custom gestures and evaluate them without requiring any programming knowledge. The paper presents the results of pilot studies as well as in-the-wild usage of the tool.

Author Keywords

Prototyping tool; gesture definition; NUI

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Interaction styles, prototyping

INTRODUCTION

Designing *Windows, Icons, Menus and Pointers* (WIMP) based applications is a well-known challenge. This challenge becomes even greater when designing for touch and gesture based applications [3, 5, 6, 8, 9, 28] due to different size, orientation and ways to interact with these gesture based applications. The increasing availability of multi-touch tabletop and surface computing opens up new possibilities for interacting with software systems. Interactions with multi-touch surfaces through gesture and touch-based interactions can either improve or hamper the user experience [1, 2, 3]. When creating gestures for interacting with *Interactive Tabletop and Surface* (ITS) applications, interaction designers have to determine if users consider them natural, understandable and easy to use [6]. These gestures might drastically hamper the user experience if *Human Computer Interaction* (HCI) principles are not taken into consideration [2]. Designers should be able to follow HCI principles not only for the designing how ITS applications look but also for designing how these applications can be interacted with. Previous research on gesture-based interaction has shown problems with the design of the interactions, the meaning of touch and gestures and how context influences them [3, 5, 6, 8, 9, 28].

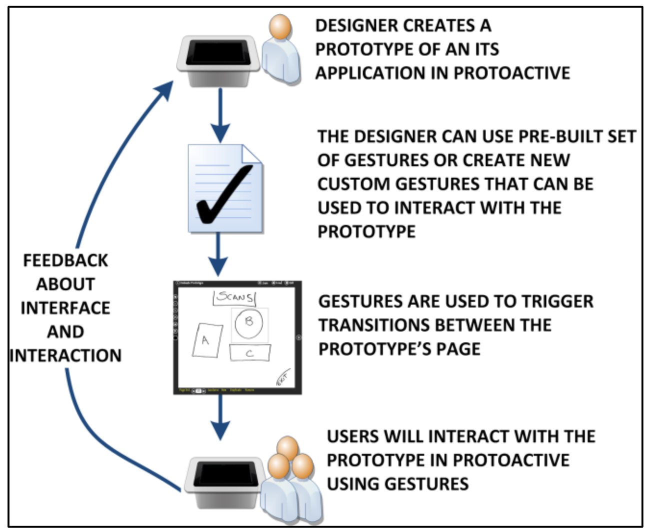

Figure 1 ProtoActive evaluation cycle

There are two main challenges with gesture design:

- the cost (effort, time and technical expertise) required to create gestures [10, 12, 13];
- the design of gestures that are suitable for specific tasks, context and users [5, 9].

The current state of design of multi-touch applications lacks processes and tools to support the design of the interaction of these applications [3, 5, 14]. Studies show a lack of proper tools and methods (such as user-centered design) to improve the design of multi-touch applications [3, 5, 28]. Having users involved early in the process through the use of prototypes has been widely researched and the advantages of sketching and prototyping to improve the design of applications has proved successful [4, 14, 17, 18, 19, 20, 21, 22, 23, 24, 25, 32]. Paper prototypes allow designers to evaluate the output of a system, while the input (how the user interacts with a system) is assumed to be obvious; they allow designers to evaluate what users want to do. With non-trivial interactions, paper prototypes are not sufficient [17, 22].

Tool support for ITS application design has to make it easy to design gestures, respecting the time and cost constraints of prototyping, and make it easy to evaluate the usability of those gestures. The aim of our research is to provide designers with a tool that will help them deliver highly usable ITS applications. In order to do so, we developed a sketch-based tool for evaluating ITS interfaces and interactions in the context of the tasks in the application. This paper presents two contributions: first, ProtoActive, which is composed of

- a storyboarding sketching tool, designed to draft prototypes for ITS applications, based on *Active Story Touch* (AST) [35];
- an integrated gesture learning tool – *Intelligent Gesture Toolkit* (IGT) that uses a specialized anti-unification method[41] to create gesture definitions based on samples of gestures;
- a tool that emulates ITS applications and allows designers to gather feedback on their usability.

Second, this paper presents studies that evaluate the benefits and limitations of ProtoActive. Figure 1 shows the design cycle supported by our tool: a designer creates a prototype of an ITS application and defines how a user will interact with it. The designer can then evaluate the design with the interactions built into it, gathering early user feedback about the application. The whole cycle shown in Figure 1 can be performed without any programming effort or any of the associated costs of programming (e.g.: setting up a work station or managing source code).

To gather requirements for ProtoActive (as a sketch-based prototyping tool), the authors used existing research about computer-based prototyping tools, problems found in existing tool support for prototyping and a qualitative study that consisted of semi-structured interviews with five *User Experience* (UX) designers from different companies.

BACKGROUND

This background section will explain the application domain of the tool proposed here. This will be followed by defining our prototyping tool based on the current taxonomy of prototyping. Finally, the authors provide a definition of gestures and tangibles that will be used in this paper.

Application domain and technical aspects

ProtoActive is a prototyping tool designed to help designers create and evaluate ITS applications and their interactions. In this paper these interactions will be referred to as *gestures* and are described in further detail in the following section. ProtoActive is currently supported by Microsoft Surface 1, Microsoft Surface 2 (PixelSense) and tabletops, tablets or desktop computers with touch enabled monitors, running Microsoft Windows 7 or Microsoft Windows 8. ProtoActive's gesture learning and recognition mechanisms use a modified version of Gesture Toolkit and its *Gesture Definition Language* (GDL) [10].

Gestures

The gestures defined with IGT, differently from stroke based gesture recognizers such as $N [11] and RATA [13], contain information about order, direction and time, which allows IGT to recognize both types of gestures (static and dynamic). In this paper, gestures are considered as single or multi-touch 2D touches; hand postures; detection of fiduciary markers; and concurrent interactions that occur on the surface of an ITS device. Gestures do not include interactions above or in front of a display that do not touch the surface.

Tangibles

The detection of fiduciary markers allows the definition of a subset of interactions: tangible interactions. By allowing designers to define gestures that incorporate fiduciary markers, ProtoActive allows designers to prototype the detection of physical tangibles. By attaching a fiduciary tag to a physical object, a designer can use it as a tangible in ProtoActive and have users evaluate the physical tangibles. Figure 7 illustrates tangibles that can be detected on ProtoActive by their fiduciary markers. Any physical object with a flat surface where a fiduciary tag can be attached can be used as a tangible in ProtoActive.

Prototypes

Buxton [24 p.139] makes a distinction between sketches and prototypes as having different purposes due to the difference between the time spent on them; even though both are tools that can be used in early stages of the design, sketches are earlier drafts whereas prototypes are created later in the design process when ideas are starting to converge. Having more sophisticated and interactive sketches allows designers to take advantage of having users involved and providing feedback about the interactions in ITS applications. Rudd *et al.* [27] suggests advantages and disadvantages of low and high-fidelity prototypes. The design of ProtoActive aims to leverage the advantages of low-fidelity prototypes and it proposes to address the disadvantages: this research aims at providing a prototyping tool that incorporates interactivity at the level of high-fidelity prototypes allowing usability tests based on interaction but having the low effort cost of low-fidelity prototypes allowing the evaluation of multiple design and interaction concepts.

Prototypes created in ProtoActive are sketch based with a low *level of visual refinement* following Buxton's [24] principle that prototypes with a low level of detail and refinement encourage users to provide more feedback. In later stages of design, ProtoActive can also be used in collaboration with image editors and tools to create prototypes with a higher level of visual refinement. Different *breadths of functionalities* [20] can be covered with ProtoActive as it allows designers to easily create several pages in the prototype that can cover a wide range of functionality. Different *depths of functionality* [20] can also be achieved with ProtoActive; in the same fashion that a whole application can be prototyped using ProtoActive, a single task or behavior can be designed in depth and evaluated using ProtoActive. ProtoActive allows a *rich interactivity* [20] evaluation as it allows designers to create their own interaction techniques through custom gestures and use them to interact with the prototype, simulating the behavior of the application through page transitions triggered by gestures. In ProtoActive, there is no mechanism to communicate with any form of *data model* [20]; this was done to keep designers from spending too much time with details such as populating data sources.

Lim *et al.* [21] focused on the support for design exploration in the prototype. This study reveals two key dimensions: *prototypes as filters* (leading the creation of meaningful knowledge about the final design as envisioned in the design process) and *prototypes as manifestations of design ideas.* ProtoActive allows designers to create interaction-based prototypes, where the design and visual details can be evaluated along with the gestures used to interact with the application.

RELATED WORK

Research on design of ITS applications

Hesselmann and Boll propose *Surface Computing for Interactive Visual Applications* (SCIVA), a user-centered and iterative design approach addressing some challenges in designing ITS applications [3]. Their design process gives a general overview of the most important aspects in design of ITS applications. The solution in this paper provides a tool suite that allows designers to follow three steps of the SCIVA design process: defining manipulation functions, conducting user studies to create gestures and evaluating the system with the user to detect flaws from previous steps.

Studying ways to interact with tabletops, Hinrichs and Carpendale found that the choice and use of multi-touch gestures are influenced by the action and social context in which these gestures are performed, meaning that previous gestures and the context of the application influence the formation of subsequent gestures [5]. Also supporting the contextualization of interaction is Krippendorff [23] highlighting that design is not only about making things but also about making sense of things. Both studies suggest that to evaluate interactions it is necessary to contextualize them in the scenario that they will be used.

Trying to understand users' preferences for surface gestures, Morris *et al.* [28] compare two gesture sets for interactive surfaces: one created by end-user elicitation and one authored by three HCI researchers. Their results showed that their participants had similar gesture preference patterns and these preferences were towards physically and conceptually simple gestures. The most popular gestures were designed by larger sets of people, even though the participants did not know who or how many authors created the gesture. Their findings suggest that participatory design methodologies involving user input should be applied to gesture design, such as the user-centered gesture elicitation methodology.

Studying the inconveniences that can be generated by touch based interactions, Gerken *et al.* [29] focus on how users compensate for conflicts between non-interactivity and interactivity created by unintended touch interaction when using a multi-touch enabled tabletop. They conclude that touch-enabled devices can lead to "*touch-phobia*", reducing pointing and leading to less efficient and fluent communication. Their suggested solution is to make touch smarter and more context-aware, which supports the need for better design principles in the creation of touch and gesture-based interactions.

Norman and Nielsen [2] highlight the new concerns that should be addressed by designers when creating touch-based interfaces and ways of interacting with them. The authors propose a balance between creative means of interacting while preserving basic HCI principles, but guidelines for processes that can help designers follow a user centered design approach in the development of ITS applications are limited [3]. Hence, there needs to be a way to evaluate the usability of gesture-based applications in early stages of the design, to preserve HCI principles and have users involved in early stages of the design. There is a need for a tool that supports designers following a user-centered approach.

In order to help the prototyping of tangible tabletop games, Marco *et al.* propose *ToyVision* [15]. *ToyVision* is a toolkit that helps the prototyping of tangible tabletop games. It utilizes a practical implementation of the tangible user interface description language (TUIDL) [16] to model the playing pieces in an XML specification. This toolkit offers a valuable aid in designing tangible based applications as it proposes a classification of the tangibles (tokes) that can be used in the prototypes, which expands the interaction options for tangible based prototypes. To the contrary of ProtoActive, this toolkit requires programming effort.

Research on prototyping

Derboven et al. show the importance of creating prototypes for ITS applications [22]. Their study introduces two prototype methods for multi-touch surfaces. By comparison, their approach consists of physical materials such as paper, cardboard and markers, while our approach proposes an ITS tool, allowing users to create and evaluate the prototypes on the device the applications are designed for.

Sefelin *et al.* [18] compare paper prototyping with prototyping using software tools. Their study suggests three scenarios where paper prototyping would be a preferable medium: when the available prototyping tools do not support the components and ideas which a designer wants to implement; when a designer does not want to exclude members of the design team who do not have sufficient software skills; and when evaluations can lead to a big amount of drawings, which then can be discussed inside the design team. ProtoActive allows designers to create free-hand sketches on a drawing canvas that allows designers to better explore their creativity. In order to simulate the paper-based experience, ProtoActive has an interface that allows designers to create prototypes without requiring much time to learn to use the application, reducing required expertise.

Drawbacks of current prototyping tools

A drawback among the current prototyping tools is the lack of customization of interactions. ProtoActive provides a set of pre-built gestures that can be expanded by allowing designers to provide samples of a gesture to create new gesture definitions that can be used to interact with the prototypes. This feature was implemented to overcome drawbacks from the following tools: *CrossWeaver* [30], *Balsamiq Mockups* [36] and *Fore UI* [39].

Prototyping tools that allow custom interactions also come with the cost of requiring a programming step for customization. This was seen as a drawback as it adds to the cost of prototyping. ProtoActive does not require any programming effort: creating prototype pages, linking them through gestures, creating custom gestures and evaluating them can be accomplished in ProtoActive through its GUI. The need for this feature was gathered from limitations of the following tools: *Raptor* [31], *Sketchify* [32] and *Microsoft Sketch Flow* [38]. By allowing designers to sketch in a similar fashion as sketching on paper, ProtoActive allows designers to create interfaces that are not constrained by a pre-built set of controls. Among the tools studied, another limitation was the lack of a feature that allows designers to free-hand sketch pages. Having pre-built UI widgets might increase the productivity and the speed of creating prototypes, but this comes at the cost of constrained creativity, especially of concern for the design of ITS applications that is a field that is still evolving (and so are the UI widgets used in these applications). ProtoActive is a sketch-based prototyping tool that proposes to mimic the visual refinement of paper prototypes. This was done according to Buxton's principles about sketching and low-fidelity prototypes looking quick and dirty which encourages users to provide more feedback [24]. Having a sketch-based prototyping tool was gathered from drawbacks from *UISKEI* [33], *SILK* [32], *DEMAIS* [34], *Balsamiq Mockups* [36], *Axure Rp* [37], *Microsoft Sketch Flow* [38] and *Fore UI* [39].There is however a way to also help designers in further steps of the design, by allowing them to import high-fidelity images of prototypes into ProtoActive and link these pages using ProtoActive's features.

PROTOACTIVE

ProtoActive is based on *Active Story Touch* (AST) a tool developed by Hosseini-Khayat *et al.* that targets the creation of low-fidelity prototypes for touch-based applications [35]. AST had to be modified in order to cover the needs of an ITS application and to allow designers to define their own gestures. Designers in ProtoActive elicit user feedback through sketch-based prototypes that take into consideration the size constraints of the target ITS device the application will be used on. These prototypes allow the evaluation of how users will interact with the application by having a pre-built set of gestures that can be expanded through a gesture recorder tool (IGT) that allows the creation of custom gestures without requiring the designer to write any programming code. These gestures can be used by the user to interact with the prototypes.

Design guidelines based on interviews

A qualitative study was conducted in order to gather requirements for ProtoActive. We conducted semi-structured interviews with UX designers from industry. The semi-structured interviews lasted around 40 minutes each and covered usability issues of sketching on a multi-touch device. The interviews aimed to collect experiences with other prototyping tools from the designers and gather their opinions about desirable features for a sketch-based prototyping tool for touch-based applications. During the interviews, participants could use paper, tablets or tabletop devices that were available at the interview location, to demonstrate behavior and functionalities. Based on the interviews, ProtoActive was designed to be a sketch based prototyping tool. Prototyping tools with pre-built drag and drop *User Interface* (UI) widgets would bias designers to use these widgets thus constraining creativity. From one of the participants: *"I've been working with 3D applications* (for ITS) *for a while and the concept of these components* (pre-built widgets)*, they don't quite apply"*. In ProtoActive, sketches are performed on touch-based screens, meaning that in a similar fashion to paper, the sketches are visualized on the same display where they are made.

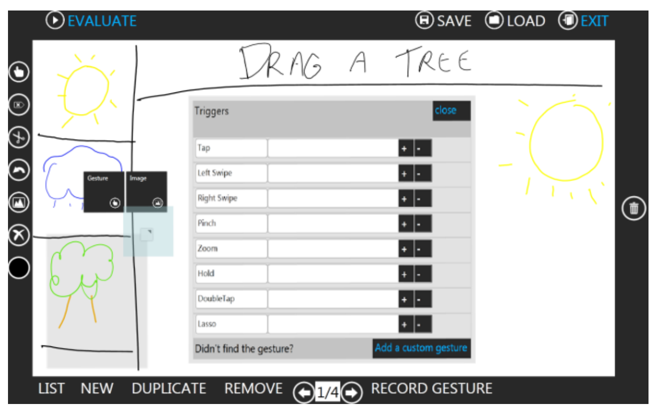

Figure 2 ProtoActive gesture menu

ProtoActive functionality

Prototypes in ProtoActive can be interacted with via gestures and this can be done through the use of *gesture areas*. *Gesture areas* are movable and resizable areas defined on the page of a prototype that can be bound to one or multiple pairs of gesture and page.

ProtoActive comes with a pre-built set of gestures (Figure 2): *Tap* (a single tap with the finger on the surface), *Double tap* (subsequent taps with the finger on the surface), *Pinch* (gesture using two fingers moving towards each other), *Swipe left* and *Swipe right (*single finger moving in the swipe direction), *Lasso* (single finger gesture of an arbitrary shape establishing a closed loop), *Zoom* (gesture using two fingers moving in opposite directions). If a designer wants to use a gesture that is not listed, he can create custom gestures using IGT [41]. IGT (Figure 3) has a canvas where the designer can perform the gesture he wants ProtoActive to learn (involving multi-finger,hand postures and fiduciary markers). The designer can provide as many samples as he wants, as different samples allow the anti-unification algorithm [41] to identify the different nuances that the gesture definition should cover. For each sample provided (and for the anti-unified definition created) ProtoActive shows the primitives (properties) of the gesture provided as a sample.

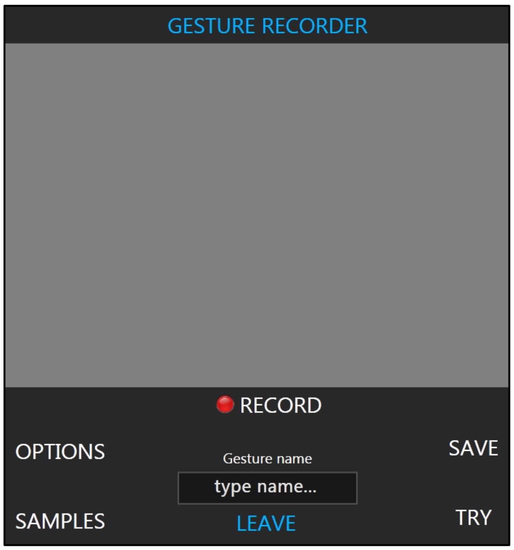

Figure 3 Gesture recorder (IGT) screenshot

For example: length, orientation, number of fingers, basic shapes (circle, line), detection of fiduciary tags, detection of a hand and how these primitives relate to each other. This is the only mechanism that designers have to see how a provided sample was recognized by ProtoActive. Alcantara *et al.* described how IGT integrates with ProtoActive (previously named as ASG) and how it creates gesture definitions without requiring any programming expertise by using samples of the gesture performed by the designer [41]. The main purpose of a prototyping tool is to elicit user feedback about design ideas, so in order to allow designers to evaluate their prototypes, ProtoActive has an *evaluation mode* where the only way to move through pages is through the defined *gesture areas*. If a gesture from the gesture-page binding is recognized on the *gesture area*, the prototype will show the corresponding page of the binding. Figure 4 illustrates how a designer can use a sequence of pages to simulate the behavior of the application. When evaluating the prototype in Figure 4, the first screen (quadrant 1) is a login screen that has a *gesture area* that is activated when a certain tag is placed over the gesture area. Quadrant 2 shows a scan image screen with images A, B and C. Image B has a *gesture area* that is bound to two gestures: place *open right hand* gesture that navigates to quadrant 3, where it shows image B selected and with a menu; and a *"X"* gesture, that navigates to quadrant 4, meaning that image B was deleted. Both *open right hand* detection and *"X"* gesture can be created using IGT, thus making these two gestures available in ProtoActive. It is important to mention that making use of the features of ProtoActive: pages creation, gestures creation, gesture binding to specific areas of the prototype and gesture recognition can be done without writing one single line of code. The machine learning process to create gestures was described in Alcantara *et al.* [41].

EVALUATION

In order to evaluate ProtoActive, the evaluation was conducted in three stages: first, a pilot study was conducted to evaluate ProtoActive in designing ITS applications having developers with experience in developing multi-touch applications for tabletops. The second study incorporates the results of the first pilot study and was conducted with designers experienced in designing tangible applications and focused on getting qualitative feedback about using ProtoActive to design tangible applications for tabletops. Finally, the third stage was an evaluation of ProtoActive *in the wild* [40], where ProtoActive was used by designers for a period of at least two weeks in their projects.

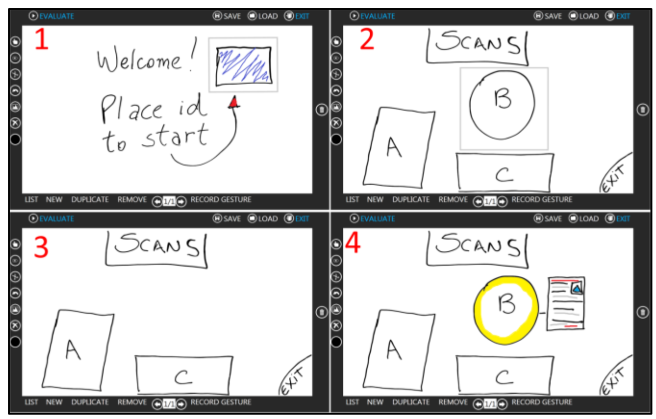

Figure 4 Navigating between the pages in ProtoActive

Pilot study of gesture based prototypes

A pilot user study of ProtoActive was conducted with seven participants. Each one of the participants had a minimum of six months of experience developing ITS applications for academic projects. Participants were presented with a demo of the features of ProtoActive that lasted on average ten minutes. The demo explained how to draw, navigate between pages in ProtoActive and how to create a gesture. In order to avoid biasing participants, the pre-built set of gestures was not offered to the participants. Participants were asked to create a prototype for an ITS medical application to select MRI scans. The scenario given to participants covered three main functionalities in a similar fashion as shown in Figure 4: a log in screen; a selecting a scan image; and bringing up a menu over an image to delete it. The participants were asked to create the prototypes using ProtoActive on a Microsoft Surface and view their designs by clicking "*evaluate prototype*" when done. According to the *Think Aloud Protocol* [42], participants were encouraged to verbalize their impressions and comment throughout their experience with ProtoActive. By the end of the evaluation, participants were asked to complete a survey that asked their impression of using ProtoActive.

Time spent on tasks

The average time to build the prototype for the login page was four minutes and forty six seconds with a *standard deviation* of one minute and thirty seven seconds (SD=1:37); the *select image* took an average of five minutes and twenty seconds (SD=2:59) and finally the last page of the prototype that should show a menu and delete a scan image took an average of five minutes and fifteen seconds (SD=1:52). The longest time was spent when the participant was not satisfied with gesture recognition.

Defining and evaluating gestures

Defining a gesture in IGT had usability issues regarding the information that is shown to participants. While providing a sample gesture to IGT, the only feedback that participants used for checking whether the provided sample was properly analyzed was the canvas that contained the strokes from the gesture. One of the participants that tried to look at the GDL of the gesture, said that it did not mean a lot to him and that "*it seemed fine*". Another participant mentioned that "(GDL) *it doesn't look clear enough to read it*". The participants were asked to create any gesture they thought to be appropriate for the task. Figure 5 and Figure 6 show the gestures created for each task and the occurrence of the gesture for that task. For the last task, the combination of opening a menu and deleting an image produced a different combination of gesture for each participant: *square and tap*; *tap* and *swipe right; z shaped gesture*; *swipe up and swipe left*; *swipe up-left and tap*; *swipe right and tap*.

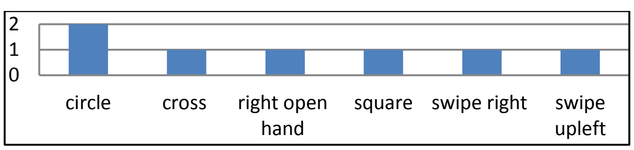

Figure 5 Gestures used for login

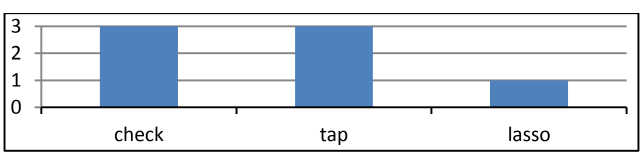

Figure 6 Gestures used for selecting an image

Discussion of pilot study results

The study shows the potential of different interaction approaches that can be used for the same task and emphasizes the need for a tool like ProtoActive that allows designers to explore different interaction approaches but evaluating these interactions in user studies using prototypes. For the first (logging in) and the last task (opening a menu and deleting), the different ways that an interaction can be designed for the same task illustrate how ProtoActive could help designers explore and evaluate different ways for a user to interact with an ITS application. The survey showed that, overall, the participants were satisfied with both IGT and ProtoActive, with IGT eliciting a few remarks when it appeared that a gesture was not recognized: feedback about the samples recognized for a gesture definition and problems with sketching in ProtoActive.

Improving gesture definition

Due to inconsistent hardware performance in touch recognition, it is necessary to provide designers with feedback about how a sample of a gesture was recognized. The problem of showing designers how a sample was recognized is not a trivial aspect, as having detailed information about the recognized sample might require expertise from designers to understand it; and without any information designers cannot identify potential recognition problems. In order to solve this problem, the chosen approach was to show a thumbnail of the print of the sample (the stroke generated while providing the sample) next to the definition of the sample in GDL. A designer isn't required to read the GDL of each sample but if he wants to see the gesture definition, he can look for detailed information of the steps recognized in the provided sample that might affect the gesture recognition.

Improving prototyping on a touch-based device

Another problem noticed during this evaluation was caused by a design decision about the sketching features of ProtoActive: to take advantages of a multi-touch device, selecting strokes, erasing and defining *gesture areas* needed a combination of two hands to happen; while one finger stays pressing the correspondent button, the other would perform the action on ProtoActive drawing canvas (*e.g.*: one hand holds the selection button while the other performs a lasso on canvas to select strokes). This feature was not well accepted by participants, due to hardware limitation, in some occasions an event would be miss-triggered, detecting that a finger was moved up from the device; depending on the distance between the button and the place on the canvas that the action was performed, it felt uncomfortable for participants; and in some other occasions, participants would move up the finger holding the button by mistake. The solution was to change this functionality to a regular button on the screen that doesn't need to be held.

Pilot study of prototyping TUI applications

This pilot study investigated the value of prototyping in the design of tangible applications. The study was conducted with five UX designers that had experience in the design of tangible applications; four designers from academia and one from industry. In order to validate prototypes in the tangible application context, designers were asked to create a prototype in ProtoActive and include the use of some clay, printed tags and plastic toys with tags attached to them (which were provided for the study). The aim of this study was to have participants think aloud about prototyping for tangible applications, to collect information about the value of prototyping for tangible applications and how a tool could better improve this process. Participants were given a scenario where they needed to design functionalities of a *Geographical Information System* (GIS) application. Participants were asked to prototype a login screen and a map that would change layers when specific tags were placed on the surface tabletop. According to the *Think Aloud Protocol* [42], participants were encouraged to verbalize their impressions and comment while creating the prototypes in ProtoActive. After creating the prototypes, a semi-structured interview was conducted. The interview covered the importance of prototyping for tangible applications, regarding the application and the physical tangibles; the participant's impressions about ProtoActive and how they think ProtoActive could have aided them in their previous tangible-based projects. The participants had never previously used prototypes for the design of the tangible applications. Participants' design ideas had always been communicated through paper sketches and tangibles were used with a trial and error approach.

Time spent on creating prototypes

The participants were impressed with the amount of design ideas that could be covered in a prototype that took less than thirty minutes to be created with ProtoActive. When asked about how useful ProtoActive would be to quickly evaluate design ideas for tangible applications, one of the participants commented: *"coding the interface and the interactions would take forever (...) but if I would use sketches on a paper, I am not sure that I could represent it* (tangibles interactions) *just as nicely"*. Using ProtoActive, as mentioned by the participants, consumed less time than some bad design decisions had cost them in previous projects and could even help to discuss design ideas between teammates: "(to discuss ideas between teammates) *it is so much easier if you can see what you're talking about"*.

Designing and evaluating tangibles

Figure 7 shows tangibles that were provided to participants and could be used in the evaluation; Figure 7 compares a tree created by a participant with a fiduciary tag and a plastic toy that could be used to activate the vegetation layer on the study. The participants' comments on these two options allowed the researcher to understand how crucial to this stage of design the shape of the tangible is. This was mentioned by one of the participants: *"sometimes the concept is still too abstract that the shape of the tangible doesn't matter (...) but there are other cases when it might be important to differentiate, some shapes automatically represent what you want to show, for example, this is a tree and represents vegetation"*. The interviews showed that participants found that creating clay prototypes of the tangibles is a valuable asset, especially for tangibles that imply movement and require ergonomics studies. For situations where a tangible does not need any special shape, the clay did not seem necessary, and the participants chose to use tags simply attached to colored plastic toys.

Participants also commented that a valuable asset of this approach is to also bring clay and printed tags for the evaluation of the prototypes with users, allowing them to make suggestions and even have them create their own clay prototypes during the prototype evaluation. As mentioned by one of the participants: "*you need to prototype it* (the tangible) *as well as it might affect the interaction*".

Evaluations in the wild

Finally, an evaluation of ProtoActive's efficacy for designing applications *in the wild* [40] was conducted by asking two UX designers in industry and academia to use ProtoActive in their design process. The aim of this evaluation was to have designers use ProtoActive in their own environment and to assist with the design of applications they care about. The evaluation was structured in two phases. First, we provided the tool installation and a brief explanation of the tool, in video format, explaining the features of the tool. When the participant had spent at least two weeks becoming familiar with ProtoActive, the author contacted the participants individually, sending a survey. The responses on the survey was used as a guide for a semi-structured interview aiming to collect data about the gestures created using ProtoActive, the application being designed by the participant and the sketches created.

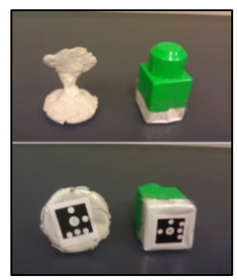

Figure 7 Clay custom tangibles and the plastic toy tangible

Using ProtoActive to design for a vertical multi-touch device to be used in oil platforms

A UX designer used ProtoActive to evaluate design ideas of a gesture-based commercial application to be used in a proprietary dual-capacitive touch display that supports two simultaneous touch points. The display was created to resist extreme temperature conditions. The participant is a UX designer who has eight years of industry experience and no experience in programming gestures for touch-devices. The participant uses low-fidelity prototypes regularly in his job, has used different prototyping tools including pen and paper and considers prototyping a critical part in the design of ITS applications. The designed application is a main system navigation to be used in oil platforms that will likely be used by users wearing protective gloves. In this scenario, besides evaluating interface and gestures, ProtoActive was used to study how designers in an environment with extreme temperature conditions interact with a touch-based device: using gloves or stylus pens. Also, since the designer was working directly with the proprietary custom device during design of the applications in ProtoActive, he was able to test the device capabilities and identify a problem when working with two simultaneous touch points. The overall comment from the UX designer was: "*Overall it's a very promising tool. We had no other tools at all for looking at gestures, so it fills a necessary void. We are unfortunately in an early development stage of our device and with ProtoActive discovered some issues with our touch screen drivers with dual touch and gestures*".

Regarding the drawbacks and problems found using ProtoActive, the UX designer found that it is not clear how many samples would be enough for a good gesture definition. He suggested that for the anti-unified gesture definition, an image was shown illustrating a heat map of an overlap between all the gestures, where overlapping strokes would have higher temperature visualization.

validate as step 1
Touch state: TouchUp
Touch shape: Circle
Touch direction: Right
Touch path length: x
validate as step 2
Touch state: TouchUp
Touch shape: Circle
Touch direction: Right
Touch path length: 1.5x ..2x
validate
Touch limit: 2
Relative position between 1 and 2: Left

Figure 8 Circle gesture defined by participant

Regarding reading GDL the UX designer said that he grew more comfortable and could understand better the language after the time he spent using it. The participant's only suggestion to how this could be improved would be to provide a list with all the possible primitives that can be identified in GDL, but overall, the participant was satisfied with reading GDL and said that it was a good way to determine if a sample was properly recognized. Regarding usability issues with ProtoActive, the participant's only suggestion was to have a way to fix the position of some *gesture areas*, avoiding unintended drags on the page.

Using ProtoActive to design a tabletop game

The participant in this evaluation is a PhD candidate who used ProtoActive to prototype a new version of the high automation interface for tabletop of the pandemic game described by Wallace *et al.* [43]. The participanthad no previous experience in designing ITS applications or programmatically creating gestures, but with experience in using pen and paper to prototype interfaces. In order to design the prototypes, the participant used a tablet that supports up to two touch points.

The final application will be used in a tabletop device, but according to the participant, a tablet was used for prototyping due to:

- availability of the device, as the tabletop that could be used to prototype is shared among other teammates for different projects;
- portability, as sometimes the design had to be shown or evaluated in different locations, having a tabletop would impair the evaluation process.

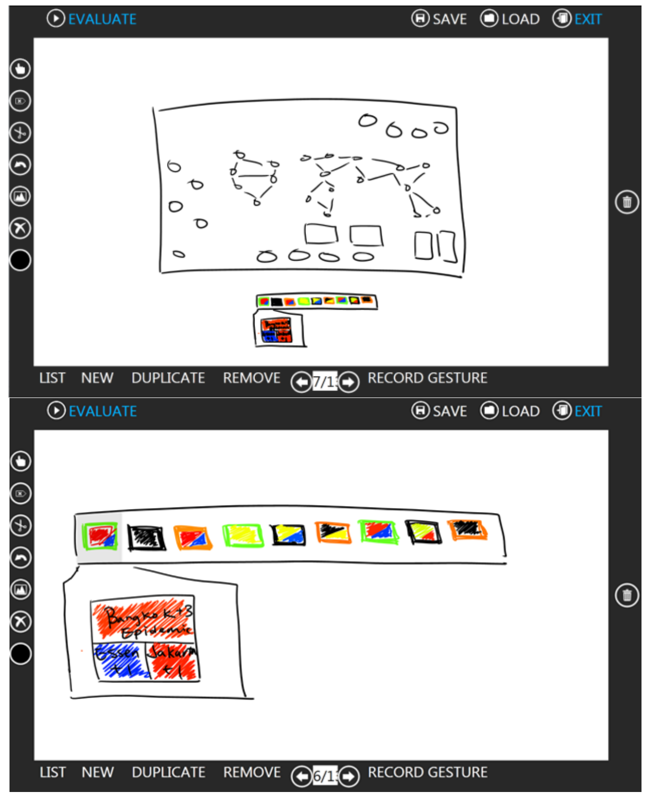

Figure 9 Using ProtoActive to provide more detail about items

According to the participant, ProtoActive was used in the following scenarios:

- using ProtoActive as a tool to brainstorm and sketch different ideas. Later, if needed, interactivity can be added to the sketches on ProtoActive and they can be evaluated;
- creating a prototype of small tasks and guide users to play with the prototype to elicit discussion about the interface and the interactions in it;
- using the prototypes to transmit ideas about design and interaction options. In this scenario the designer was the one interacting with the prototypes and was mostly used during meetings to communicate the design ideas to supervisors and teammates.

In total, the participant estimated to have used ProtoActive for fifteen hours spread along three weeks, generating twenty different prototypes, and having four users that evaluated the prototypes. Regarding the gesture definition feature, the participant commented: *"I think the defining custom gesture functionality was pretty good. It is unclear what order to carry out actions for first time users. However, once learnt,* *I think it is pretty good"*. The participant mentioned that for most of the interactions used in the prototypes the gestures pre-built in the tool sufficed and only three custom gestures were created: *two fingers hold* and *two fingers swipe*.

Figure 9 illustrates how the participant used ProtoActive to better illustrate specific points of a prototype. The top of Figure 9 shows how a menu will appear contextualized within the game screen; the bottom of Figure 9 shows the menu in more detail, showing how a designer can obtain feedback about different depth of functionalities. Also, as can be seen in Figure 9, a prototyping tool based in pre-built UI widgets would change the level of abstraction as most of the interface items in the prototypes are undefined shapes.

DISCUSSION AND LIMITATIONS

Allowing designers to create custom gestures allows the evaluation of different interaction ideas contained in the costs and time constraints of low-fidelity prototyping. This was shown by the evaluations in the pilot studies that contained gestures that do not exist in the prototyping tools investigated in related work. Providing designers with ways to evaluate these gestures in the final application context (through using the custom created gestures in interactive prototypes) allows these innovative interactions to be developed following a user-centered approach as recommended by Norman and Nielsen [2]. The different evaluations showed that ProtoActive fills the need for creating prototypes for ITS applications, but it arguably has limited support for gathering data during evaluation. Designers often rely on their own equipment to record the video and audio of evaluation sessions of the prototypes. ProtoActive support for evaluation relies on the same as paper prototyping.

ProtoActive uses Gesture Toolkit [7] for gesture recognition and definition. This means that some of the limitations in Gesture Toolkit are inherited. While the GDL supports multi-step gestures, it is currently limited to gestures with sequential steps and that need to fit in the toolkit gesture primitives. The feature to allow the gesture recognizer to break the gesture into parts facilitates the sequential process but it requires some experience from the designer to decide if dividing a gesture into steps or not will generate the best gesture definition for its needs.

CONCLUSION AND FUTURE WORK

This research offers two main contributions. The first is ProtoActive, a sketch-based prototyping tool for ITS applications. ProtoActive allows designers to evaluate not only the output of sketch-based prototypes (namely what happens when a user wants to accomplish a task) but also the input on the prototypes and how a user wants to interact and accomplish a task. In order to allow the evaluation of this input, the prototypes in ProtoActive can be interacted with via a pre-built set of gestures or through customized gestures. ProtoActive supports designers following user-centered design of ITS applications. The second contribution in this paper is an evaluation of ProtoActive consisting of two pilot studies and two evaluations in the wild. The first pilot study gathered the different gestures that participants created to perform similar tasks. The variety of gestures created for the same task suggests that designers benefit from a tool like ProtoActive to evaluate different and innovative interactions. The second pilot study evaluated ProtoActive features for prototyping TUI applications, using fiduciary markers. One of the participants stated that by using such a tool, hours of development could be saved by evaluating the tag-based gesture in a prototype that took thirty minutes to be created. This shows that potential problems and design issues could be addressed before the implementation phase. Feedback from the second pilot study shows that by being so easy to use, such a tool could also be used to explain a design idea and act as a communication artifact between team members (which was in fact used by the evaluation in the wild for the tabletop game). Our participants recognized the value of using ProtoActive to experiment and evaluate design ideas in an early stage of application development. Future work should address the usability issues found during the evaluations of ProtoActive. Our method of defining and recognizing gestures could be improved to be adaptable to other gesture recognizers. Also, a mechanism for resolving conflicting gestures during gesture recognition should be created. In the current version of ProtoActive a *gesture area* can support multiple gestures, but does not detect potential conflicts between gesture definitions. The future conflict resolution mechanism should also then have a way of warning designers about conflicting gestures.

REFERENCES

1. Norman, D.,A. (2007). The Design of Future Things. Ed. Basic Books.
2. Norman, D., Nielsen, J. (2010). Gestural interfaces: a step backward in usability. Interactions, vol 17, issue 5.
3. Hesselmann, T., & Boll, S. (2011). SCIVA: designing applications for surface computers.EICS 2011, 191-196.
4. Moggridge, B. (2007). Designing Interactions. MIT, Ch 10 – People and Prototypes. Press, Cambridge, MA.
5. Hinrichs, U., Carpendale, S. (2011). Gestures in the Wild : Studying Multi-Touch Gesture Sequences on Interactive Tabletop Exhibits. CHI'11, 3023-3032.
6. Wobbrock, J. O., Morris, M. R., & Wilson, A. D. (2009). User-defined gestures for surface computing. *CHI '09*. Pages 1083-1092.
7. Khandkar, S. H., & Maurer, F. (2010). A Domain Specific Language to Define Gestures for Multi-Touch Applications, DSM '10, Article 2 , 6 pages.
8. Lao, S., Heng, X., Zhang, G., Ling, Y., Wang, P. (2009). A gestural interaction design model for multi-touch displays. BCS-HCI '09, 440-446.
9. Allan Christian Long, Jr., James A. Landay, and Lawrence A. Rowe. (1999). Implications for a gesture design tool. CHI '99. 40-47.

10. Lyons,K., Brashear,H., Westeyn,T., Kim,J.S., Starner, T. (2007). GART: the gesture and activity recognition toolkit. HCI'07. 718-727.
11. Anthony, L.,Wobbrock. J.O. (2012). $N-protractor: a fast and accurate multistroke recognizer. GI'12.117-120.
12. Kin, K., Hartmann B., DeRose T., Agrawala M.. Proton: Multitouch Gestures as Regular Expressions. CHI’12, ACM 978-1-4503-1015-4/12/05.
13. Plimmer, B., Blagojevic,R., Hsiao-Heng Chang,S., Schmieder,P., Zhen, J.S. (2012). RATA: codeless generation of gesture recognizers. BCS-HCI '12, 137-146.
14. Wiethoff, A., Schneider, H., Rohs, M., & Butz, A. Greenberg, S. (2012). Sketch-a-TUI: low cost prototyping of tangible interactions using cardboard and conductive ink. Embodied Interaction, 1, 309-312.
15. Marco,J., Cerezo,E., Baldassarri,S. (2012). ToyVision: a toolkit for prototyping tabletop tangible games. EICS '12. 71-80.
16. Shaer,O., Jacob, R.J.K. (2009). A specification paradigm for the design and implementation of tangible user interfaces. CHI’09. 16, 4, Article 20, 39 pages.
17. Rudd, J., Stern, K., Isensee, S. (1996). Low vs. high-fidelity prototyping debate, interactions, v.3 , p.76-85.
18. Sefelin, R., Tscheligi, M., Giller, V. (2003). Paper prototyping - what is it good for?: a comparison of paper and computer-based low-fidelity prototyping, CHI '03.
19. Virzi, R.A., Sokolov, J.L., Karis, D. (1996). Usability problem identification using both low- and high-fidelity prototypes, CHI’96, p.236-243.
20. McCurdy, M., Connors, C., Pyrzak, G., Kanefsky, B., Vera, A. (2006). Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. CHI '06. 1233-1242.
21. Youn-Kyung Lim, Stolterman, E., Tenenberg, J.. (2008). The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. CHI’08. 15, 2, Article 7 , 27 pages.
22. Derboven, J., Roeck, D. D., & Verstraete, M. (2010). Low-Fidelity Prototyping for Multi-Touch Surfaces. . Presented in the workshop Engineering Patterns for Multi-Touch Interfaces held in EICS’10.
23. Klaus Krippendorff. (2006). The Semantic Turn: A New Foundation for Design. Taylor & Francis, Boca Raton, FL.
24. Bill Buxton. (2007). Sketching User Experiences: Getting the Design Right and the Right Design.Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
25. Constantine, L. L. (2004). Beyond user-centered design and user experience: Designing for user performance. Cutter IT Journal, 17, 2.
26. Robertson, S., Robertson, J. (2006). Mastering the Requirements Process (2nd Edition). Chapter 12. Addison-Wesley Professional.
27. Rudd, J., Stern, K., and Isensee, S. (1996) Low vs. high fidelity prototyping debate. Interactions, 3, 1, 76-85.
28. Morris, M., R., Wobbrock,J., O., Wilson., A.,D. (2010). Understanding users' preferences for surface gestures. GI '10. 261-268.
29. Gerken,J., Jetter,H.C., Schmidt,T., Reiterer, H. (2010) Can "touch" get annoying?. ITS '10. 257-258.
30. Sinha, A.K., Landay,J.A. (2003). Capturing user tests in a multimodal, multidevice informal prototyping tool. ICMI '03. 117-124.
31. J. David Smith and T. C. Nicholas Graham. (2010). Raptor: sketching games with a tabletop computer. Futureplay '10. 191-198.
32. Obrenovic, Z., Martens, JB.(2011). Sketching interactive systems with sketchify. ACM Trans. CHI’11. 18, 1, Article 4, 38 pages.
33. Segura V.C.V.B., Barbosa,S.D.J., Simões, F.P. (2012). UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior. AVI '12. 18-25.
34. Bailey, B.P., Konstan, J.A., Carlis, J.V. (2001). DEMAIS: designing multimedia applications with interactive storyboards. MULTIMEDIA '01. 241-250.
35. Hosseini-Khayat, A., Seyed, T., Burns, C., Maurer, F. (2011). Low-Fidelity Prototyping of Gesture-based Applications. EICS’11. 289-294.
36. Balsamiq Mockups – Available at www.balsamiq.com. Accessed July 2012
37. Axure RP: Interactive wireframe software and mockup tool. Available at http://www.axure.com/. Accessed October 2012.
38. Microsoft Sketchflow. Available at http://www.microsoft.com/expression/products/sketchflow_overview.aspx. Accessed March 2012.
39. ForeUI: Easy to use UI prototyping tool. Available at http://www.foreui.com/. Accessed October 2012.
40. Johnson,R., Rogers, Y., van der Linden, J., Bianchi-Berthouze, N. (2012). Being in the thick of in-the-wild studies: the challenges and insights of researcher participation. In Proceedings of CHI '12. 1135-1144.
41. Alcantara, T., Denzinger, J. , Ferreira, J. , Maurer, F. (2012). Learning gestures for interacting with low-fidelity prototypes. RAISE’12,.32-36.
42. Lethbridge, T. C., & Sim, S. E. (2005). Studying software engineers: Data collection techniques for software field studies. Empirical Software Engineering, 10(3), 311–341.
43. Wallace, J.R., Pape, J., Yu-Ling Betty Chang, McClelland,P.J., Graham,T.C.N., Scott, S.D. and Hancock, M. (2012). Exploring automation in digital tabletop board game. CSCW '12.231-234.