

A collaborative face-to-face design support system based on sketching and gesturing

Gustavo Zurita ^a, Nelson Baloian ^{b,*}, Felipe Baytelman ^a

^a Management Control and Information Systems Department, Business School, Universidad de Chile, Santiago, Chile

^b Computer Science Department, Engineering School, Universidad de Chile, Santiago, Chile

Received 31 August 2007; accepted 13 September 2007

Abstract

Mobile and pervasive computing has seen a rapid development in the last years. Portable, handheld computing devices are getting more popular as their capabilities increase. Therefore, people having the need to work on-the-field have now the possibility to be supported by computer technology, for example, garden or building designers. Ad-hoc networking capabilities of handheld devices enable the development of supporting tools for collaborative work anywhere. This paper presents MCSketcher, a system that enables face-to-face collaborative design based on sketches using handheld devices equipped for spontaneous wireless peer-to-peer networking. It is especially targeted for supporting preliminary, in-the-field work, allowing designers to exchange ideas through sketches on empty sheets or over a recently taken photograph of the object being worked on, in a brainstorming-like working style. Pen-based designed human–computer interaction is the key to supporting collaborative work. The gestures-based command input results in a lightweight yet efficient and easy to use interface. Graph structured hierarchical documents help to overcome the problems of a small screen size.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Mobile collaborative design; Handhelds; Gestures; Sketches; Pen-based; Concept maps

1. Introduction

The growing acceptance of handhelds enables users to take advantage of numerous facilities that mobile information systems can provide in environments that computer technology could not reach otherwise [1]. In addition to supporting individual tasks as note-taking, scheduling, address storage, etc., handhelds can build ad-hoc wireless peer-to-peer networks and support collaborative face-to-face applications anywhere [2,3]. Although peer-to-peer approaches to collaborative computing are still not common, according to [4], they are becoming increasingly interesting to developers and users, since they allow for new forms of cooperation that do not naturally fit in the

client–server model, such as ad-hoc and opportunistic collaboration. From a technological point of view, peer-to-peer communication is well suited for sharing objects, because it allows different participants (peers) to maintain their own knowledge structures while exchanging information.

Increasingly centralized design and manufacturing systems are being found insufficient to respond to highly dynamic situations. With the advent of the Internet, product design and manufacturing is becoming increasingly distributed and new environments are required to develop and produce high quality products faster and cheaper [5]. Potential beneficiaries of systems implementing these features include users whose activities involve on-site collaborative design sketching. For example, a group of architects may need to jointly work on a construction site using sketches to exchange ideas about re-designing facilities [6]. Or a group of engineers conducting an on-site inspection might require high mobility and efficient communication to jointly work on possible deficiencies and

* Corresponding author.

E-mail addresses: gnzurita@fen.uchile.cl (G. Zurita), nbaloian@dcc.uchile.cl (N. Baloian), fbaytelmanp@fen.uchile.cl (F. Baytelman).

improvements of the facilities exchanging graphic sketches while moving around the premises [7]. The construction industry presents particular opportunities for using mobile information systems to improve collaborative design practices on building sites.

A handheld's most natural data-entry mode is using the stylus (a.k.a. a pen-based or freehand-input-based system). This enables users to easily write down their ideas and/or draw design sketches imitating the use of pen and paper [8–10]. However, most currently available handheld applications implement interfaces following the approach used for applications running on desktop computers using widgets (buttons, menus, windows) instead of freehand-input-based paradigms (via touch screens) and/or sketching [6,11,12].

According to [8], graphics-editing applications running on handhelds do not require a mouse-and-palette-based interface, nor do they need to rely on such elements as colors, fonts or lines for creating sketches. They should rather implement interfaces using natural freehand-input stroke to interact with them. Applications implementing this approach have already been developed for scenarios such as sketching informal presentations [10] and military courses of action diagrams [13], and for being used as a non-technological method of idea generation in meetings. But no system using only interconnected handhelds in a wireless ad-hoc network with a pen-based system to create a sketch-based collaborative design-editing mechanism, has been proposed yet, as it was done for PC-based systems [14], but they do not share the advantages of mobile platforms.

The ability to draw design sketches on the fly, at any physical location and even while in movement, allows users to discover opportunities for substantial improvements at an early stage of the design process and thus enhance efficiency of productivity. Through sketching the user can employ visual symbols, describe spatial relations [6], and/or exchange opinions [15] using a fast and efficient way for sharing and discussing complex ideas [9,15]. All this, however, requires sketch diagramming support, allowing users to engage in face-to-face communication and explanation of their rough designs simultaneously [9,15,16]. The ability to handle graphic representations while engaging in a face-to-face communication is a natural mode of expression [9] crucial to knowledge creation and capture among a group of people [15].

In this work, we present MCSketcher: a Mobile collaborative sketching system using wirelessly interconnected handhelds in an ad-hoc network. It uses the pen-based paradigm, enabling users to draw sketches collaboratively and share their “graphic opinions” while maintaining at the same time a face-to-face communication to explain their designs. Two further fundamental aspects add value to the proposed application: the use of conceptual maps for organizing and structuring sketches [17], and the use of gestures as a simple way of implementing data-entry functions required by the user [12].

2. Related work

According to [8,12], sketching and gesturing with pen-based systems are natural modes for design–task-oriented interaction. In [9] it is noted that a sketch is a quick way of making designs that (a) facilitate the creator's idea generation process, (b) stimulate communication of ideas with others, and (c) stimulate the use of early ideas, thanks to the accessibility and interpretation they provide. It has been shown that the participation of various persons in the elaboration of a sketch using computer support improves the creativity of the group [16]. Various computer systems supporting sketch-based interaction have been developed in recent years. Desktop systems providing collaborative support include:

- Networked virtual environments (net-VEs). These are distributed graphical applications that allow multiple users to interact in real time, providing a shared workspace, as well as communication mechanisms [17].
- SKETCH [18] is a collaborative system based on a client–server model for supporting conceptual designs, that provides distributed users a cooperative workspace for sketching, as well as interactive design exploration and edition. However, it does not provide awareness of other users' presence. SKETCH includes an interface for creating and editing 3D sketches of scenes based on the use of simplified (2D) drawing commands interpreted as operations to be applied to objects in a 3D world. All objects are thus 3D and rendered in orthographic views.
- NetSketch [19], which is an application based on the SKETCH interface that supports distributed conceptual design, in which scene models are constrained to the relatively simple shapes that can be created and rendered. NetSketch uses a peer-to-peer network topology but cannot always guarantee model consistency among all users.
- A collaborative system for conceptual design is described in [14] allowing users located in geographically distant areas to cooperate by sketching, exploring and modifying their ideas interactively, with immediate visual feedback. The system can be used for urban and landscape design, rapid prototyping of virtual environments, animation, education and recreational activities.

Handheld-based note-taking and sketching systems have also been proposed:

- Citrin et al. [11], describe a software architecture that supports pen-based mobile applications through a client–proxy server organization, allowing many graphical applications designed with a mouse/palette-keyboard interface to be accessed through pen-based mobile devices offering shape, gesture, and handwriting recognition.

- Davis et al. have designed an informal system that supports team capture of meeting notes [20]. The system, known as NotePals, uses a detail-in-context technique during note capture. The user selects a location for new information on the screen, and the system highlights the insertion location. The user then enters notes at the bottom of the display.
- PebblesDraw focuses on the use of PDAs for computer-supported cooperative work [2]. The touch screen on the PDA provides access to a remote space. In group meetings, this space is viewed as a “common ground” where PDA users can simultaneously post information. To visualize all the information in the remote space, users can switch focus between their PDAs and an external monitor displaying the shared space.

3. System design principles

As we already mentioned in the first chapter, we want to provide users with a design supporting tool which can be used on-site in a collaborative way. This is because design must often be produced at the very location they are requested and sometimes while people are moving around. Such situations are common for engineers in construction [7], on-site construction inspectors [21], garden designers and architects. They require high mobility and support mechanisms for establishing collaborative interaction with other colleagues, employees, or clients [7]. Sometimes they may even create the design based on a photo captured and pasted into the handheld screen background (see Fig. 1). Handheld computer devices are an appropriate technology for providing high mobility and portability, creating ad-hoc networks through peer-to-peer connections. In fact, handhelds are considered to be a good platform for reading brief, concrete content because their interface is simple and neutral to different information formats, thus allowing faster information reading. They are also considered to be well suited for supporting collaborative work groups (for many examples see [22]). However, their reduced screen size and the necessary use of virtual keyboards or widgets for entering and handling information, introduce new complexities to the user-handheld interaction, [12]. In order to overcome these problems we propose the following design principles:

- Interaction based exclusively on gestures, minimizing the number of widgets and the need of a virtual keyboard and maximizing the space available for entering content. The content consists of exclusively free handwriting inputs. Although free handwritten text may take more space than typed text, it allows a flexible combination of sketching and writing. Gestures can combine in one action a command and its arguments, thus avoiding errors. In [9] it is noted that a sketch is a quick way of producing designs that (a) facilitate the creator’s idea generation process, (b) facilitates the sharing of ideas

among group members, (c) stimulate the use of early ideas, and (d) gives the opportunity to see and get inspired by other group members’ ideas. A pen-based input interface also enables the use of gestures for interacting with the system, and sketches for facilitating data-entry [8–10,12]. As observed in [12], collaborative design based on gestures, sketches and an interface allowing pen-based interaction enhances the design process in a natural and harmonious way, enabling the sharing and exchange of design information in order to improve efficiency (see Section 4).

- Many systems use the metaphor of pages and/or scrolling bars for the generated documents in order to offer more available working space to the user. This is indeed a simple and very intuitive way of organizing a document. However, when the working area is extremely small, which is the case of handhelds; it seems to be better to organize the content in a structure which is intuitive and provides additional information in the structure itself without having to enter more data. A structure like a concept map is certainly more suitable to organize the ideas generated during the meeting than the “list of pages” (which in fact is a simple form of a concept map) since it is an intuitive yet flexible structure which holds intrinsic information relative to the content in the structure of the map. This type of shared visual space has been already successfully applied in discussion groups [17], design groups and collaborative activities.
- Sketching is a powerful means for supporting interpersonal communication [15]. Face-to-face communication may involve the use of diagrams and drawings in a way that enables users to share views while they talk to each other. This process avoids ambiguities and helps to communicate new and complex ideas swiftly [8,9]. It has been shown that people cannot produce finished designs in real time or draw objects and relations between them that result in complex designs without interrupting the flow of verbal information [15]. On the other hand, with sketching, only a minimum of verbal exchange among those working on a design is required to achieve a common and fluid communication channel [8,9,15] (see Sections 5.1 and 5.2).
- It has been found that in many pen-based interfaces supporting sketching and gesturing users have to switch explicitly among these two input modes. Yang [4] has found that a slow or ineffectual sketch/gesture mode switching technique, may become a mayor bottleneck in the usability of the system, originating a significant source of errors, confusion and complexity. To avoid this, a single mode interface is implemented in which any pen-based input is first analyzed, in order to find out if it matches a known gesture for the system which should trigger an action. If not, it is considered a sketch input. Of course, this means that it is not possible to enter sketches matching a gesture. Therefore fewer gestures should be implemented not to confuse them with frequently used gestures (see Section 5.2). According

to [23], a survey intended to shed light on the problems and benefits users experience with gestures over pen-based interfaces, found that the most frequent actions were selecting, moving and deleting. The users consider these actions to be an efficient form of interaction, easy to learn, use and remember, thus giving an added value to the interface.

4. System description

As discussed above, MCSketcher is intended to support the collaborative on-the-field design with handhelds. We think a design session should proceed as swift as possible. Therefore, there is no login procedure to start one. Participants just start the application on their handhelds; the applications discover automatically each other. The necessary logical connections in order to build a collaborative session are made automatically by the software at the beginning. We consider the authentication process not necessary because it is not probable that others not taking part of the design would be at a distance the ad-hoc network would be able to reach.

4.1. The document's structure and navigation

A working session with MCSketcher starts with a blank page for all users, which is synchronized for everyone. Any participant can draw a sketch and this will be transmitted to all the rest. Any participant having a built-in camera can take a photograph and use it as a background image for the page. This image will also be distributed to all the rest. The use of images previously stored in the handheld is also possible.

Over this first-level page, diverse “Design spots” can be defined. These are areas that are linked to another page which can be used to describe a particular object or region of the sketch in more detail (see Fig. 1). This can be done

recursively creating in this way document structured like a tree, where pages are the nodes and design spots are the links from a father page to its sons.

Users can create “design spots” and navigate into or out of them, building an implicit tree-shaped conceptual map. Because the depth and complexity of the document tree can hinder users’ location awareness inside the document, a view of the document tree (see Fig. 2) can be displayed by pressing the document tree icon (shown in Fig. 1).

Because users start in the synchronized navigation mode, they also move together through the document. This means, when any user changes the page following the link of a design spot, all other users will follow and have the page changed to the target one. However, a user may want to navigate independently through the document, for example: for writing a sketch in a new page without disturbing the ongoing discussion. Users can switch to an independent navigation mode by pressing the “Session” icon (see Fig. 1). With this feature designers can work in

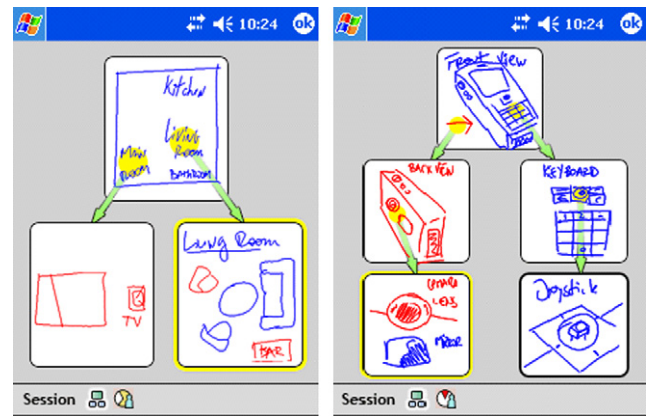


Fig. 2. Two examples of document tree view. Yellow nodes show current user locations (“living room” on the left, and “camera lens” on the right). Thick borders indicate unvisited modified nodes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Darkened margin denotes user is in an inner sub-node

Different colors show singular designer's contributions

Highlighted Session menu shows the work needs to be saved

Design spots show there are other sketching pages attached to this one, which can be explored by following the link shown as a yellow circle. In this case, there are 2 of them

“Document tree icon”

The “group icon” shows that 2/3 of the users are in this node

Fig. 1. MCSketcher system screenshot.

parallel, defining new nodes and drawing sketches in different areas, thus enhancing their collaborative work design. Users can join the synchronized navigation mode by pressing the “Session” icon again. If all users decide to work in a non-synchronized way, the last one to abandon the synchronized mode retains the “master session”. When others want to join the synchronized node again, they will join master session.

Unvisited modified nodes, or nodes including modified sub-nodes, display their design spot (the colored mark behind the sketches) in a stronger tone to provide awareness. In the document tree view these unvisited modified nodes are also highlighted (Fig. 2). The two highlighting methods notify users of changes, inviting them to visit the modified spots.

4.2. Communication architecture

To meet all the requirements for the system it is necessary to develop a full peer-to-peer architecture. This means that all users will have the same program running on their handhelds and there should be no central service (see Fig. 3). The program must be able to recognize the presence of other participants and establish a secure communication with them in order to transfer data for synchronizing the applications. In this system the discovery of other participants is done via multicasting, after which a point-to-point communication link is established to exchange data. Each running program is sending constantly a multicast message revealing its presence and the parameters for establishing communication. This message will be consumed by the peers and used to maintain a list of active participants which will be used when there is the need to distribute synchronizing information. All this functionality is encapsulated in a module implementing a single send-to-all method which can be called when synchronizing information should be distributed among all participants, easing in this way the programming of the rest of the system. This module is part of a framework previously developed by the same authors for supporting the programming of peer-to-peer mobile

applications, and has also been used for implementing other handheld-based systems. The structure of the synchronizing information is an object which is converted to a XML representation first, and then sent as text information to the other applications. This was done in order to enable the communication with programs and systems developed on other platforms like.NET. The coding and decoding of the object to its XML representation is done by a standard procedure provided by the NanoXML package, which is a small, open source XML parser for Java.

Another important module of this framework is the one providing gesture recognition functionality. In this module the recognition of every gesture is implemented as a separate object class. Therefore it is very easy to extend the available gesture set by just writing new classes. Finally there is the main module implementing the system itself and its graphic interface which uses all other modules.

5. Interface design

As we already said, the role of traditional graphical user-interface widgets used for PC-based application for triggering actions needs to be reevaluated in the context of the small screen of the handheld device [8]. In particular, their number and size should be reduced and context-limited [24] in order to achieve a balance between the two techniques (sketches and gestures). Thus, some actions may be triggered through gestures directly over the drawing area, while others may be activated by more traditional controls such as interface widgets. In this system, actions closely related to the sketches are triggered by gestures while those operations independent from the drawing or related to state control are triggered by buttons, with icons reflecting their current state and mode the system is in.

5.1. Metaphors for user actions in a collaborative sketching application

Actions available in the system may be classified in three categories: drawing, session management and navigation. Drawing actions comprise background selection, basic sketching (drawing strokes), selecting, moving, cutting and pasting. Session management actions include creating a new session, synchronizing navigation, plus opening, saving and exporting a session as a PDF document.

In collaborative workspaces, navigation actions are relatively complex. Navigation actions include creating pages (nodes) and moving among them, zooming in and out on design spots and displaying document trees. In order to improve the awareness on how many users are following the synchronized navigation, the system displays an indicator consisting of a pie chart showing the percentage of the users in the synchronized mode.

The system uses its own ad-hoc gesture recognition system based mainly on the strokes entered by pen interaction and their drawing speed to which gesture-recognition rules are applied. Such a rule matching implementation is more

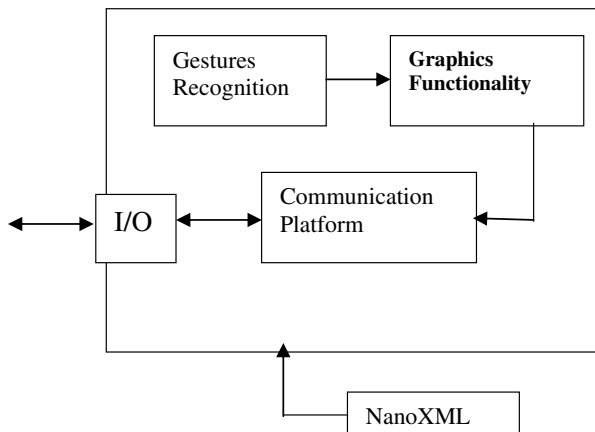


Fig. 3. The MCSketcher system architecture.

lightweight and easier to introduce than the traditional neuronal network approach because there is no need to train the system.

Most drawing actions are executed using the following gestures over the drawing area:

- Background selection is performed by surrounding the screen area with a rectangular gesture, which triggers a standard file-open dialog. The user can then choose any image, including a recently taken photo on camera-enabled handhelds.
- Selecting is done by any of the following three methods. The first one consists on clicking with the pen on a given trace or line (see Fig. 4a), which will select all other traces touching it (Fig. 4b). The second method is to double-surround them with a continuous closed shape (Fig. 4c and d), to select a group of traces which are not necessarily connected. The last method, used as an alternative to the double-surrounding is to draw a dense dot and then, without releasing the stylus, draw a line which touches the different elements that the user wants to select (Fig. 4e and f). This last method also “copies” the selected strokes into a clipboard (see Pasting in this section). The different methods help the user to do the selection of a group of strokes more easily under different scenarios: for instance, double-surrounding is easy and fast for complex drawings (like writing), while dot-selection is faster for selecting strokes in large drawings.
- Deselecting is done by clicking any empty space.
- Employing any of these methods more than once in succession will add or remove items from the selection so that the user can make complex selections using simple gestures.
- Pasting is done by drawing a dense point and releasing the stylus. This duplicates elements previously copied into the clipboard, and places them where the point has been drawn (see Fig. 5).
- When one or more items have been selected, two small handles appear at the right side of the selected strokes (a round blue dot and a small red square as seen in Fig. 6a). There is a set of simple actions for editing selected shapes:
 - Moving is accomplished by dragging any selected stroke or concept. Dragging will move the selection as a whole.

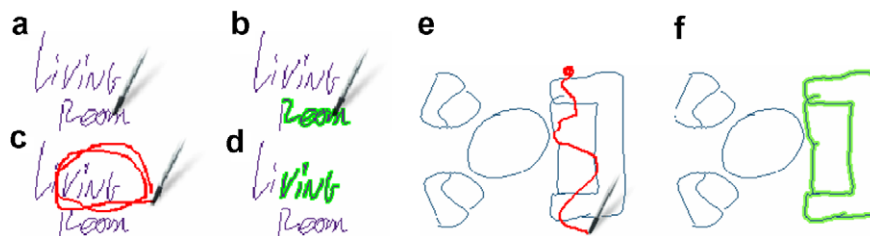


Fig. 4. (a) Holding the stylus on a trace (“room” on upper left) will (b) select connected graphics (partially darkened “r”, and “oom” on upper right). (c) Double-surrounding (“ving” on lower left) will (d) select enclosed traces (darkened “ving” on lower right). (e) Drawing a dense dot and then moving the stylus over other traces will (f) select all touched elements.

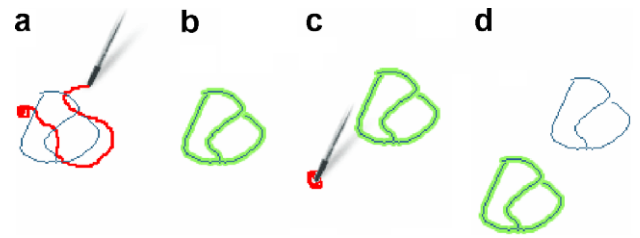


Fig. 5. (a) Drawing a dense dot and drawing over some sketches. (b) Selects and copies touched strokes. (c) With the dense dot gesture. (d) Copied graphics are pasted.



Fig. 6. (a) Resizing selected strokes by dragging the red square handle. (b) Rotating selected strokes by dragging the blue round handle. These handles are always available when any selection is done, but they have been removed from other figures in this paper to simplify understanding of explained features. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Resizing is done by dragging the red square handle, located at the upper right corner of the selected group of elements (see Fig. 6a).
- Rotating the selected elements can be achieved by dragging the blue round handle, located at selection’s right size (see Fig. 6b).
- Removing is performed by drawing a “connected cross” (see Fig. 7). If nothing is selected, this gesture removes every touched element. If one or more traces are currently selected, only those elements will be removed.
- Comments and annotations can be added “outside a page”. Clicking and dragging any corner of the current page, shrinks the size of its view, showing a darker background (see Fig. 8). This backdrop represents the “outside of the page” area, where notes or comments can be added. These annotations are also shared in real time with other users. This helps to understand sketches from other users, or remember the purpose of their own ones drawn some time ago. Users leave the annotations view by dragging the corner of the zoomed page back to the screen edge.

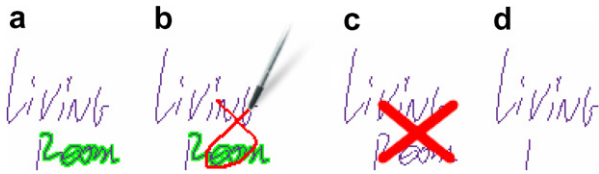


Fig. 7. (a) Some strokes are selected; (b) connected cross gesture represents Remove command; (c) feedback appears when command is recognized and (d) only selected strokes are removed.

- Any other gesture done with the stylus not matching the previous ones is considered as a drawn stroke.

Content editing is made through two steps: first simple selection gestures, and then with some basic editing gestures [25]. Chosen selection techniques and resizing and rotating interface allow users to create rich sketches with simple methods.

For sessions and users management there is a Session menu, including the usual New, Open, Save and Export as PDF functions, as well as a Synchronize command for entering/leaving the synchronized navigation. The Export as PDF function builds a PDF file with the expanded document tree, similar to the document tree view (Fig. 2). The file can be reproduced on any computer device using the

optimized PDF viewer included in the OS. Finally, the Session menu icon changes its shape to notify the user if the session needs to be saved because of changes introduced since it was last saved or opened (see Fig. 1).

Basic navigation actions are also triggered with gestures. To create a new node, the user double-clicks on any area. This will open a new empty sheet attached to the clicked location as a new design spot. Once created, a soft colored mark (customizable and yellow by default) behind the sketches indicates that the spot is a link to a new page. Double-clicking the spot will take the user inside the page, expanding its view to full-screen size. On sub-pages (everywhere but the root sheet) the margin of the screen will be shown progressively darker as the depth of the page in the document’s tree grows (see Fig. 1). Double-clicking this margin returns the user to the parent node. Repeating this action will eventually move the user to the root sheet.

5.2. Awareness

In order to avoid overlapping sketches, the system displays an alert showing where other users are currently drawing (see Fig. 9). Although this feature alerts users to avoid sketching in the same location, they may still draw in the same area if desired.

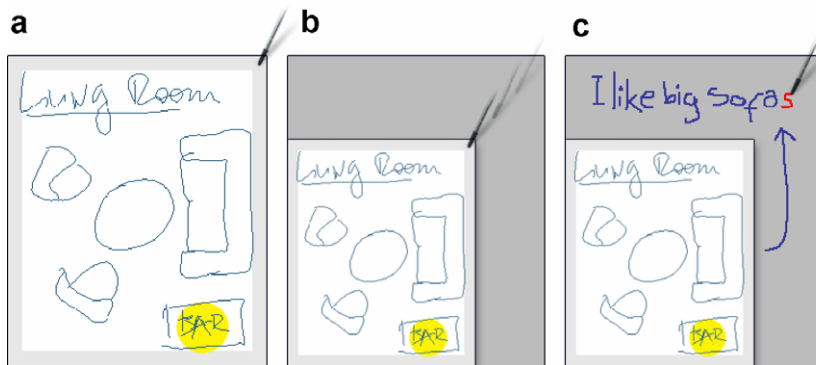


Fig. 8. (a) Original page, displayed at full screen. (b) The user drags the corner of the page, displaying the space “out of the page” for annotations. (c) Annotations outside the page can be done using the same sketching techniques for drawing and writing inside.

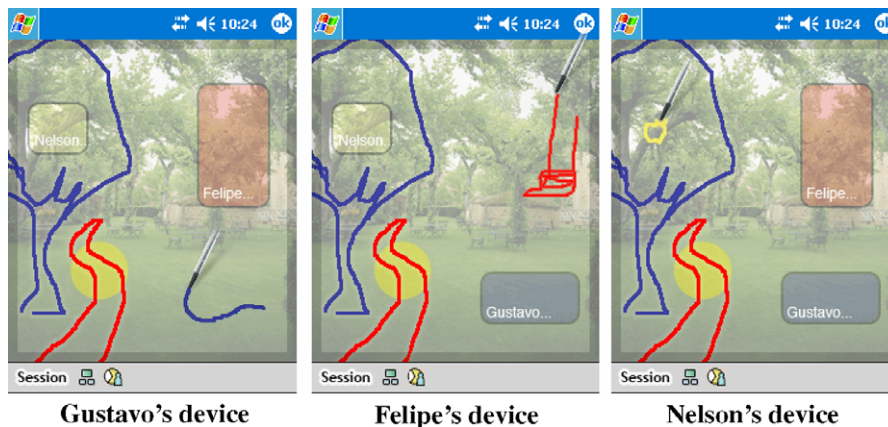


Fig. 9. In the example, three users draw at the same time. Each one is aware of where the others are drawing thanks to shown drawing feedback. This helps users to avoid drawing at the same space.

Already during the evaluation of the system it was clear that there was a need of feedback from the system to the user when a gesture was recognized and the corresponding action triggered. Thus, MCSketcher displays different kind of feedback for each recognized gesture:

- Removing displays a cross.
- Entering and exiting from node displays “zoom-in/out rectangles”.
- Removing cross (see Fig. 6c) is displayed for every user currently looking to the sketches being deleted.
- Dense-dot (for dot-selection and paste gestures) turns the “ink” into a different color to inform the user, the dot will be analyzed as a gesture and not as a sketch. This informs the user about how long he should wait before continuing the gesture.

Sound alerts or *earcons* [11] acknowledge and confirm a user’s action. Three such sounds have been included, one for zooming in (entering a design spot), another for zooming out (exiting a design spot) and a third for creating a new design spot.

There are some context awareness mechanisms implemented: an indicator on the menu bar shows the number of users in the current node (“group icon” of Fig. 1). If the other users in the current node are relatively few, clicking on it will switch the user to the tree’s busiest node in order to join the group. As mentioned in Section 4.1, the document tree view displays the full document map and highlights the user’s current location, as shown in Fig. 2. This view can be displayed by pressing the “document tree icon” (Fig. 1). The user can click any element of the tree to access that page.

6. Evaluations of MCSketcher

We made a usability test of the system using think-aloud method to measure its effectiveness. The test was aimed to detect problems in: (a) the design collaboration supported by the system, identifying the circumstances where the system does not provide the adequate support for or hinders the collaborative design work among the members; (b) the user interface, identifying which aspects make the understanding and learning the way it works difficult, the user satisfaction when using the diverse functionalities, particularly, when sketching, using gestures, and navigating. The results of the test show that MCSketcher is an effective tool for collaborative design on the field, and the interface also provides an effective way of communicating design ideas between the participants.

6.1. Evaluation parameters

In the think-aloud evaluation method users verbalize their thoughts about the system while using it, thus avoiding distortions, wrong cognitive translations or omissions that may be introduced by a guided questionnaire [26].

The process is observed by a person, who is in charge of gathering the oral information in real time while the participants are moving on the field designing something related to their work. According to [21], this method can be used to evaluate the computer-supported collaboration inside a working group. In each evaluating session, the participants work on a different real collaborative design on the field issue, establishing social interactions of communications and verbalizing their thoughts at the same time. During the collaborative design session, the observer is responsible for recording its development and the way the participants use the system. In this way it is possible to register the usability problems detected and/or verbalized by the participants in real time. Seven think-aloud test of real collaborative design activity on the field were organized with 3–4 people each. Four groups were engaged in construction design, of which two used a photograph as background for the first page. Three more groups were engaged in a garden design activity, one of them using a photograph of the field as background. The task for the team engaged in construction was to make a proposal about the final shape of a certain building which was not ready and its construction had stopped for already one year. The people engaged in the garden design were given the task of designing a small park located in a totally abandoned area.

Before starting the first session of every group, people were instructed on how to use the system during 10–15 min. During the next session we asked participants to engage themselves on mapping the system for about 30 min (this means, playing around with the system in order to “map” the functionalities of the system in their minds). All other session lasted for about 25–35 min, in which the observer took notes about usability problems detected and verbally expressed by the participants.

6.2. MCSketcher effectiveness

To measure MCSketcher effectiveness, during the tests we looked at the complexity of the collaborative design produced, the time to complete the task, how useful was the system for the participants, and how easy was using the interface based in sketches and gestures. The testing showed no critical usability or effectiveness problems regarding the design support and interface use aspect issues. Designers found that learning how to use MCSketcher was not a difficult task in general, although some problems were detected at the beginning with the use of the gesture for selecting and deleting, which required some time to be internalized by the users. Users from all groups were able to produce a collaborative sketch for the design task in no more than 35 min. The designs users produced during the testing were of low or medium complexity, sometimes proposing a totally new design and sometimes proposing a modification to the current situation of the construction. However, in both cases participants tended to leave important parts of the design in a rough state. For example, they did not spend much time trying to resize

or rotate objects; however the action of moving a selected part of the sketch was the most used functionality. This confirms the hypothesis that the system supports collaborative design at an early stage.

Some of the provided functionalities were not used very often. This was the case of the annotation functionality. They preferred to “annotate” their sketches in the page itself. Also only few designers used the functionality for copying and pasting part of the sketch, but this can be attributed more to the nature of the tasks than to the difficulty to use them.

The set of implemented awareness mechanisms worked as expected in most cases. Some of them were implemented after the first prototype, so we had previous information about their need. This was the case for users trying to write in the same region of the page. After introducing the awareness mechanism the number of “collisions” was much smaller. Also the feedback to all users that a certain part of the sketch was deleted avoided the problem that other users did not know what happened to a certain portion of the sketch. Another functionality which was introduced after the first prototype and improved the awareness of the users about what was going on during the synchronized navigation of the document was the use of earcons when entering and leaving a page. Earcons were useful for confirming user actions, eliminating ambiguity regarding system response to similarly triggered actions.

Of course, synchronized navigation has shown to have more awareness problems than independent browsing of the document. However, users worked much more often synchronized than independently. This may be an indication that implemented functionalities supporting navigation mode do really help.

Conceptual maps and hierarchical navigation were found to overcome the limitations of the handhelds’ small screen size. “Design-spots”, “document tree icon”, and dark screen borders were found to be simple to understand and to deal with, and they were considered effective. On average, the resulting documents with the designs used five screens (pages, nodes), with no more than three design-spots for each screen.

The “document tree icon” view proved to be good enough to maintain awareness about the general structure of the document and about all hierarchical pages. Therefore the zoom feature has been discarded. Sketches were found to be easily recognizable even when reduced to a minimal size, thus obviating the need to zoom and scroll, thereby increasing the simplicity and usability of the system.

The few gestures used (press-and-hold, double-click, double-surround and connected-cross) were sufficient to master all the functions of the system, thus keeping the collaborative design process simple and improving system usability. Due to the simplicity and contextualization of the permitted gestures, we found it was not necessary to differentiate sketching from gesture recognition in separate modes.

Final users found it easy to communicate ideas while maintaining face-to-face conversations and using sketches to explain their designs. Many designers said the system was great for producing an early design proposal without getting into the details, at the same time they found it provided good support to communicate their ideas.

7. Discussion and future work

We believe that the most significant contribution of the work reported here is to have combined the various data communication capabilities of sketching in a single product supported by the mobility of wirelessly interconnected handhelds and a natural gestures-based interaction style to facilitate collaborative design.

MCSketcher is a lightweight design collaborative system that enables group members to easily share their ideas through sketches using simple actions. Sketching and gesturing with a pen-based system in a one mode interaction has shown to be easy to handle and especially valuable for creative collaborative design on the field tasks. For designers, the ability to rapidly sketch objects with not too precise sizes, shapes, and positions, is important in order not to hinder their creativity or the process flow. The designers can explore more ideas without being burdened by concerns about less important details (colors, fonts, alignment, etc.). In the early phase, sketching also improves communication, both with collaborators and the target audience of the designed artifact. For example, an audience examining a sketched interface design, will be inclined to focus on the important issues at this early stage, such as the overall structure and flow of the interaction, without being distracted by the details of the look.

The results described in Section 6 show that, MCSketcher provides the user with constant and opportunistic awareness information. MCSketcher is still evolving and there is plenty of room for making more formal usability and utility evaluations which will improve the future development but the preliminary empirical testing suggests that the development is going in the right direction. Various new functionalities and features are already planned to be included soon. One of these is the use of more earcons for providing feedback to improve management of objects, operations and system interactions.

Another planned extension is the introduction of gesture rendering for supporting specific design work. In architectural design, for example, elements representing furniture, bathroom fixtures, etc., could be placed in the design by sketching certain strokes. Similarly, to design graphical interfaces the buttons, text input areas and dialog boxes could be incorporated using simpler strokes such as squares. To facilitate development of functionalities to support different design types, sets of rendered sketches could be added and removed as plug-ins to or from the existing system. This would also avoid any conflicts resulting from a given sketch having different meanings in different contexts.

Acknowledgements

This paper was partially funded by Fondecyt 1050601 and DI – Universidad de Chile Nro. I2 04/01-2.

References

- [1] W. Citrin, P. Hamill, M. Gross, A. Warmack, Support for mobile penbased applications, in: *Proceedings of MOBICOM*, 1997, pp. 241–247.
- [2] B. Myers, H. Stiel, R. Gargiulo, Collaboration using multiple PDAs connected to a PC, in: *Workshop on Shared Environments to Support Face-to-Face Collaboration CSCW*, 2000, pp. 285–294.
- [3] P. Luff, C. Heath, Mobility in Collaboration, in: *Proceedings of Computer Supported Collaborative Work CSCW98*, ACM Press, 1998, pp. 305–314.
- [4] G. Xexéo, A. Vivacqua, J. Moreira de Souza, B. Braga, J. Nogueira, B. Zinder, R. Castilho, B. Miranda, COE: a collaborative ontology editor based on a peer-to-peer framework, in: Weiming Shen (Ed.), *Advanced Engineering Informatics*, vol. 19, 2005, pp. 113–121.
- [5] S. Weiming, Special issue on collaborative environments for design and manufacturing, *Advanced Engineering Informatics* 19 (2005) 79.
- [6] E. Lank, S. Phan, Focus + context sketching on a pocket PC, in: *Proceedings of CHI Extended Abstracts*, 2004, pp. 1275–1278.
- [7] A. May, V. Mitchell, S. Bowden, T. Thorpe, Opportunities and challenges for location aware computing in the construction industry, in: *Proceedings of Mobile HCI*, 2005, pp. 255–258.
- [8] J. Landay, B. Myers, Sketching interfaces: toward more human interface design, *IEEE Computer* 34 (3) (2001) 56–64.
- [9] R. van der Lugt, Functions of sketching in design idea generation meetings, in: T.T. Hewett, T. Kavanagh (Eds.), *Creativity and Cognition*, ACM, New York, 2002, pp. 72–79.
- [10] L. Yang, J.A. Landay, Z. Guan, X. Ren, G. Dai, Sketching informal presentations, in: *Fifth ACM International Conference on Multimodal Interfaces ICMI-PUI*, Vancouver, 2003, pp. 234–240.
- [11] S. Brewster, Overcoming the lack of screen space on mobile computers, *Personal and Ubiquitous Computing* 6 (2002) 188–205.
- [12] G. Dai, H. Wang, Physical object icons buttons gesture (PIBG): a new interaction paradigm with pen, in: *Proceedings of CSCWD 2004*, LNCS 3168, 2005, pp. 11–20.
- [13] K. Forbus, J. Usher, V. Chapman, Sketching for military courses of action diagrams, *Intelligent User Interfaces* (2003) 61–68.
- [14] Z. Fan, M. Chi, M. Oliveira, Sketch-Based Collaborative Design System, *SIBGRAPI* (2003) 125–131.
- [15] D. Kenneth, R. Forbus, J. Usher, Towards a computational model of sketching, in: *Proceedings of Intelligent User Interfaces*, 2001, pp. 77–83.
- [16] V. Goel, *Sketches of Thought*, MIT Press, Cambridge, 1995.
- [17] U. Hoppe K. Gaßner, Integrating collaborative concept mapping tools with group memory and retrieval functions, in: *Proceedings of the Computer Support for Collaborative Learning, (CSCL) 2002 Conference*, 2002, pp. 716–725.
- [18] R. Zeleznik, K. Herndon, J. Hughes, SKETCH: an interface for sketching 3D scenes, in: *SIGGRAPH 1996*, ACM, 1996, pp. 163–170.
- [19] J. LaViola, L. Holden, A. Forsberg, D. Bhuphaibool, R. Zeleznik, Collaborative conceptual modeling using the SKETCH framework, in: *Proceedings of the IASTED International Conference on Computer Graphics and Imaging*, 1998, pp. 154–157.
- [20] R. Davis, J. Landay, V. Chen, J. Huang, R. Lee, J. Li, J. Lin, C. Morrey, B. Schleimer, M. Price, B. Schilit, NotePals: lightweight note sharing by the group, for the group, in: *Proceeding of CHI*, 1999, pp. 338–345.
- [21] S. Cox, J. Perdomo, W. Thabet, Construction field data inspection using pocket PC technology, in: *International Council for Research and Innovation in Building and Construction, CIB w78 Conference*, 2002.
- [22] A. Schmidt, M. Lauff, M. Beigl, C.S.C.W. Handheld, Workshop on handheld, in: *Proceedings of the Computer Supported Collaborative Work – CSCW'98 November*, Seattle, 1998.
- [23] A. Long, J. Landay, L. Rowe, PDA and gesture use in practice: insights for designers of pen-based user interfaces, April from <<http://bmrc.berkeley.edu/research/publications/1997/142/clong.html>> (retrieved on 2007).
- [24] K. Hinckley, P. Baudisch, G. Ramos, F. Guimbretiere, Design and analysis of delimiters for selection-action pen gesture phrases in Scriboli, in: *Proceeding of CHI 2005 ACM*, 2005, pp. 451–460.
- [25] L. Denoue, P. Chiu, T. Fuse, in: *Conference on Human Factors in Computing Systems, CHI'03 Extended Abstracts on Human Factors in Computing*, Ft Lauderdale, Florida, USA, 2003, pp. 710–711.
- [26] T. Boren, J. Ramey, Thinking aloud: reconciling theory and practice, *IEEE Transactions on Professional Communication* 30 (3) (2003) 261–278.