Wireless Internet Collaboration System on Smartphones

Nadya Belov,
Ilya Braude,
Werner Krandick,
and
Joshua Shaffer

Technical Report DU-CS-05-05
Department of Computer Science
Drexel University
Philadelphia, PA 19104
April, 2005

Wireless Internet Collaboration System on Smartphones

Nadya Belov and Ilya Braude and Werner Krandick and Joshua Shaffer

Department of Computer Science
Drexel University, Philadelphia PA 19104, USA

Abstract. The Wireless Internet Collaboration System allows wireless smartphones to be used for mathematics communication, that is, for the creation and exchange of mathematical formulas, diagrams, and text between two or more participants. The system solves the problem of providing the expressive power of mathematical notation on devices with limited input, output, and processing capabilities. The system also provides a turn-management protocol that supports mathematical dialogue between two or more participants who are geographically dispersed and thus cannot use gestures or glances for communication. The architecture of the system is extensible and can be adapted to heterogeneous and evolving hardware platforms.

1 Introduction

As wireless smartphones accompany increasing numbers of users at all times the devices offer themselves as the medium of choice for improvised synchronous collaboration. However, the small size of smartphones—while necessary for their ubiquity—seems to limit the complexity of collaborative tasks that can be carried out. In an attempt to explore—and push—those limits we started to investigate how smartphones can support mathematical teamwork.

The domain of mathematics was carefully chosen. The challenge of representing mathematics in typeset form has led to the development of document preparation systems such as TEX and LATEX which are in widespread use today. Mathematical handwriting recognition which requires the recognition of up to about $2,000$ distinct symbols continues to push the limits of general handwriting recognition. We also believe that the cognitive challenges that arise in intellectual teamwork in any domain also arise in the more universal domain of mathematical collaboration.

An immediate need for mathematical collaboration tools exists in international online communities such as the Math Forum [1] which is headquartered at Drexel University. Students and teachers of all levels use the Math Forum in order to ask questions, solve problems, and pursue interests in mathematics. The service *Ask Dr. Math* receives as many as 9,000 e-mail questions a month and over 200 trained volunteers—teachers, professionals in the workplace, college students, and hobbyists—answer close to 4,000 of these directly within a few hours of their submission. As smartphones become more widespread some of these activities might be carried out using smartphones, and the potential for synchronous mathematics communication might increase.

The WIRELESS INTERNET COLLABORATION SYSTEM (WICS) allows wireless smartphones to be used for the creation and exchange of mathematical formulas, diagrams, and text between two or more participants. The WICS enables two or more persons with smartphones or traditional computers to participate in a session that may convey textual, graphical, and mathematical information to the other participants. Users can draw, edit, and label geometric shapes, send chat messages, and compose formulas. A turn taking mechanism moderates the communication. The system can be extended by adding services that provide individual users with additional functionality.

The WICS solves a number of technical challenges. Among them is the problem of providing the expressive power of high-quality formal and informal mathematical notation on devices with limited input, output, and processing capabilities. The WICS also provides a protocol that supports mathematical dialogue between two or more participants who are geographically dispersed, and cannot use gestures or glances for communication. Finally, the WICS architecture is extensible and thus adaptable to heterogeneous and evolving hardware platforms.

A number of remote collaborative systems were developed in the early 1990s (e.g. *Liveboard* [2]). *Liveboard* allows participants to share their thoughts and ideas on one common whiteboard area. The system is extensible, and can interface with portable and non-portable computing devices to allow contributions to the whiteboard from heterogeneous sources.

The *Pebbles* project [3] is an example of facilitating collaboration on mobile devices. PDAs are used as "remote controls" through which users are able to operate the mouse and keyboard of a computer during a meeting. Additionally, the *Pebbles* project has been expanded to include a shared whiteboard application in which the users can make annotations simultaneously. This addresses the need for diagrams to support collaboration has long been recognized in the domain of education [4].

A paper by Dustdar and Gall [5] introduces a peer-to-peer architecture with similar goals to the client/server architecture of WICS, namely to allow participants to collaborate with each other on many platforms, including the mobile platform.

The WICS architecture is a multi-domain service as discussed by Jørstad, Dustdar and Do [6]. Our Internet-based system provides services regardless of where the user is roaming. WICS services are available and can be accessed by smartphones, PDAs, and desktop or laptop computers.

We derive the specifications for WICS by revisiting whiteboard communication for mathematics in Section 2. We observe collaborative whiteboard sessions and outline the set of features WICS needs to support. An example WICS session is described in Section 3. In Section 4 we describe the WICS architecture. Section 5 discusses features of the user interface of WICS that alleviate the cognitive load of the users. Section 6 introduces the platform on which the WICS was implemented, along with some of the challenges imposed by smartphones. Lastly, Section 7 concludes the paper by providing future directions and extensions of the WICS.

2 Whiteboard Communication

Collaboration and mathematical communication, especially when motivated by tutoring, have traditionally been accomplished in front of a physical whiteboard, where every participant can publicly display his or her ideas. Furthermore, although not always obvious to each participant, gestures, physical cues, and voice intonations are relied upon to aid in communication [7].

In order to design a system that provides a means for virtual collaboration, it is necessary to identify specific key features of traditional collaboration. This is a similar approach to that of the *Colab* project at Xerox Park [8], whose purpose was to study the habits and practices of meetings of small groups (of at most six participants). Similar to the observation methods of *Colab*, we chose to observe a series of collaborative sessions in which a student and teacher attempted to solve a mathematical problem. The sessions between student and teacher were transitioned from their traditional environment (face-to-face collaboration) to the environment typical for virtual collaboration sessions. The observations of these collaborative sessions were designed to highlight environmental differences. Specifically, participants are not present in the same room and cannot communicate using voice. These sessions are broken into three distinct types:

1. *Traditional Collaboration*: eye contact, conversation and whiteboard annotation are allowed;
2. *Eye Contact Deprived*: conversation and whiteboard annotation are allowed; and
3. *Eye Contact and Conversation Deprived*: whiteboard annotation are allowed.

Through observation of the participants' behavior, we extract the necessary features required for a wireless collaborative system. Specifically, we observe that participants employed a turn taking structure throughout all types of collaborative sessions (face-to-face and virtual). Additionally, participants utilized the whiteboard far more extensively for small notes and captions as their ability to communicate face-to-face and utilize gesturing and voice communication were eliminated. Furthermore, several key stages of communication were observed, each with its own characteristics. In our observation of a traditional collaboration session, three distinct problem solving stages emerged:

1. *problem understanding*, requiring gesturing and physical queuing;
2. *solution proposition*, which can be iterated over several times, requiring voice interaction along with graphical support on the whiteboard; and
3. *solution resolution*, requiring intensive writing and editing of the solution, as well evaluation of the validity of the solution.

As is described in §6, current consumer smartphone technology does not permit concurrent use of the Internet and voice communication. Furthermore, given the motivation for such a technology, it is not yet clear whether the ability to converse on the phone while using the collaborative system will benefit the user. Namely, if the system can be utilized ad-hoc, anywhere and at anytime, it may be distracting to those around the participants to hear the entire session's conversation. For example, if one of the members in a particular session is on a train, it may not be appropriate to speak on the phone loudly, revealing the information discussed amongst the members. Therefore, we chose to design WICS without voice communication. Table 1 shows how effective communication can be facilitated through channels other than voice. Specifically, each stage of the collaboration process listed under *Contents* in Table 1 can be facilitated through multiple mediums. The first two columns in the table show the various options that we have evaluated to be appropriate through our observations of collaborative sessions.

Basic Required Comm. (Opt 1)	Basic Required Comm. (Opt 2)	Contents
Gesturing & Verbal	Writing	Problem Understanding
Gesturing & Verbal → Writing	Writing	Understanding of Prerequisites
Verbal	Verbal → Writing	Approach Description
Writing	Writing	Notation definition
Writing	Writing	Approach Implementation
Gesturing & Verbal	Verbal	Result Evaluation (= Solution)

Table 1. Problem Solving Flow.

Furthermore, people tend to naturally follow common communication protocols such as the *phone conversation protocol* [9]. Communication protocols are also observable, albeit without much structure, in classroom situations and study sessions where the communication is many-to-many.

Because we are focusing on facilitating mobile collaboration in the many-to-many paradigm, we studied possible natural communication patterns that could be mimicked in an ad-hoc setting. Establishing a turn taking protocol is especially important in a mobile collaboration system. Participants are not able to see each other during their communication session, eliminating their ability to respond to physical cues that may help participants decide to give up or take their turn. Lastly, because participants are not able to see each other's reactions in virtual sessions, labeling of the figures and diagrams that are drawn on the shared whiteboard becomes of the utmost importance. For example, in a typical face-to-face collaboration session, simply pointing at the actual figure may be enough to emphasize it. However, since a participant cannot be guaranteed that the user they are communicating with is looking at the screen or at a particular diagram at that moment, he or she must label the diagrams and then refer to them in his or her explanation. Therefore, in observing the diverse set of collaborative sessions, we have learned that effective labeling and explicit turn management mechanisms must be integrated into any virtual collaboration system in order to assure effective communication.

3 WICS End-User Perspective

As WICS is a collaborative tool, the interface through which it is used is of paramount importance. This section provides a description of the user interface and its use.

3.1 Graphical User Interface

User interaction with the WICS system occurs through the WICS GUI. The graphical user interface consists of several modes: *Chat*, *Whiteboard*, *Caption*, *LaTeXService*, *Information*, and *Palette*. The user is able to switch from one mode to another via the menu system provided on the screen. The *Whiteboard* and *LaTeX* modes are examples. A more detailed description of the user functionality provided in each mode is discussed below.

Chat The *Chat* mode allows users to communicate via traditional chat messages that are relayed by the server to everyone in a session. The chat mode is reachable through the navigation menu. A user can type in a message and press the `Send` button. Their message is then relayed to all the other session participants. The format in which messages appear on the screen is as follows: "`username: message`".

Whiteboard The *Whiteboard* mode allows users to make annotations on a shared whiteboard. Users can draw various shapes in many colors, draw freehand, type text and even place formulas on the whiteboard. Like the *Chat* mode, the *Whiteboard* mode is reachable via the navigation menu.

Caption *Caption* mode allows the entry of text that is used as a supplementary description of an annotation on the whiteboard.

LaTeXService The *LaTeXService* mode is reachable via the navigation menu and displays an entry field for an expression using LaTeX notation. Because the WICS system is designed to be usable on smartphones with small keyboards, special characters such as caret which are hard enter via the keyboard in a timely manner, can be found in a character palette on the same screen. The user is able to enter his expression through a combination of selected keys from the virtual key palette and physical keyboard, press `Okay` and preview his entry. The ready and approved LaTeX expression is displayed in the *Whiteboard* mode where the user is able to use it as part of his explanation.

Information The *Information* mode provides the user with her status information including connection status, token possession status and names of other session members. Although not a vital mode, the *Information* mode keeps the user well informed as to the status of her session communication.

Palette The *Palette* mode allows the user to select a tool which he or she wishes to use for whiteboard annotation. Additionally, this mode provides customization preferences such as shape colors, line thickness and any other system preferences.

3.2 GUI example

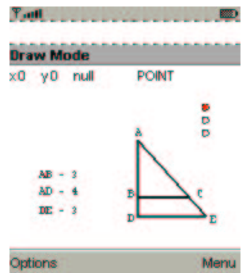
Collaborative sessions begin with logging in (c.f., Figure 1(a)) to the WICS Server and then joining or creating a session (c.f., Figure 1(b)). After the user enters a session, participants may begin their mathematical collaboration. The collaborative process is mediated through the turn-taking mechanism described in § 4.2. The importance of this is how it effects to change the modalities of communication, allowing modeling of the kinds of dialog that one finds common in informal mathematical collaboration in the form of glances and gestures that seek for acknowledgment.

Additionally, the whiteboard provides an input system that is appropriate for communicating about mathematics (c.f., Figures 1(c), 1(c), 1(d)), providing both tools for drafting figures (with the stylus), as well as writing equations through the LaTeX service, as demonstrated in the aforementioned figures.

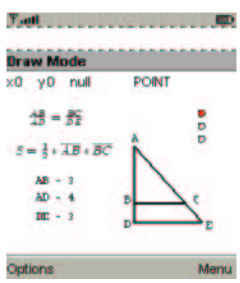
WICS also provides a relay style text chat, allowing collaborators to use natural language when appropriate (c.f. Figure 1(e)).

(a) WICS: Login Screen

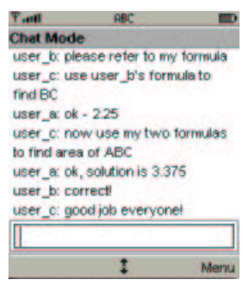
(b) WICS Session Creation Screen



(c) WICS Whiteboard Screen with Problem Description



(d) WICS Whiteboard Screen After Problem Note Addition and User-contributed LaTeX Formulas



(e) WICS Partial Chat Transcript of Arrival at Solution

Fig. 1. Problem solving session storyboard.

4 System Architecture

The WICSis an ad-hoc, dynamic, and portable tutoring system designed to provide a means of communication in the mathematical domain between two or more users. WICS is cross-platform and lightweight in its footprint; its small bandwidth requirements allow for the utilization of networking services of cell phone providers. WICS uses IBM's WebSphere JVM which is 1.6MB, and WICS code requires only 79Kb of storage space on the Treo. The system provides the users with the necessary capabilities to express mathematical problems and verbal explanations through its mixed-initiative enabled user-interface.

4.1 WICS Architecture

The WICS application suite is built on top of the Java platform. It utilizes both the Java 2 Standard Edition (J2SE) and the Java 2 Micro Edition (J2ME) platforms. J2ME is build to run on mobile devices such as mobile phones and PDAs. The Java abstraction layer allows us to port the system to various architectures easily.

The WICS network design mostly resembles a traditional client-server architecture. However, it allows for independent, and possibly 3rd party services to be run on multiple servers independent of the main WICS server. WICS clients are able to utilize these services using a Service Manager. Figure 2 shows a high level diagram of the WICS architecture.

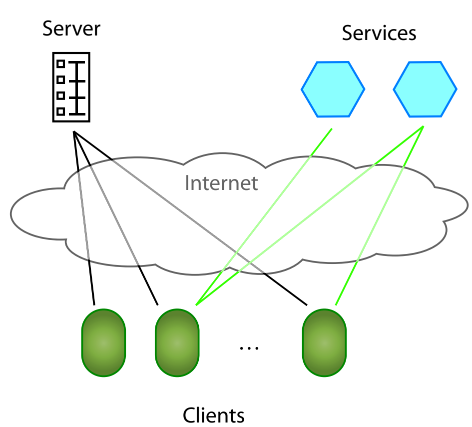
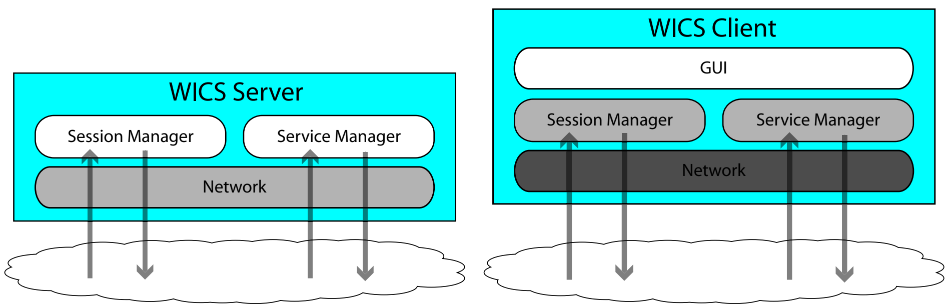


Fig. 2. WICS Network Architecture

The wireless connection medium that we use in our prototype is provided by the data transfer functionality of the mobile service provider. This allows us to access WICS services anywhere a user is able to use her phone.

WICS Server The WICS server provides authentication, session control, validation, and other services for the clients. The server's primary responsibility is to manage and maintain user sessions. In our system, a user session consists of at least one user. A session must have a unique id, designated by its creator, the first user to join this particular session. Each user in a session also has a unique id, required by the server for communication purposes. The server communicates with each user individually, by receiving messages which specify an annotation (textual or graphical) proposed by the user. The server verifies the appropriateness of the annotation and propagates the change to all the other users in the session. See Figure 3(a) for a high level diagram of the server.



(a) WICS Server Architecture

(b) WICS Client Architecture

WICS Client Each user runs his or her own client that participates in a communication session through the WICS server. The client has the capabilities to recreate all of the annotations it has received from the server. Specifically, let us imagine a scenario where two users, *a* and *b* are collaborating together. User *a* draws a circle in the middle of the screen. As soon as *a* is finished drawing the circle, a message regarding the circle is relayed by *a*'s client to the server. The message contains the type of the shape drawn, the color and the exact coordinates of the figure as it appears on *a*'s screen. As described in §4.1, the message is passed to user *b*'s client which is then able to recreate the annotation for user *b* to see. If a client joins a session that is already in progress, it will be updated with the latest state of session. The main task of the client is to recreate the annotations of which it is notified of by the server and to relay its own user's annotations to the server for propagation to all of the session members. The client's other main task is to facilitate the communication with any services (see §4.1) that may be available. Although, already implied by the description of the architecture, it is important to point out that the client provides a graphical user interface, as shown in Figure 3(b). The interface provides tools for the user to draw and label figures as well as augment drawings with captions and communicate using text messaging.

WICS Services WICS architecture furthermore supports interaction with outside services. Since the message protocol utilized in WICS is easily parsed and readable, many outside applications can be extended with a façade to work with the system. In turn, these services can also greatly extend the functionality of the WICS system.

Services are run completely independent of the WICS system. They are published to the clients through a directory on the WICS server. When a service starts, it notifies the WICS server of its name and capabilities. Clients are then able to take advantage of the services that are advertised by the server to augment their own capabilities. However, as shown in Figure 3, once a client is aware of a service, the communication is performed directly between the client and the service, not through the WICS server.

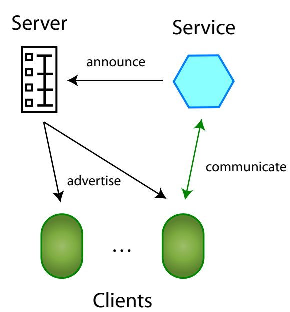


Fig. 3. WICS Services Architecture

Services can greatly accelerate the user's task by rendering high quality math formulas in LaTeX, provide ready-made diagrams, factor formulas, solve equations, graph functions, and more. Each client keeps an individual service manager, as some services may be available to some clients and not to others. For example, if a client is able to render its own LaTeX formula or plot its own graph, it will not need the functionality of those services.

The current version makes use of a LaTeX service which is primarily utilized to create well formed formulas that can be attached to diagrams describing various relationships and equalities. LaTeX math mode provides a rich set of expressions that we utilize to create small, easily readable and unambiguous expressions.

4.2 WICS Protocols

Network Protocol Each message exchanged between the server and a client, must conform to the WICS format.

The message structure as shown in Figure 4, has been designed specifically for this system and is extensible and rich in design to allow a variety of communication types.

A concept of a magic number specified in our protocol as `magic string` and is used as a means of assuring that the message originated from a WICS client or server. This string is always "wics" and occupies the first 4 bytes of the message header. The next byte specifies the `version` number which is used to allow flexibility in the WICS

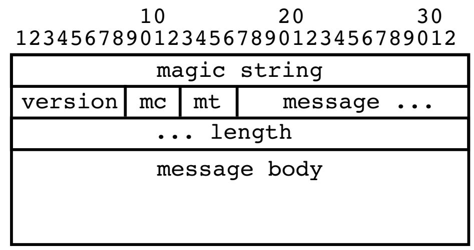


Fig. 4. WICS Message Protocol

system where it may be necessary to run a client and a server that are of different versions. Each version may have more or fewer capabilities and needs. By specifying the version number, the server will be able to service that particular client accordingly, or vice versa. Next, the message class (`mc`) and message type (`mt`) (4 bits each) are used to specify the exact operation and message that this particular message is used to carry. That is, for example, an update of a shape will have to specify that the message class is "update" and the message type is "shape". Using these two concepts the message protocol is flexible enough to allow for additions and updates without having to change the format of the message. Because the `mc` and `mt` fields are 4 bits each, a total of 2^8 different messages are supported by the protocol. The last two components of the message, `message length` and `message body` are used to specify the specifics that would come with the given message class and message type. The `message length` field is 6 bytes long and specifies the length of `message body` in bytes.

The message structure is flexible enough to allow text messages, as well as messages designated to notify of a newly drawn shape and its color, a new equation, and caption. The system does not communicate actual diagrams to all the users, every time a new one is created. Instead, all diagrams are described in a WICS message and each client is responsible for recreating and adding the changes on its own. Through this approach, we are able to retain lightweight communication suitable for even cell phone networks.

Turn taking Protocol The turn-taking protocol is designed to facilitate structured communication amongst the participants in a WICS session. The structure of the protocol along with its states and transitions is shown in Figure 5.

As shown in Figure 5, a turn-token can either be in use by a user in a session, or be available for any user in the session to claim it. A token can also be in use by a user and requested by another user. It is important to notify a user who currently possesses the turn-token of the request made by other user(s). This information will help the current token holder decide when to release the token. Figure 6 shows the state transition diagram for a token for a participant. Each participant can either possess or not possess a turn-token, and may request a token if he or she does not possess one. The user is notified of a request if he or she is currently in the possession of a turn-token and another user wishes to acquire possession of the token. Figure 6 shows the actions a user may take to achieve the desired state.

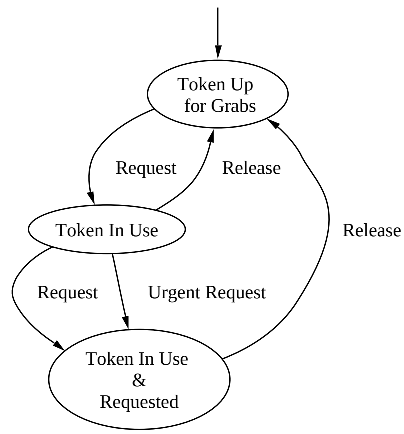


Fig. 5. Turn taking protocol: Token State Transition Diagram.

5 Mixed-Initiative Framework

Effective collaboration is facilitated by alleviating a user's cognitive load in performing rudimentary tasks. An example of such a task is interaction with a graphical user interface. Instead of concentrating on interacting with the system to produce the desired effect, the user should concentrate on the information he or she is attempting to express. Mixed-Initiative is a concept that facilitates task completion as part of a collaborative effort between the computer and the user [10]. WICS integrates a mixed-initiative framework that automates certain common time-consuming tasks.

5.1 Automatic Line Labeling

The WICS Client allows its users to collaborate using automatically labeled geometric line drawings suitable for a high school geometry class. The implicit engineering requirements of such a system are as follows: (1) labels must not overlap; (2) labels must be positioned such that little ambiguity arises; (3) textbook conventions should be emulated when possible; and (4) the output should be formatted the same way on all clients.

Free-form and parametric drawings, by themselves, are inappropriate for mathematical collaboration; figures must be labeled. However, systems that require users to manually label figures are too taxing, given the limited input devices standard on most smartphones. The trivial case of drawing and labeling a line requires the following steps:

1. Acquire possession of the stylus,
2. Draw the line,
3. Select label site,

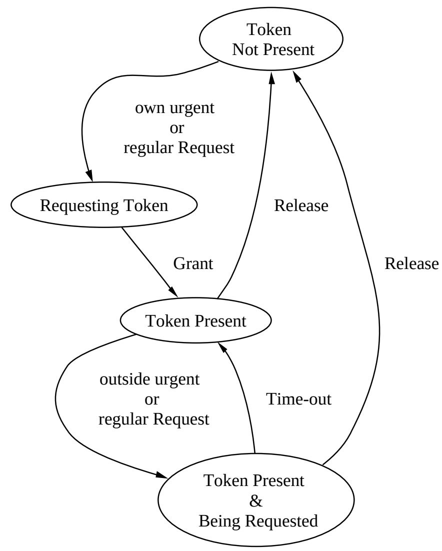


Fig. 6. Turn taking Protocol: Token State Diagram In Participant's View

4. Remove the stylus from contact with the screen, and
5. Enter a label for line.

If the system were able to automatically label figures, it would halve the number of steps required for input:

1. Acquire possession of the stylus,
2. Draw the line, and
3. Remove the stylus from contact with the screen.

In our formulation of the figure labeling problem, all mathematical figures can be represented as a set of control points from which labels may be drawn. In addition, a figure may consist of a single point. The problem of labeling a set of sites or points such that no two labels overlap has previously been studied, and has been proven *NP*-COMPLETE [11, 12]; thus, an optimal, algorithmic approach is not tractable.

In order to find candidate positions, each line is drawn with the Bresenham line-drawing algorithm on an ancillary canvas of integer values. These values are incremented where the line is drawn. In addition to the points on the ancillary canvas given by the line-drawing algorithm, points within a radius of ε are also incremented in inverse proportion to ε. Therefore, line segments that cross or that are close within ε will have a higher value on the ancillary canvas.

Textbook conventions have shown that the middle of a line segment is the preferred location for line labels [13, 14]. Therefore, after a line is drawn on the WICS whiteboard, the midpoint of the line is first sampled as a possible label location. The label is placed next to the midpoint and adjacent to the line if this value, extracted from the ancillary canvas is below a specified threshold value.

Otherwise, using a random number generator with a preset seed that is the same on all clients, a set of random points are selected on the line. The random point with the lowest value on the ancillary canvas in the set is then be used as the label position.

All line labels consist of a maximum of three characters chosen sequentially from the English language. After the candidate label location has been selected, the values on the ancillary canvas within a bounding box large enough for the label are incremented.

This line labeling algorithm is executed every time a new line is added for all lines on the whiteboard insuring that the labels are placed in the same location on each WICS client.

6 Smartphones–Treo 600

WICS is currently deployed on the Treo 600, however, since it is implemented in Java it is not bound to any specific hardware configuration or architecture. The Treo 600, shown in Figure 7, is commercially available in the United States, Europe and Asia. This smartphone runs the Palm 5 operating system. The Treo combines in its small size (11.2 $\times$ 6.0 $\times$ 2.2 cm, 170 g.) the following features: phone, organizer, digital camera, and Internet browsing capabilities. Lastly, the Treo operates with a 144Mhz processor with 32MB RAM, of which 24MB are available for storage.

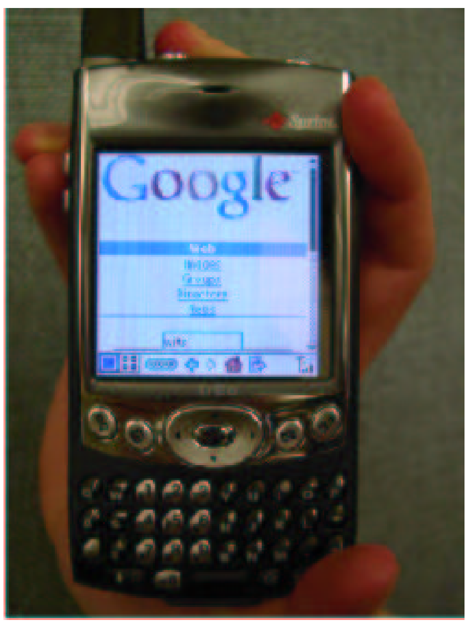


Fig. 7. Treo 600

6.1 Implementation Challenges

The implementation of the WICS system was carried out on the Treo 600 smartphone. The Treo is small and has a slow processor, at least compared to current state of the art in desktop computers. Usability assurance and the design of the graphical user interface posed significant challenges. The screen size limited the number of features that could be placed on a screen palette without depriving user-space. A small keyboard with many characters reachable through a combination of several key presses presented an additional challenge in the speed a user could enter special characters. This is especially applicable to the LaTeX service screen where a user needs to enter statements in the appropriate LaTeX format that utilizes many special characters such as the caret, square and curly brackets and ampersand among many others. In order to facilitate speedy entry of special characters, a character palette was developed and placed in the LaTeX mode.

Most portable computers including smartphones, cell phones and personal digital assistants are capable of running Java micro-edition applications. The micro-edition of Java lacks many libraries and data-structures that are otherwise provided in traditional Java editions. The lack of data-structures and libraries posed technical challenges in the layout of the graphical user interface. For example, a menu system was implemented instead of a tab system for all of the different user modes.

Internet connection on most smartphones including the Treo is provided by phone service provides via the Code-Division Multiple Access (CDMA) protocol. Although the protocol is fast in theory, practical network speeds are very slow. As a result of this challenge, an efficient communication protocol was needed to minimize the amount of information sent over the network and increase speed. Our network protocol, described in detail in §4, allows for variable-length messages that do not waste bandwidth.

7 Future Directions and Concluding Remarks

The WIRELESS INTERNET COLLABORATION SYSTEM, WICS, is a platform-independent, mobile collaborative system that allows improvised synchronous intellectual teamwork among geographically dispersed participants. WICS is designed to be usable on portable computing devices and has been implemented on the Treo 600 platform. The system design incorporates the notion of services, possibly outside of the system, to which the user may subscribe. At this time, the LaTeX service provides well-formed LaTeX formulas for publication on the shared whiteboard. Geometric figures are created together with labels that may be used in the accompanying formulas.

The future challenges for WICS include the inclusion of existing web content in a team session, the development of a front end for formula entry that can be used by casual users, increased customizability and adaptability of the user interface, and the addition of services such as computer algebra systems.

Making the software available under the GNU Public Licence will allow interested user communities to adopt the system or to change it according to their communication needs.

8 Acknowledgements

This research was supported in part by NSF ITR-Grant REC-0325872. The authors would like to express their gratitude to Dmitriy Bespalov, Daniel Lapadat, Andrew Ragone and for their contributions to the implementation of WICS.

References

1. http://www.mathforum.org: The math forum: An online math education community center (1994–2004)
2. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., Welch, B.: Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In: CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (1992) 599–607
3. Myers, B.A., Stiel, H., Gargiulo, R.: Collaboration using multiple pdas connected to a pc. In: CSCW '98: Proceedings of the 1998 ACM conference on Computer supported cooperative work, ACM Press (1998) 285–294
4. Kreutz, R., Kiesow, S., Spitzer, K.: NetChat: Communication and collaboration via WWW. Education Technology & Society **3** (2000)
5. Dustdar, S., Gall, H.: Architectural concerns in distributed and mobile collaborative systems. J. Syst. Archit. **49** (2003) 457–473
6. Ivar Jørstad, Schahram Dustdar, D.v.T.: Evolution of mobile services: An analysis of current architectures with prospect to future. In: Lecture Notes in Computer Science. Volume 3272. (2004) 125–137
7. Norwick, D., Hansen, B., Ward, K.: Coordinating turn-taking with gaze. In: Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference. Volume 3., Philadelphia, PA, IEEE (1996) 1888–1891
8. Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., Suchman, L.: Beyond the chalkboard: computer support for collaboration and problem solving in meetings. Commun. ACM **30** (1987) 32–47
9. Hiltz, S.R., Turoff, M.: Structuring computer-mediated communication systems to avoid information overload. Commun. ACM **28** (1985) 680–689
10. Horvitz, E.: Principles of mixed-initiative user interfaces. In: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems. (1999) 159–168
11. Wagner, F., Wolff, A.: Map labeling heuristics: provably good and practically useful. In: SCG '95: Proceedings of the eleventh annual symposium on Computational geometry, ACM Press (1995) 109–118
12. Wagner, F., Wolff, A.: A practical map labeling algorithm. Comput. Geom. Theory Appl. **7** (1997) 387–404
13. Richard N. Aufman, Richard D. Nation, D.K.C.: Applied College Algebra. Houghton Mifflin Company (2004)
14. Saunders, H.M., Carman, R.A.: Modern Technical Mathematics. Wadsworth Publishing Company (1984)