Instant Hoot

Alena Meade, Ekta Shah, Sushmita Subramanian

SAUI Fall 2006 Instructor: Jason Hong

Alena Meade | Ekta Shah | Sushmita Subramanian

Abstract

Instant Hoot is a system designed to improve the text messaging experience for smart phone users. The application takes into account users' calendar events, current location, current time, and users' text messaging history to identify the relevant communication scenario. It then prompts the user with the given scenario and automatically suggests a set of likely user responses that contain easily modifiable fields. Instant Hoot increases efficiency in text messaging and reduces the time spent by users interacting with their mobile devices to send quick messages.

1. Introduction

Mobile devices are convenient and a good way to communicate quickly. However, calling is not always appropriate in every scenario, such as when the other person is in a meeting or when users just want to reconfirm an event without having to talk. In these cases, text messaging seems to be the appropriate mode of communication given that people carry their phones everywhere. Yet, text messaging adoption rates have not climbed as quickly as cellular phone adoption rates in the United States, primarily because of additional cost, difficulty of use, and inefficiency. Instant Hoot targets these latter two problems and aims to improve text messaging on smart phones to make it both faster and easier to use. We hope to do this by creating a contextually aware system that provides relevant information to the user and reduces redundancy for the user via automation.

2. Previous Work

Based on current mobile instant messaging and SMS practices, development of contextual messaging systems could provide a great benefit to users in terms of speed and relevance in communications. Karen Tang and Jason Hong at

Carnegie Mellon University found that users regard mobile IM and SMS communications to be more functional and purpose driven than desktop/laptop IM, and that a greater portion of text communications on mobile devices were organizational in nature.[1] Currently, however, there are no context-capturing text communication applications for mobile devices. Improving context-based communications is therefore an established topic in research and application development for mobile devices.

In another study on the communications breakdowns of dual-career families, Tang, Hong and Siewiorek found that "There is a need for increased contextual awareness.[2] People are open to the idea of contextual technology aiding their lives." The study examined the communication practices of families with at least two children and in which both parents were in the workforce. It was found that status check message, confirmation messages, and scheduling and rescheduling messages were the most frequent communications between family members. The results also showed that the families were interested in a mobile coordination system which offered contextual reminders, which they felt could reduce the number of coordination messages. Furthermore, studies on the research group's prototype coordination system *inTouch*, found that users found form-based messaging an attractive feature of text messaging systems, as it made text messaging more approachable. Participants were not concerned about losing their personal writing style with a form-based system, provided that they could also edit the form-based message. This study again showed user's need and desire for a context-based message coordination system for mobile devices and illustrated that users would be receptive to template-like messages given the ability to customize them.

However, there is an inherent tension between automating contextual-awareness and preserving the correctness and flexibility with which humans respond in context-based communication scenarios.

Schilit, Hilbert and Trevor break these scenarios down into two steps: context acquisition and communication based on the context.[3] "It is not obvious that application designers should simultaneously try to maximize autonomy in both dimensions since this removes human common sense…."(Figure 1). In the case of cell-phone communications this is particularly relevant due to the wide range of communications which are performed on these devices, from business scenarios to personal conversations.

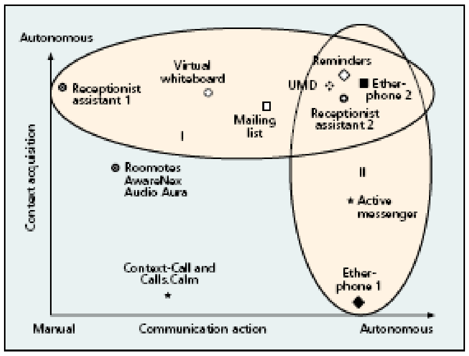


Figure 1: Context-aware communication dimensions. [3]

Schilit, et al furthermore suggest several considerations in the development of contextual technology. These include, among others, improving relevance of information, minimizing disruption to the user and reducing overload. Many of these considerations are based on reducing the amount of information for users to process. Information which is not relevant to the user's context may be filtered out and automatic notifications may bypass the user. For instance, a way to minimize disruption might be to notify a caller when you are at the movies before the call goes through.

Location is another important aspect of context-aware mobile systems. Smith et al [4], developed a location disclosure system to inform users where their contacts were currently located. In response to a "Where are you?" message, the system can, after an explicit confirmation by the user, send out the location information. Locations can be customized to meaningful aliases such as "at lab" or "Joe's Diner". The system could also be set to issue automatic location disclosures, such as upon reaching or departing a specified location. Primary design concerns for this system were reliability, privacy issues, and deployment barriers. The system was meant to be compatible with hardware which users may currently carry with them, such as cellular phones.

3. Instant Hoot

The remainder of the paper is organized as follows. In the next section, we will discuss the current issues regarding text messaging. The remainder of Section 3 describes the Instant Hoot approach and the scenarios the system covers. In Section 4, we describe the system architecture. Lastly, in Section 5, we share results from our design iterations.

3.1 Exploring the Text Messaging Arena

We began the project by informally interviewing our friends and family to get a glimpse of how and if our contacts currently use text messaging and other modes of communication. We found that the majority of the people we asked avoided text messaging for two reasons: difficulty of use and cost. We decided to investigate this further by conducting an online survey to gather data about how cell phone users use (or why they don't use) text messaging on their phones. Most of our respondents use text messaging quite frequently, but it seems that the primary reason that people text is reflexive in response to receiving a text message (87.5%). Alternative modes of communication are very prevalent - 143 out of the 171 respondents said that they choose to call over text messaging, 121 choose to IM, and 99 email as an alternative to text messaging. We probed our participants to find out difficulties in text messaging and the benefits of alternative modes of communication, and once again, we found that the primary reason people did not use text messaging was cost, followed closely by text messaging taking too long. For survey questions and results, see Appendix E.

"[It's a] pain to type text on a cellular phone."

"I never find [template messages] useful enough and like to text message based on context."

Past solutions have tried to address this problem of inefficiency by shipping phones with a set of template messages that users can quickly send. In our survey, we found that 79.9% of our respondents had these template messages on their phones. However, only 17% of participants actually utilized this feature. This is due to a variety of reasons, the primary one being that users do not find template messages to be personalized enough to fit their needs. Some users found that the template messages were too formal whereas others found them to be too casual. In addition, users did not want to navigate through multiple menu levels to access templates. These findings led us to realize that there is no way to create templates that will fit all users - individuals have personalized needs.

"They're too proper…"

"The wording is too casual."

3.2 Our solution

In response to the findings from our studies, we designed Instant Hoot to make text messaging easier and faster by integrating contextual information and adapting the notion of template messages to fit individuals' needs. The system will have knowledge of events on the user's calendar along with the user's location and time. These factors will trigger the system to auto-suggest appropriate text messages to the user based on a set of specific scenarios. To solve the issue of impersonal template messages, the system suggests messages that the user has previously sent in the same context so that these "templates" will be appropriate for the scenario and specific to the user. We also added a "mad-libs formatting" feature which makes words within the message that are most likely to need editing active so that users can quickly modify these words while minimizing keystrokes needed to edit the message.

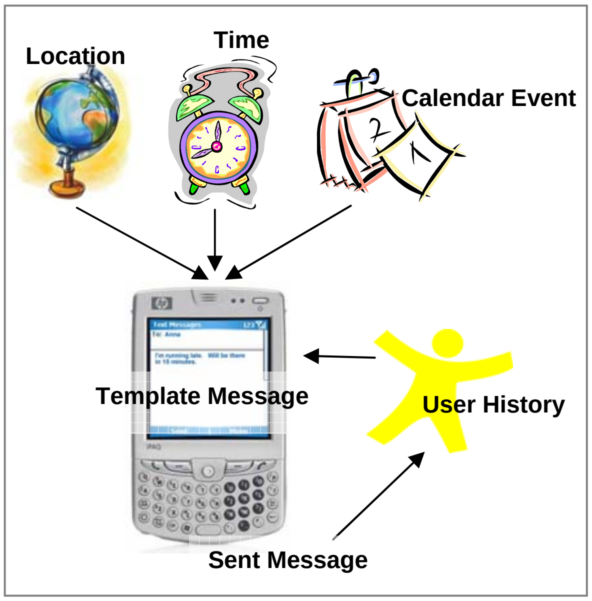
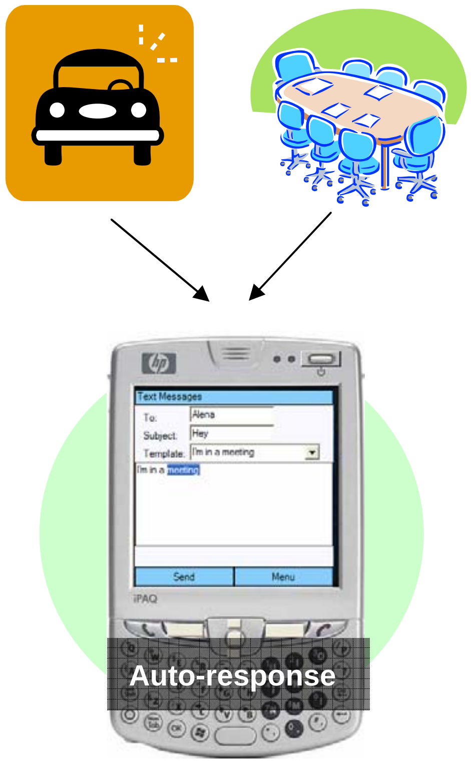


Figure 2: Overview of Instant Hoot. System takes into account location, time, and stored calendar events and provides user a list of template messages. The resulting sent message is saved according to the scenario

3.3 Instant Hoot Scenarios

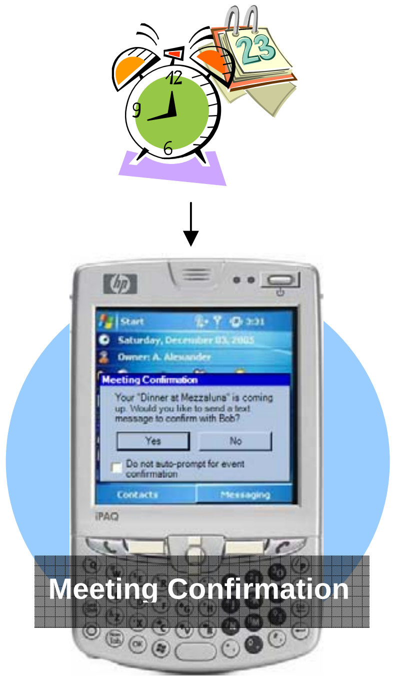
Our project supports four scenarios – Auto-response, Meeting Confirmation, Running Late, and Location Reporting. We will explain and address each of these scenarios below.



To: Alena
Subject: Hey
Template: I'm in a meeting
I'm in a meeting

Auto-response: Users in our survey mentioned that they often text in response to a received message, particularly when it is inappropriate for them to call. Two specific scenarios mentioned were attending a meeting and driving. Efficiency is crucial in these scenarios and users cannot spend time to type a message. Instant Hoot automates these replies for the user by detecting the correct scenario and suggesting appropriate responses. In the meeting scenario, the system will be triggered by the user's calendar events to prompt a reply that the user is in a meeting. In the driving scenario, when a user receives a text message, the system will detect the phone's motion and create a text message that informs the user's contact that they are driving. The first time either of these scenarios is encountered, we prompt the user with a draft of the text message, allowing them to make quick edits to the message that they want to send in this scenario. On the sent message confirmation screen, we offer the user the option of auto-responding with the message they just sent the next time this scenario is met. Research has shown that users are unlikely to change the default settings on most systems. Therefore we wanted to make it easy for the user to change key settings without having to navigate to the settings page to do so. In the auto-respond scenario, the default settings are to respond with a busy message when driving and when in a meeting. The user can view and edit the message before sending. The first time this scenario is encountered, however, the text message sent confirmation screen will display an option to send the message automatically the next time this scenario is encountered.

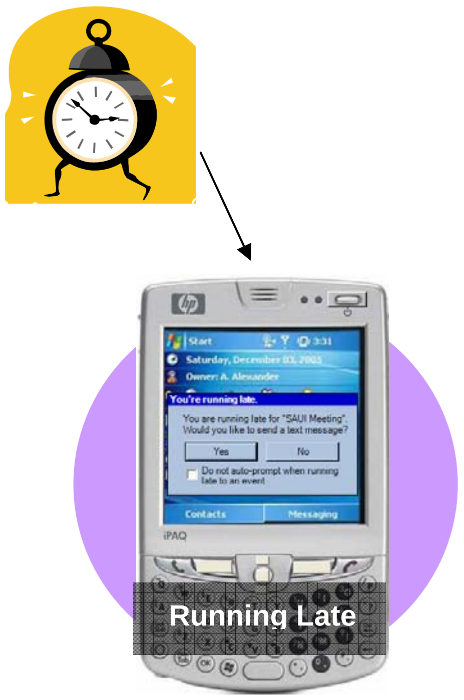
Meeting Confirmation: In our initial interviews, our online survey, and past research [2], we found that a popular reason for text messaging is status checks. Users often want to ensure that upcoming plans have not changed. Therefore, our system will prompt the user when a non-repeating event is approaching with the option of confirming the meeting/event with the other attendees. Users can customize their settings to determine how far in advance of an event they would like the system prompt. These settings can be integrated with users' current calendar reminder settings so that users do not have to enter similar information in two places. Editable settings for the meeting confirmation scenario include integrating this feature with the user's existing calendar reminder settings and how many minutes in advance they wish to be prompted for one-time events. As the default, integrating with existing calendar settings is disabled and the user is prompted 30 minutes prior to a one-time event.



Your "Dinner at Mezzaluna" is coming up. Would you like to send a text message to confirm with Bob?

Do not auto-prompt for event confirmation

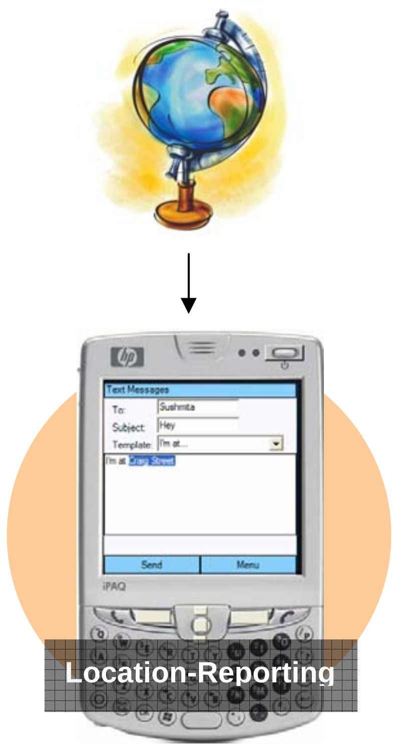
Running late: As we mentioned, there are certain scenarios where urgency is crucial and users do not have the time to spend typing out a text message. An important example of this is when users are running late for an event and want to inform other attendees. When an event is approaching, Instant Hoot will take into account the event location and event time, and compare this with the user's current location and time. If it appears that the user will not make it to the event's location by the event time, the system will prompt the user with a drafted text message that states that the user is running late. The drafted message will initially be a standard template message, but will adapt to include users' previously sent messages. Default settings for the running late scenario are to prompt the user 5 minutes before a calendar event to which the system senses they are not going to be on time. The user may edit how many minutes before or after an event has begun to be prompted by the system.



You are running late for "SAUI Meeting". Would you like to send a text message?

Do not auto-prompt when running late to an event

Location Reporting: Another scenario that we discovered from our initial interviews and our background research was when someone text messages to ask where a user is. This is a type of response that can be automated by a system. The wording does not really vary, but users can specify their location on the level of the neighborhood, street or calendar event. For example, writing “I’m in lab” may be more meaningful than saying “I’m at 300 S. Craig St.”. To handle this variation, the system will once again take into account user history but will store the extra dimension of user history according to the location. Thus, if a user sends “I’m at the lab” when they are at 300 S. Craig St, the system will auto-suggest this response the next time they are in the same location. Users are able to edit the location reporting scenario settings by specifying with what granularity the system should report their whereabouts. Users can choose between reporting their location by street, neighborhood or by calendar event location. The default setting here is to report at street level.



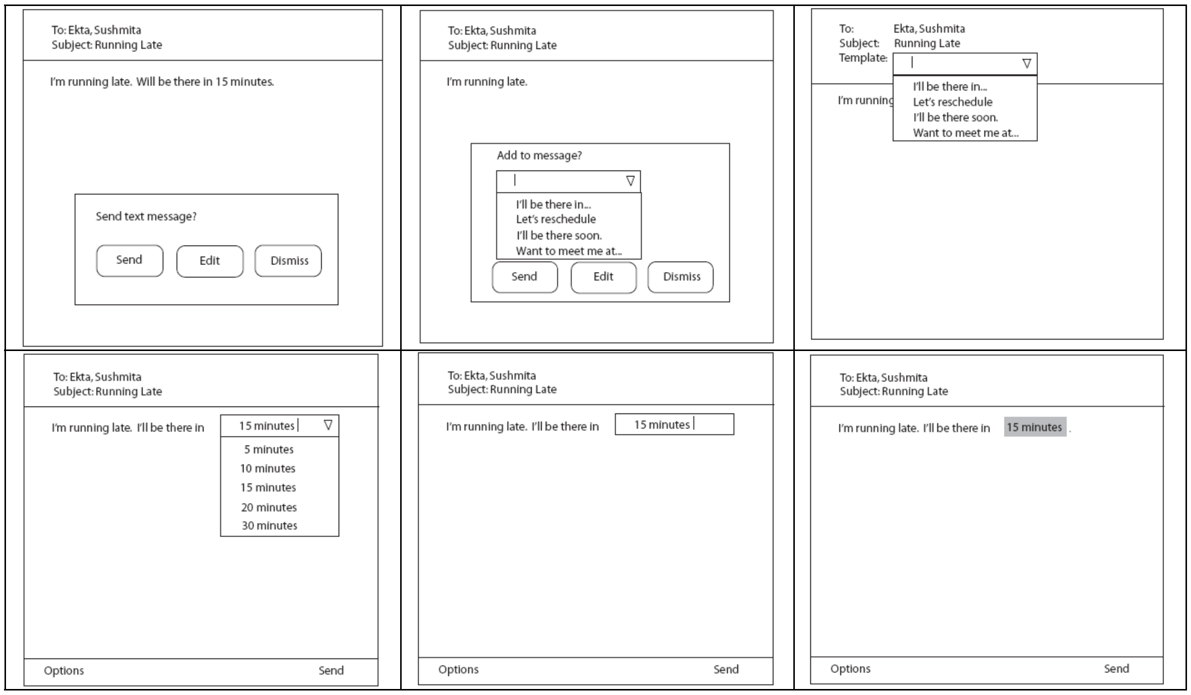
To: Sushmita
Subject: Hey
Template: I'm at...
I'm at Craig Street

3.4 Instant Hoot Architecture

We created mockups and storyboards of the user interface of our mobile device using Adobe Illustrator and Macromedia Fireworks. We then implemented a prototype of the system using Visual Studio developing in C#. We created a Windows Form application for each of the above scenarios that simulated the actions that would occur after the scenario was triggered.

3.5 Evaluation

We conducted a number of studies to evaluate and iterate on our designs. After we completed our initial mockups and storyboards, we performed informal think alouds with paper prototypes on seven people.


To: Ekta, Sushmita
Subject: Running Late
I'm running late. Will be there in 15 minutes.
Send text message?

To: Ekta, Sushmita
Subject: Running Late
I'm running late.
Add to message?

To: Ekta, Sushmita
Subject: Running Late
I'm running

To: Ekta, Sushmita
Subject: Running Late
I'm running late. I'll be there in 15 minutes

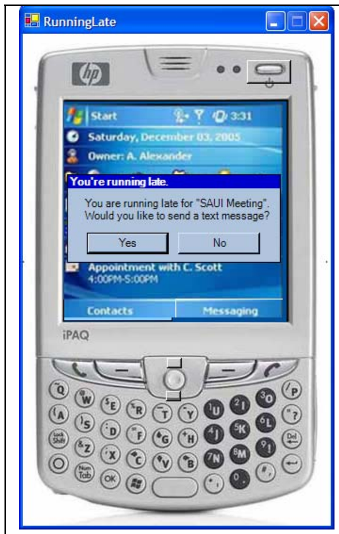
To: Ekta, Sushmita
Subject: Running Late
I'm running late. I'll be there in 15 minutes

To: Ekta, Sushmita
Subject: Running Late
I'm running late. I'll be there in 15 minutes .

Figure 3: Wireframes that outlined our different design ideas. The top row shows various possibilities for the user to select a template message both via a pop-up, and via a drop-down within the text message. The bottom row shows varieties of mad-libs formatting.

The study provided us with a few user preferences about controls and features, but its main benefit was to aid in fleshing out various design ideas and flows as we prepared for the think-alouds. (Figure 3)

We then implemented each of our design variations for one of our scenarios and performed a second think aloud on 10 people with the aim of finalizing one design flow out of our many variations. (Figure 4) Eight out of ten users preferred an embedded template selection over a popup editing window. Users not only provided us with a preferred design flow, but also exposed ambiguous wording in our designs and missing features, such as auto-saving drafts of text messages, auto-fading our confirmation page, as well as considering how mad-libs formatting can be added to new template messages that users modify.

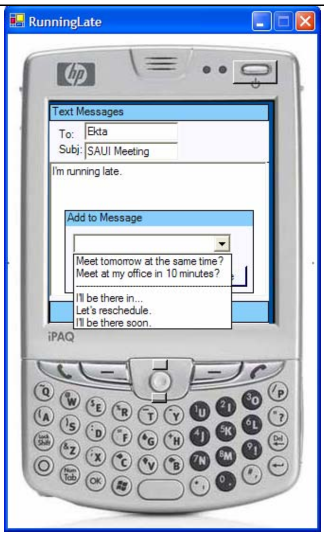


You're running late.

You are running late for "SAUI Meeting". Would you like to send a text message?

Yes No

Initial popup window that appears to inform user of scenario.

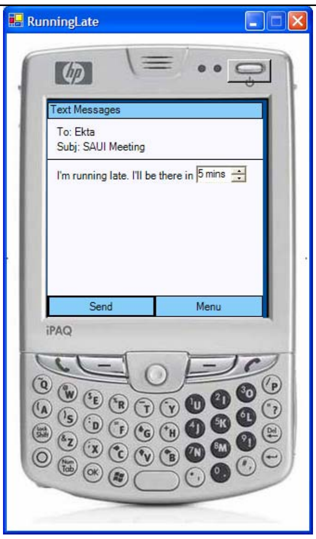


To: Ekta
Subj: SAUI Meeting

I'm running late.

Meet tomorrow at the same time?
Meet at my office in 10 minutes?
I'll be there in...
Let's reschedule.
I'll be there soon.

Message popup window that allows user to enter additional text.



To: Ekta
Subj: SAUI Meeting

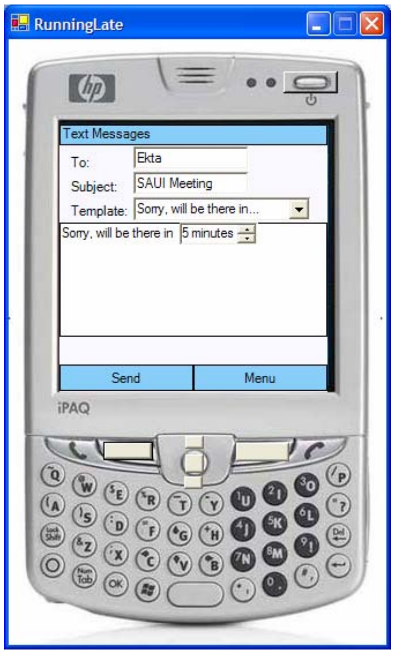
I'm running late. I'll be there in 5 mins

Scrollbox allows user to easily edit the minutes field.

Figure 4: Sequence of think-aloud scenario run-through. To see a full storyboard of the running late scenario, refer to Appendix B.

We modified our implemented scenario based on the feedback from this think aloud, and performed analytical evaluative methods on our design before expanding it to all the other scenarios. We assessed the task of confirming an event by performing a cognitive walkthrough of this scenario's prototype implementation. The task included responding to a system prompt asking whether the user would like to confirm an event with other participants, performing the text message editing process, and finally sending the message. The method confirmed that our system was successfully easy to learn and use and followed usability principles. The only problem found was that users may not notice the template selection dropdown, but we believe the in-message dropdown is still the best design choice, compared to alternatives of a pop-up (which was discounted in our think aloud) and the existing system in which templates are found within menus that involve searching through multiple layers. Therefore, we do want our template suggestions to be on the same page as the text message, and a dropdown control is the most feasible option given the limited space on a cell phone screen. The results of the cognitive walkthrough can be found in Appendix D.

We also performed a heuristic evaluation on the design and found that our system conformed to the criteria defined by this usability method. We found a few minor usability problems, but did not feel the alternative solutions justified changing our design. For example, the changes in the message body are not immediately reflected in the template dropdown (though the modified sent message is visible in the template dropdown the next time the scenario is encountered). We believe this is not a severe violation of system status because it may be confusing for the user if the template is updated each time the user edits the message body. The results of the heuristic evaluation can be found in Appendix C.



To: Ekta
Subject: SAUI Meeting
Template: Sorry, will be there in...
Sorry, will be there in 5 minutes

Figure 5: Final design: template dropdown within text message and scrollbox to control mad-minutes

We expanded our final design flow to all of the scenarios and performed a final think aloud study of six participants to evaluate the end result. While our implementation of Instant Hoot was not as usable on the computer as we would have liked, it was confirmed that it provided a quick and simple means for sending text messages for the simulated scenario. Users were happy to use a system that reduced the current pain of sending text messages and using templates. In addition, the idea of using user history to adapt templates to fit their text messaging style was well appreciated by our participants.

4. Discussion

We learned from Schilit et al's research [3] that there are a few factors that are important to consider in contextual technology: improving relevance, minimizing disruption, improving awareness, reducing overload, and selecting channels. We address the first four factors:

Improving relevance: This is the core concept in Instant Hoot – we integrate contextual information (time and location) and make decisions based on this information and the user's calendar of events. However, Instant Hoot does not attempt to improve relevance by filtering out information (messages) which are not relevant to the context, but by filtering *responses* appropriate to a given scenario.

Minimizing disruption: We provide the option of auto-responding in specific scenarios so that the user does not need to deal with replying when otherwise occupied. For example, when the user is in a meeting or driving, the system can auto-respond to any incoming text messages.

Improving awareness: We improve awareness in our system by informing the user that they are running late or detecting when they are driving to inform an incoming text sender that they are otherwise occupied.

Reducing overload: We provide a dropdown of relevant suggestions (messages that the user has previously sent in the same scenario) within the text message. We only include template messages that are relevant to the specific so that the user is not encumbered by a full list of messages that they do not need to see in the given context. We also add the mad-libs formatting for easy editing so users do not need to expend extra effort to get to words they are likely to modify.

There are a few other considerations to consider regarding a contextual messaging system:

What happens when two scenarios conflict?
Since our system only has four scenarios, we do not foresee many conflicts. The location response scenario has the most potential to conflict with other scenarios because it is triggered by an external contact rather than by context alone. A likely example of a potential conflict is if a user is running late and another attendee messages the user to ask where he or she is. These two scenarios are not likely to happen at the exact same instance, so one will occur first. If the running late scenario comes up first, and the user does not mention his/her location, the user's contact may still wish to text the user asking his or her location, in which case the system will respond to the message normally as a location reporting scenario. If the sequence of scenarios were reversed, on the other hand, and a user received a

"Where are you?" message before the "Running late" scenario was encountered, the system should take into account that the user has just responded to the attendee with their location and drop the running late prompt for this event.

Should a location response scenario arise around the same time as a meeting confirmation or auto-response scenario, the system will not treat this as a conflict. In the auto-response scenario, the system reacts to any incoming message in the same way. In the meeting confirmation scenario, we do not see these as overlapping scenarios, so we think that both should be triggered and responded to separately.

Can Instant Hoot automate intelligently according to context?

As an example, if a user is very far away from an event location or is moving in the other direction of an event location, perhaps the system should not trigger the event confirmation and running late scenarios. Additionally, if a user has just reached an event location, he/she should no longer need to reply to a "Where are you?" type of a text message. The system addresses this issue by keeping the user, the "human common sense" in the decision loop of each scenario. First, the user is able to trigger situations by confirming the system prompt or choosing to reply to or dismiss trigger messages. Second, the user is able to edit settings for each of the scenarios, customizing the system's response to their needs.

What happens when the system is wrong?

Should the system trigger a scenario incorrectly, the user may simply dismiss the prompt using the existing "clear" or "cancel" key on their phone. As discussed above, there exists little overlap between the four scenarios we have included in the system, and thus we anticipate few errors as the system processes triggers. We considered providing blank message option in the template selection drop-down but decided that this was not intuitive and users would more likely erase an entire suggested message instead of using this option.

5. Future work

Instant Hoot is still at an early stage of development, and there are several directions for further study. Our research suggestions for the immediate future are explained below.

In order to make Instant Hoot cover additional everyday scenarios, exploring further visualizations and interactions will be important. For example, it would be desirable for the system to automatically generate new scenarios based on messages the user receives. The system would need to take into account the context surrounding a given text message and define a new scenario based on this information. It will be a challenge to define and analyze which variables would make up a scenario so that the user's reply can be replicated in some meaningful way. The addition of new scenarios will also increase the complexity of the system and the likelihood of conflicting scenarios and overlapping triggers.

We currently have a setting that allows users to rely on their existing calendar settings for meeting confirmations. We would like Instant Hoot to integrate more naturally with the existing calendaring system on smart phones. This includes merging reminders generated by both systems, so that users can smoothly use both systems and not be bothered by multiple reminders.

In addition, in our analytical evaluative methods, we found that the drop down control to suggest template message may not be intuitive or visually prominent. Although the participants in our think aloud studies did not validate this concern, another possibility for bringing up template messages could be for the system to auto complete a template message the user has begun to type. It is worth doing user studies when a prototype has been implemented on the smart phone itself to determine which method users prefer.

The ideas in Instant Hoot can also be integrated beyond the text messaging arena to take into account other applications of smart phones, including to-do lists and email. For example, a user's notes or to-do list can serve as triggers for new scenarios, such that instead of prompting a text message, the system would prompt the user to take care of an item on his/her to-do list.

These ideas were beyond our scope for this project, but are worth considering to improve the value of the Instant Hoot system.

References

1. Tang, K.P. and J.I. Hong, *Using Current SMS and Mobile IM Practices to Inform Social Mobile Application Design.* 2006.
2. Tang, K.P., J.I. Hong, and D.P. Siewiorek, *Everyday Communication Breakdowns in Dual Career Families & Their Implications for Mobile Coordination Systems.*
3. Schilit, B.N., D.M. Hilbert, and J. Trevor, *Context-Aware Communication.* IEEE Wireless Communications, 2002(October).
4. Smith, I., et al., *Social Disclosure of Place: From Location Technology to Communication Practices.* 2005.

Appendix A: Meeting Criteria

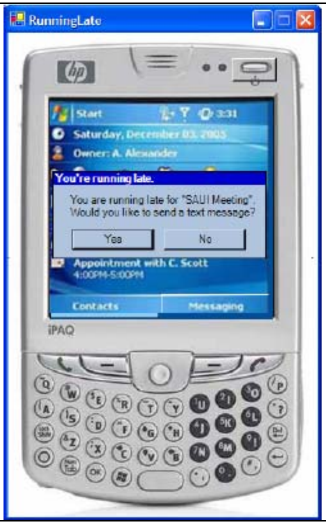
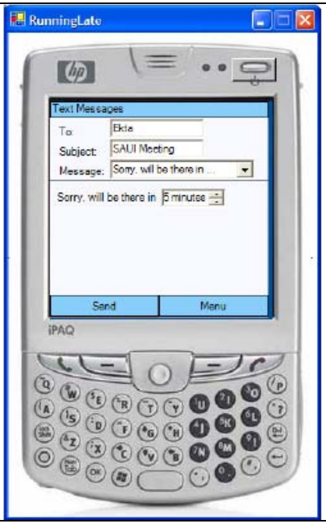
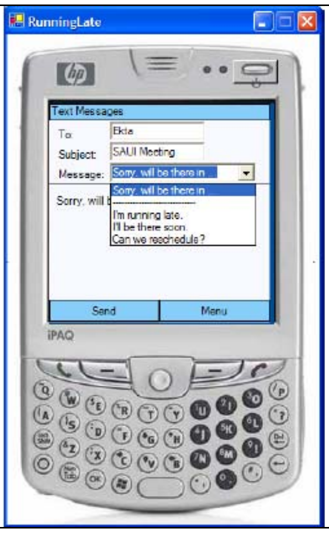
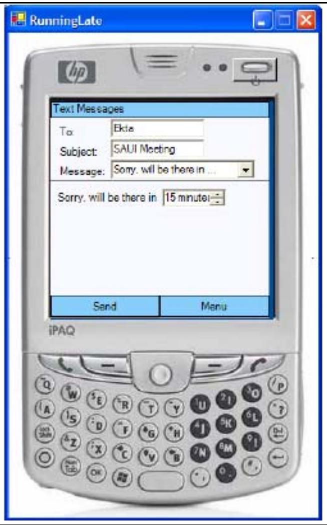
85% Criteria

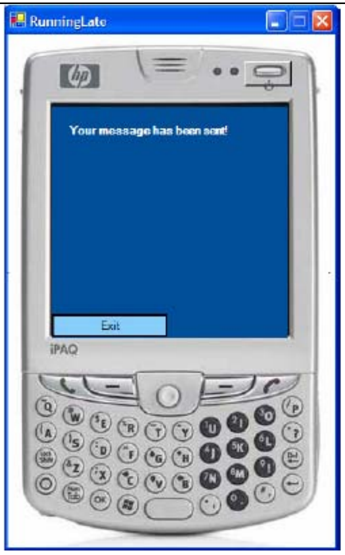
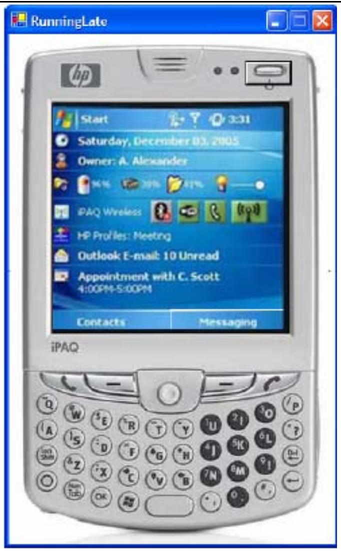
For our 85%, we successfully developed a set of scenarios to support contextual messaging. We researched and designed the level of automation that Instant Hoot would provide, and the amount of freedom we left to the user. We used analytical and empirical HCI methods to explore and evaluate the interface and user interaction designs. We implemented a working prototype in C# that allowed our testers to try the simulated system and allowed us to gather useful participant data about UI preferences.

Extras

Our user studies demonstrated that our users' wants and needs vary by individual. For our extras, we focused on personalization of the Instant Hoot system to cater to their individual needs. We designed and tested a text messaging settings page, and carefully thought out the default settings to best suit our smart phone users. We realize that users do not often navigate to the settings page, so we prompted the user about preferences in the context of our defined scenarios, rather than having the user hunt for the settings page. We also integrated the user's sent messages into the template messages instead of simply providing a set of blanket template messages intended to serve all users. Lastly, we added mad-libs formatting to several template text messages to allow users to easily modify fields that are likely to vary.

Appendix B: Storyboard

Joe is running late for a SAUI meeting with Ekta at 3pm, and it's already 2:55pm. His phone vibrates in his pocket, and he takes it out.	**He sees the popup message, and presses Yes. He is taken to a prepopulated text message.**
 You're running late. You are running late for "SAUI Meeting". Would you like to send a text message? Saturday, December 03, 2005 Owner: A. Alexander Appointment with C. Scott 4:00PM-5:00PM	 To: Ekta Subject: SAUI Meeting Message: Sorry, will be there in ... Sorry, will be there in 5 minutes
He browses through the possible template messages to find which one he wants to send to Ekta.	**He decides that he likes his wording the best, and sticks with the original message. He changes the time to be 15 minutes using the joystick on his smartphone.**
 To: Ekta Subject: SAUI Meeting Message: Sorry, will be there in Sorry, will be there in I'm running late. I'll be there soon. Can we reschedule?	 To: Ekta Subject: SAUI Meeting Message: Sorry, will be there in ... Sorry, will be there in 15 minutes

He sees a message that the message has been sent.	The sent confirmation message fades and he is taken to the home screen.
 RunningLate Your message has been sent! Exit iPAQ	 RunningLate Start 3:31 Saturday, December 03, 2005 Owner: A. Alexander iPAQ Wireless HP Profiles: Meeting Outlook E-mail: 10 Unread Appointment with C. Scott 4:00PM-5:00PM Contacts Messaging iPAQ

Appendix C: Heuristic Evaluation

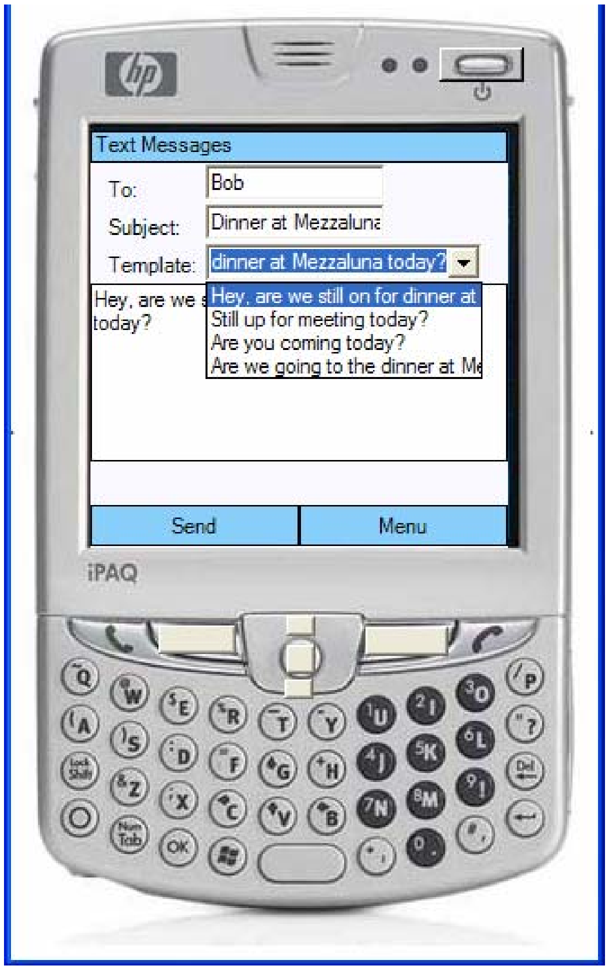
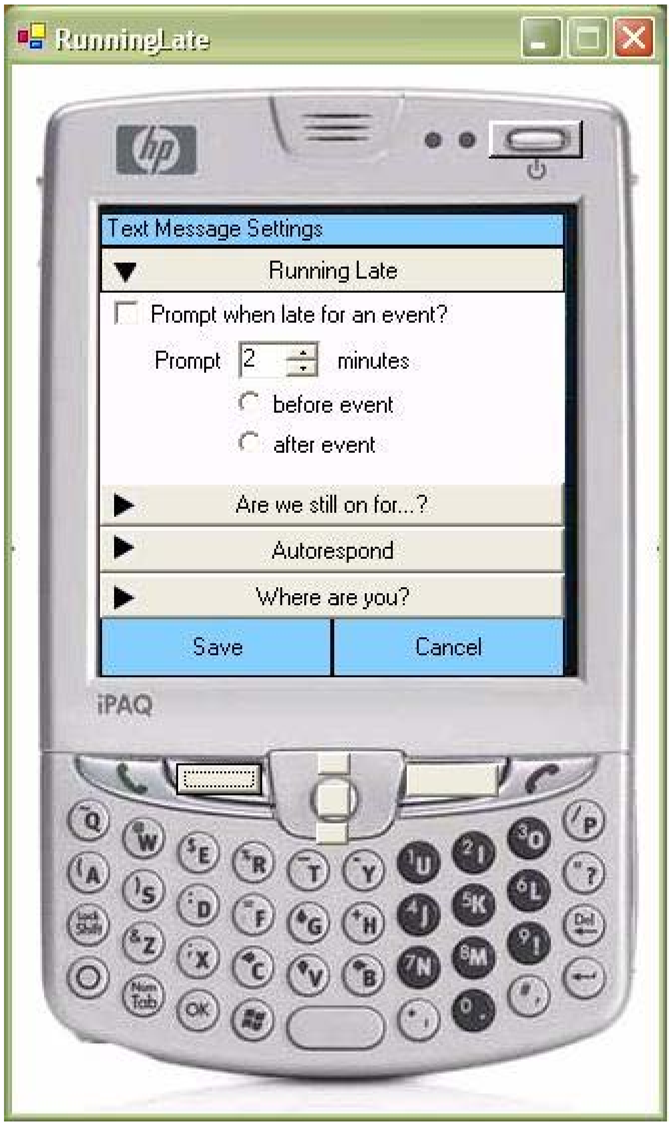

Text Messages
To:
Bob
Subject:
Dinner at Mezzaluna
Template:
dinner at Mezzaluna today?
Hey, are we still on for dinner at
Still up for meeting today?
Are you coming today?
Are we going to the dinner at Me
Hey, are we s
today?
Send
Menu
iPAQ

Table 1: Evaluating the new text messaging interface screen

Heuristic	Description of heuristic passing or violation	Justification for heuristic violations (if any)
Visibility of system status	Good that there is correlation between dropdown and message (if changes are made in template, they are reflected in the message.)	
Visibility of system status	Bad that changes in message body are not reflected in template dropdown.	OK b/c you don't want to add it if user doesn't want to end up sending. May be confusing if template is updated each time user types something b/c it seems like it would be a

		static thing. Also would slow system down.
Error Prevention	Good that we help users recognize, diagnose, and recover from errors. If users make a mistake, they can recover by choosing a template again to wipe out their mistake.	
Error prevention	Bad that we don't prompt user before sending.	This behavior is not in current systems and users do not seem to have a problem with this.
Match between system and real world	Bad: having templates dropdown in title area is a mismatch – not present in current systems.	But, this is actually an improvement than having users navigate thru menus. Also, this is a minor problem with low impact since users can ignore it if they want, low persistence b/c once they learn it they won't run into problems. It is high frequency though. Overall minor usability problem in which the pros outweigh the cons. The alternative is putting templates in menu, tradeoff: this involves too many layers.
Match between system and real world	Good: info appears in a logical order, bc template dropdown appears right before the message body to indicate that this is what it modifies rather than above the to/subject.	
Consistency & standards	Good: buttons at the bottom are consistent with current interface.	
User control & freedom	Bad: there is no undo.	Possible solution is to add an undo to the options in the menu, but this has a tradeoff of cluttering the menu further, and our user data

		shows that users do not like searching the menus for options.
User control & freedom	Good: saves draft in case user accidentally cancels without saving	
Aesthetic and minimalist design	Good: minimum things displayed on screen. Less clutter with dropdown.	
Flexibility/efficiency of use	Good: added madlibs formatting so users have quick access to words they are likely to change.	
Recognition rather than recall	Good: displays template in line with message so user doesn't have to navigate away to see it.	


RunningLate

Text Message Settings

▼ Running Late

☐ Prompt when late for an event?

Prompt 2 minutes

○ before event

○ after event

▶ Are we still on for...?

▶ Autorespond

▶ Where are you?

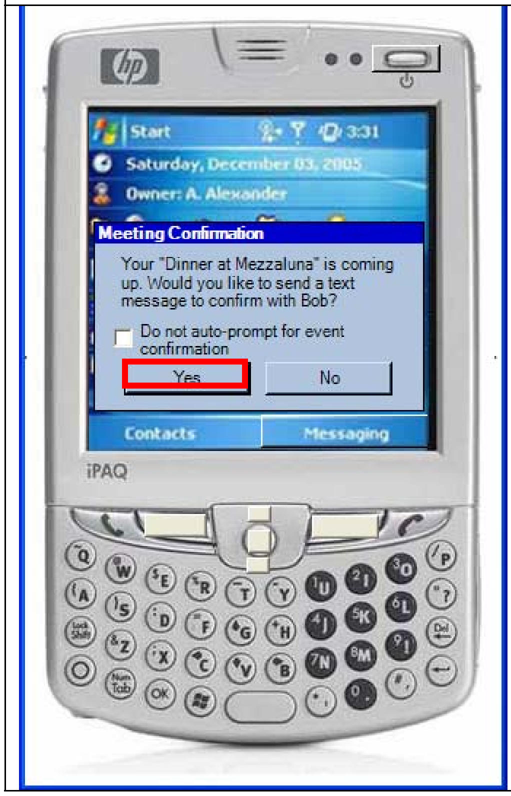
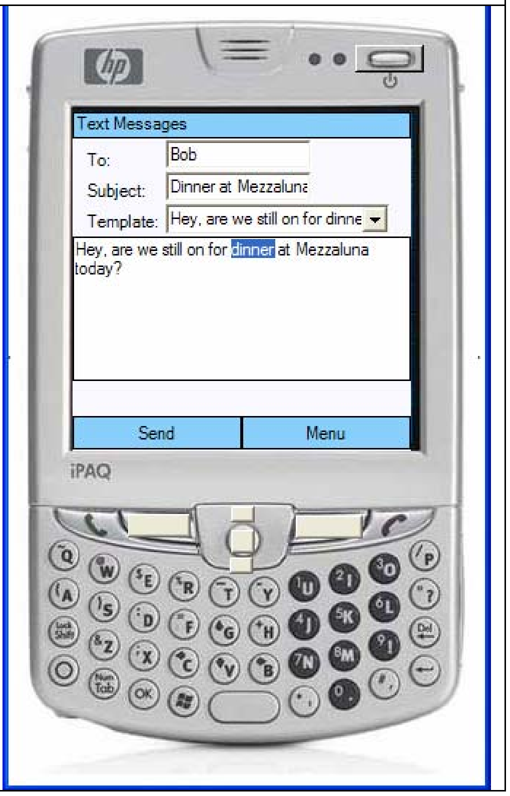
Save | Cancel

Table 2: Evaluating the settings screen

Heuristic	Description of heuristic passing or violation	Alternate Solution or Justification for heuristic violations (if any)
Visibility of system status	Good: radio buttons, checkboxes and controls get activated/deactivated so user knows what they're modifying.	
Visibility of system status	Good: use of triangles and depressed button show user which settings category they are modifying	
Aesthetic and Minimalist Design	Good: less clutter with expandable/contractible	

	setting	
Recognition vs. recall	Good: easy to see all the settings categories on one page so users know what settings exist rather than switching between diff layers.	
Match between system and real world	Bad: some labels may not be recognizable by novice users until they see the settings within the category. “are we still on for” may not be understandable.	Change from “Are we still on for…” to “meeting confirmation”
Consistency and standards	Good: buttons at the bottom are consistent with current interface	
User control and freedom	Good: can cancel if accidentally in settings page	
User control and freedom	Good: easy to change settings again after saving	
User control and freedom	Bad: don’t provide a way to revert to default settings	
Error prevention	Bad: Don’t prompt users before saving settings	
Flexibility and efficiency of use	Good: provide scrollbox controls when possible so users don’t need to type in minutes.	

Appendix D: Cognitive Walkthrough

State 01:		State 02:
	→	

Start 3:31
Saturday, December 03, 2005
Owner: A. Alexander
Meeting Confirmation
Your "Dinner at Mezzaluna" is coming up. Would you like to send a text message to confirm with Bob?
Do not auto-prompt for event confirmation

Text Messages
To: Bob
Subject: Dinner at Mezzaluna
Template: Hey, are we still on for dinne
Hey, are we still on for dinner at Mezzaluna today?

Step 1: Click Yes button

1. **Will the user try to achieve the right effect? (Goal)**

Yes, because users are familiar with popups and know they have to perform an action to make the popup disappear.

2. **Will the user notice that the correct action is available? (Perception)**

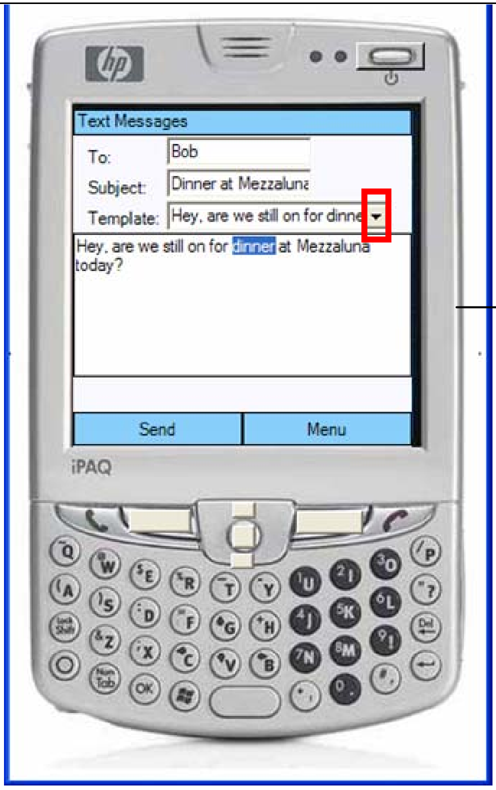
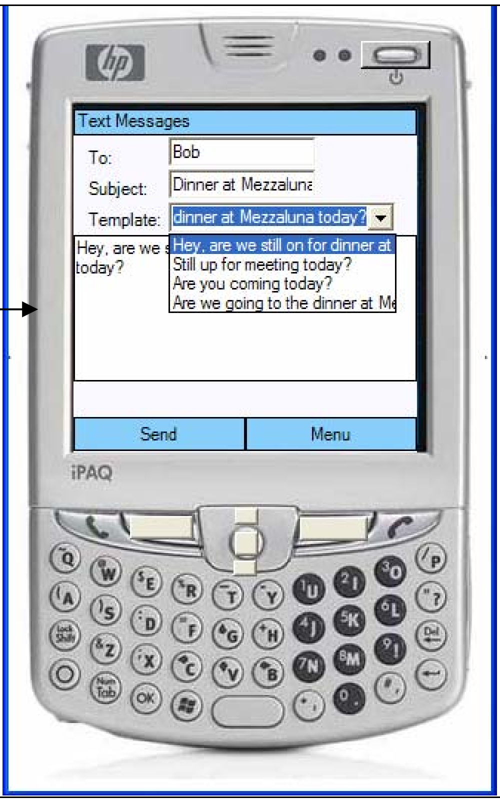
Yes, because the popup is brought into focus, and there are only 3 options on the popup, which makes the "Yes" button visible.

3. Will the user associate the correct action with the effect he or she is trying to achieve? (Comprehension)

Yes, because the popup displays a question that asks the user explicitly about whether they want to send a text message, stating who the message will be sent to and what the message is about. Users are given enough information to answer the question and know whether to press Yes or No.

4. If the correct action is performed, will the user see that progress is being made toward solution of the task? (Feedback)

Yes, because once they press “yes”, the screen will change to bring up a text message draft, and the popup will disappear.

State 02:		State 03:
		

Text Messages
To: Bob
Subject: Dinner at Mezzalune
Template: Hey, are we still on for dinne
Hey, are we still on for dinner at Mezzaluna today?

Text Messages
To: Bob
Subject: Dinner at Mezzalune
Template: dinner at Mezzaluna today?
Hey, are we s
today?
Hey, are we still on for dinner at
Still up for meeting today?
Are you coming today?
Are we going to the dinner at Me

Step 2: Change the template message

1. **Will the user try to achieve the right effect? (Goal)**

Yes, because users will see the currently displayed message and know whether they want to perform some action to edit this before sending.

2. **Will the user notice that the correct action is available? (Perception)**

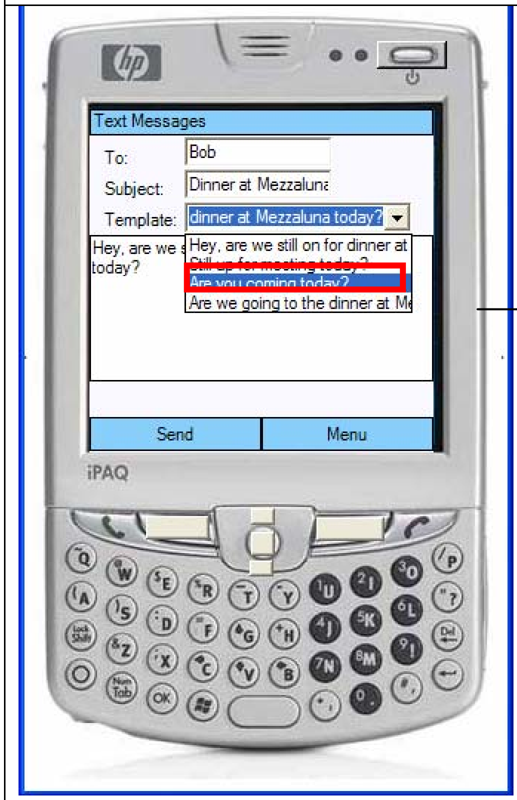
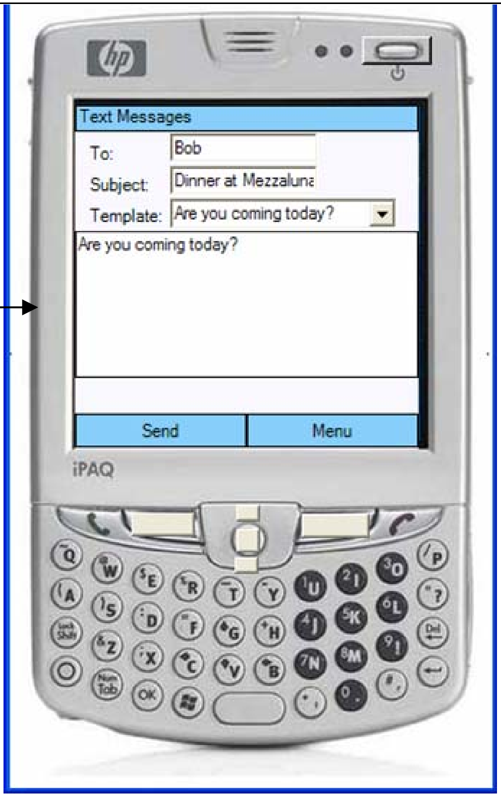
Yes, because the dropdown is visible on the screen in the heading area. Users might ignore the heading area since the To and the Subject are already populated, but they will at least perceive that something is there.

3. Will the user associate the correct action with the effect he or she is trying to achieve? (Comprehension)

Yes, because the text displayed in the dropdown reflects the text displayed in the message body, so users will realize the correlation between the two controls.

4. If the correct action is performed, will the user see that progress is being made toward solution of the task? (Feedback)

Yes, because once the user clicks on the dropdown arrow, a menu will dropdown that shows a list of possible templates.

State 03:		State 04:
		

State 03:

Text Messages
To: Bob
Subject: Dinner at Mezzalun
Template: dinner at Mezzaluna today?
Hey, are we s
today?
Hey, are we still on for dinner at
Still up for meeting today?
Are you coming today?
Are we going to the dinner at Me
Send | Menu

State 04:

Text Messages
To: Bob
Subject: Dinner at Mezzalun
Template: Are you coming today?
Are you coming today?
Send | Menu

Step 3: Click on an appropriate template.

1. **Will the user try to achieve the right effect? (Goal)**

Yes, because when looking at a dropdown, they will know that clicking on a dropdown list means that the selected item will change.

2. **Will the user notice that the correct action is available? (Perception)**

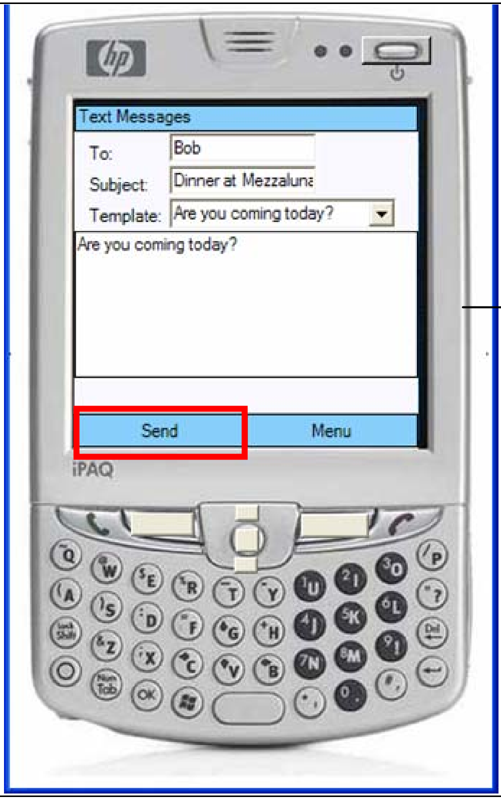
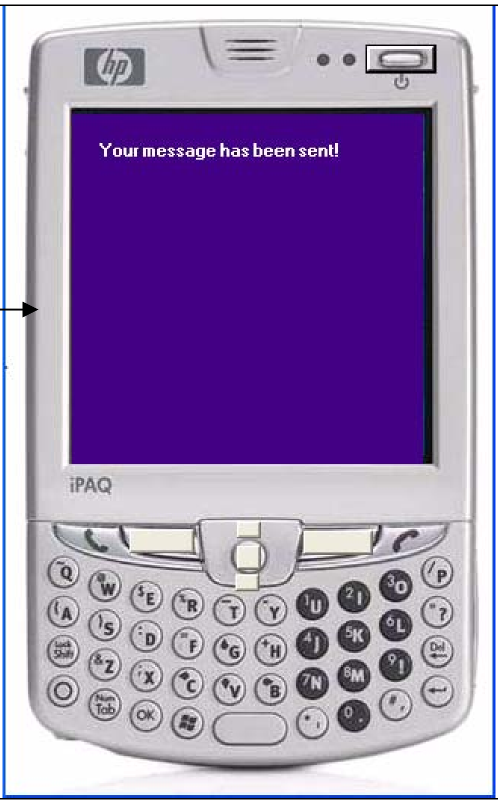
Yes, because the dropdown list is visible and they can see the template that they prefer in the list.

3. Will the user associate the correct action with the effect he or she is trying to achieve? (Comprehension)

Yes, because we assume users are familiar with dropdowns and know that you can click on different items to select them.

4. If the correct action is performed, will the user see that progress is being made toward solution of the task? (Feedback)

Yes, because both the message body and the selected template (that shows in the dropdown when it’s not expanded) will change to reflect the user’s selection.

State 04:		State 05:
 Text Messages To: Bob Subject: Dinner at Mezzalun Template: Are you coming today? Are you coming today? Send Menu	$\longrightarrow$	 Your message has been sent!

Step 4: Click Send button

1. **Will the user try to achieve the right effect? (Goal)**

Yes, because the current system has the Send button now, so users will be familiar with phone navigation. We did not change the send button functionality or placement.

2. **Will the user notice that the correct action is available? (Perception)**

Yes, because the Send button is visible and is placed where users are expecting to see it (in current systems).

3. Will the user associate the correct action with the effect he or she is trying to achieve? (Comprehension)

Yes, because users are expecting that the send button will send their text message based on experience with current systems.

4. If the correct action is performed, will the user see that progress is being made toward solution of the task? (Feedback)

Yes, because we provide a confirmation page that tells them that a message has been sent.

Appendix E: Online Survey

1. How often do you use text messaging? (select one)
    a. I'm addicted! (daily) – 54.5%
    b. Occasionally (few times a week) – 25%
    c. Rarely (few times a month) – 12.2%
    d. Never (once a year or less) – 8.3%
2. How else do you communicate with other people for quick communication?
    a. Phone – 86.6%
    b. IM – 73.2%
    c. Email – 61.7%
    d. Other – 12.1%
3. What are reasons that you choose other communication mediums over text messaging? Text messaging:
    a. Takes too long- 38.2 %
    b. Costs money – 49.3%
    c. Is not as personal – 22.2%
    d. Is hard to use – 13.2%
    e. Other – 22.9%
4. In what scenarios do you use text messaging?
    a. To tell ppl I'm running late (56.8%)
    b. To send reminders (58.9%)
    c. To schedule events (56.8%)
    d. To check in with someone (65.8%)
    e. To confirm planned event (64.4%)
    f. Sending a personal joke/story (54.1%)
    g. Replying to a text message (86.3%)
    h. Other – 14.4%
5. Who do you text message?
    a. Family - 65.1%
    b. Friends - 93.8%
    c. Work colleagues - 33.6%
    d. Classmates - 69.9%
    e. Other - 8.2%
6. Does your phone come with a set of standard messages you can send?
    a. Yes - 80%
    b. No - 11%
    c. I don't know - 9%
7. If so, do you use these standard template messages?
    a. Yes - 18.7%
    b. No - 81.4%