

Mobile Instant Messaging

a Product Design History

Missing out on SMS in Europe, Americans didn't have a compelling reason to use T9® Text Input to type on their mobile phone. But Instant Messaging, a quiet phenomenon growing as rapidly as the Internet, was being used by teens and professionals alike; could it become a “killer app” for mobile as well? Numerous technical and design challenges would have to be addressed to realize an acceptable user experience for Instant Messaging on mobile phones.

This article was written as a chapter for a book Scott Weiss proposed, *Case Studies in Mobile Design*, uniquely considering product design from the multiple perspectives of business, technology, and the user. The publisher never granted final approval for the book project, so the chapter is presented here in unabridged form.

Background and Business Context

Instant Messaging

Instant messaging is a type of Internet-based communications service that manages an exchange with another individual in order to communicate in real-time, analogous to a telephone conversation but using text-based rather than voice-based communication.¹ Participants typically create a screen name or alias which allows some anonymity. The instant messaging system supports the creation of a personal list of contacts,² similar to an address book, and indicates whenever somebody on that list is available to exchange messages. Unlike email and Short Message Service (SMS) text messaging, the conversation usually occurs only when both participants are logged into the system and available.

Most IM systems also host public or private “chat rooms” which allow a number of people to participate in one conversation, and provide privacy and security controls to avoid the kind of spam that has decreased the benefits of e-mail. The availability, or “presence” information, is distributed from central servers to all interested users, but in some IM systems the messages, pictures, and other content may be transferred directly between PC-based clients.

AOL’s Buddy List, AIM, & ICQ

AOL first added the Buddy List® feature to its proprietary dial-up client and allowed instant messages to be exchanged between AOL members who were online. Later, the AOL Instant Messenger® (AIM) desktop client offered the features in a free downloadable application, extending the AOL community to other Internet users. The AIM desktop client continued to reflect the competitive market by adding new features in regular releases, while AOL’s full



service client added features more conservatively to maintain its easy-to-use character. The AOL client version, however, allowed the user to restrict her receipt of messages to only other AOL members and offered parental controls to keep children safe from online predators.

AOL acquired the company responsible for ICQ and spent a few years adapting ICQ’s unique features to run on AOL’s large IM infrastructure. The most notable differences between the systems included: a unique numeric ID instead of a screen name, with a search feature to find ICQ users by name or email address; control over who can add a user to their contact list; the ability to send a message even when the recipient is offline; and a number of additional privacy options to restrict messages from unknown or unwelcome senders.

Other IM service providers have the same basic features, but slightly different benefits and restrictions. Other companies have created “multi-headed” clients (e.g., Trillian) to permit the user to exchange messages with users in different online communities from a single desktop application.

Commercial Opportunity

By 1999, the management at Tegic Communications was generally pleased with how things were going for T9 Text Input. Most of the leading handset manufacturers had at least licensed it, SMS

use in Europe and elsewhere was growing exponentially, and the new Chinese version of T9 gave Tegic a strong entry into that huge potential market.

But the US mobile industry was still a mess, transitioning its subscribers from analog to digital and dealing with many incompatibilities. Only a few wireless carriers³ supported SMS, and initially only within their networks. They competed on signal strength and voice plans, not data. Without text messaging, T9 was not even a “nice-to-have” feature.⁴

In the meanwhile AOL, Microsoft, and others were riding the wave of user growth as more Americans connected their home PCs to the Internet. AOL and ICQ members were already exchanging millions of instant messages; Yahoo! launched their own IM service; and Microsoft released MSN Messenger for Hotmail users.⁵

Here was the opportunity to accelerate mobile messaging and solidify T9’s value – connecting desktop IM to mobile SMS. In the US, where the carriers couldn’t interoperate, wireless IM could exploit the Internet to bridge network systems and connect desktop and mobile users. Since PC users in the US were already active users of instant messaging, wireless IM would let current desktop users “go mobile”. In Europe, where SMS already enjoyed wide popularity, wireless IM could offer the additional advantage of “awareness” or “presence”.⁶

Soon after Tegic’s launch announcement, Mobile Insights projected that wireless IM would be used by over 200 million people worldwide by 2002.

The carriers were Tegic’s primary customers for wireless IM, with the IM service providers as strategic partners. Tegic offered to provide a proxy server between the IM service provider and the carrier’s wireless data systems.⁷

Instead of settling for a strategic partnership, AOL acquired Tegic at the end of 1999 to help AOL enter the wireless market. As a result, the scope of the product plans narrowed to support and market AOL’s own instant messaging services (AIM and ICQ) rather than the service-independent proxy servers and clients that Tegic’s management had envisioned.⁸

Product Plan

The history of innovation is littered with the discarded pages of business plans which failed to account for the current (and continuously evolving) state of technology. The wireless industry was changing very rapidly, and in the US especially with a number of company mergers and technology transitions. System vendors were racing to establish their own technology as *the* standard for mobile applications, even if it wasn’t always ready for prime-time.

Within that context, Tegic leaders had chosen a phased approach for wireless IM, minimizing time-to-market while maximizing use of existing and expected handset and network capabilities:

1. Provide basic user awareness and message distribution, using the SMS infrastructure and an on-board SIM Toolkit applet⁹
2. Improve how users send and receive messages and view the status of other users, using WAP¹⁰
3. Deliver full IM capabilities, using built-in clients developed through partnerships with OEMs and portable downloadable (e.g., Java) clients
4. Reduce message delivery delays to make them more “instant”, using new technologies as they were deployed
5. Extend wireless instant messaging to support related applications.¹¹

As with most hi-tech start-ups, Tegic was led by technologists who may have never doubted that this plan would produce successful products!¹²

Following the product plan, Tegic's business development team and wireless technology experts proposed basic service requirements and a minimum set of features based on what they thought was possible and acceptable to the carriers. The service requirements included:

- Mobile-to-mobile and mobile-to-PC messaging
- Selective anonymity (via screen names)¹³
- Awareness of other users
- Basic setup via mobile phone
- Leveraging of existing wireless standards.

The minimum feature set fell into two categories, configuration¹⁴ and operation.¹⁵ The business team also addressed carriers' concerns about security and spam. It was no small thing for a carrier to open up its network to third-party servers and to millions of users who were not their customers.¹⁶

Product Design

Methodology

Tegic (and AOL) product management accepted the basic IM service requirements and feature set noted above, but it was likely that the mobile usage environment and technical constraints would change the importance of some desktop features.¹⁷ So I started over, using the Conceptual Design methodology¹⁸ to ensure that user needs were being served and not just propagating accumulated desktop IM features to a new platform.¹⁹

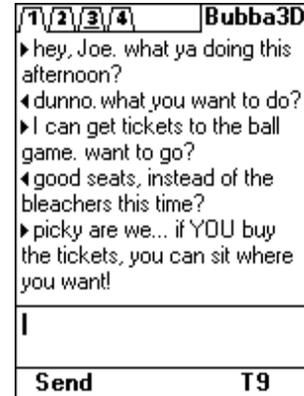
Though lacking the formal training of an ethnographer, I began by exploring and using instant messaging myself, asking questions of other regular users, and observing my teen daughters juggling 3-5 IM conversations at a time.²⁰

The results of the Conceptual Design process included the following artifacts and considerations that guided future design efforts:

- User profiles (further developed into a set of personas²¹ in the Target Audience, following)
- Usage environment (including social/geographical relationships between users)
- User requirements²² (including relative priority of IMs, dealing with unknown users, and disclosing availability)
- Task model (workflows of key activities)
- Usability objectives (later developing into Usability Goals, below)
- Constraints (the myriad technical limitations including message delivery delays and storage, device capabilities, and cross-platform inconsistencies)
- and a basic UI model.

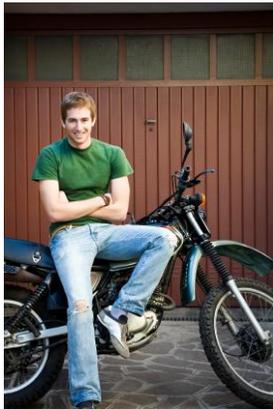
For Mobile IM these considerations suggested that the UI model should employ a list-plus-details paradigm²³ to manage the contacts and conversations, a split-screen conversation display to allow concurrent reading and writing, and contextual menus to improve efficiency. The Buddy List feature needed to be ordered by activity (rather than divided into buddy groups as on the desktop) so that the most recent conversations and/or contacts were visible at the top of the list. Some UI elements like tabbed conversations would be optional, and support for chat rooms could be deferred until later.

In addition, I theorized that some existing features would be needed for new reasons,²⁴ but given the dynamics of the mobile environment there was no way to know for sure until people started using them.



Target Audience

The personas representing typical users included a young mother, a teen girl, a mobile professional salesman, and a college guy.²⁵ These personas, however, were never integrated into the product development process.²⁶



Usage scenarios experienced by these personas included both their initial service discovery or setup situation and a daily-use scenario. Completing this exercise helped identify the challenges facing users who needed to gather enough information to use a wireless service for the first time.

Interesting dichotomies also surfaced: the primary use was personal (for informal messaging) rather than business (where email and BlackBerry continued to reign); the users were presumed to be AOL members, but more of them were not (AIM and ICQ users formed a larger audience); and teens formed a large target demographic for Mobile IM, yet they were under-represented in AOL's usability research (due to parental consent complications).

Usability Goals

I defined a few objectives (represented by the terms “Functional”, “Intuitive”, and “Fun”²⁷) to ensure that key aspects of the desktop experience transferred to the mobile environment. Eventually, these objectives became the “Five factors that make an acceptable Mobile IM user experience”:

Consistent – internally and/or externally

Easy – e.g., minimizing steps for the most frequent tasks

Interactive – calling for dynamic displays

Conversational – focused on the message exchange

Controllable – including privacy, security, and user preferences.

Usability goals need objective measures and thresholds to make them actionable. They also need the support of executives outside of the design team to enforce the consequences of an occasional failing grade. We did not always obtain that support, but as noted below, our attempts to establish a minimum acceptable user experience for Mobile IM were moderately successful.

Improvement Cycle

Even in the typical hi-tech organization where “time-to-market is everything”, iterative design is possible; it doesn’t always happen contiguously, however. As there was rarely time for user input and design evaluation before development commenced, we tried to apply what we learned as we completed one platform and moved on to the next.

The Mobile IM design improvement cycle might have been described as the following:

- Design** – Determine the constraints of the particular platform and the best way to offer the minimum feature set; create exploratory prototypes when possible; and complete a UI spec, illustrating desired outcomes and identifying key behaviors (from the user’s perspective).
- Build** – Help design workarounds to unexpected constraints, and update the UI spec as needed; define key service metrics and advocate instrumenting the client and/or server to track them; and work with the Technical Communications team to document issues that users are likely to face.
- Observe** – Determine the actionable usability problems and concerns to be tested by users; run small, quick studies; ²⁸ inspect the available service metrics; lobby for usage questions in later marketing surveys; and contrive new usability study techniques along the way. ²⁹
- Repeat** – Advocate for more than just bug fixes in the next release; merge lessons learned into the design for the next platform; and/or *delegate* – handing off responsibility for each platform to another designer introduced a new perspective, allowing old assumptions to be questioned, more usability issues to be identified, and the overall design to be improved. ³⁰

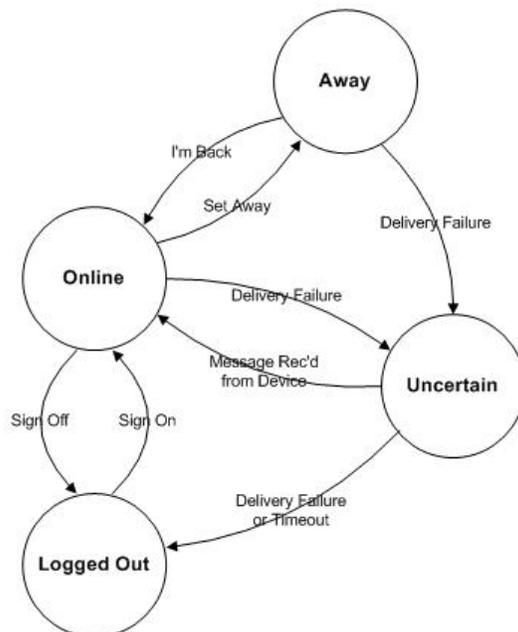
Every platform eventually benefited from design improvements in at least one subsequent major release.

Design Challenges and Compromises

Since we were often developing on the “bleeding edge of technology”, some important element of the client platform or the wireless infrastructure inevitably failed to match its promised capability until some time after the new product was launched.

A measurable amount of the design effort became re-design effort; that is, looking for workarounds and trying to keep the overall user experience acceptable. Significant constraints and design challenges included:

- Capability** – Protocols for “push” notifications ³¹ were implemented on handsets inconsistently; the typical mobile operating system (OS) was not hospitable to background applets; ³² lists and other basic UI elements often provided only rudimentary attributes; claiming space on the status line was a constant battle; and, of course, the softkeys ³³ never followed any standard.
- Reliability** – SMS delivery was not usually guaranteed because the system overhead for delivery confirmation was prohibitive. We proposed an “Uncertain” user status to represent the vagaries of wireless service, but



it needed live data from the wireless infrastructure – also prohibitive. Showing a mobile icon on the desktop IM client next to the contact’s name became the expedient means to caution the desktop user not to expect an immediate response.³⁴

Consistency – A constant challenge, especially regarding appearance and terminology.³⁵ We maintained a bias for consistency *within* rather than *between* devices. It remained unclear which Mobile IM elements were important to keep consistent with the desktop version, and whether that consistency mattered at all to the emerging mobile-only user.

Infrastructure inertia – The existing deployments of the instant messaging and wireless data infrastructures made it possible to roll out Mobile IM quickly and broadly. But a number of desirable features, e.g., the “Uncertain” status or simplified Privacy settings, were not practical if even the smallest change to these large complicated systems was required.

Costs – Though sometimes not considered by designers, in this case the billing model had a significant impact on the perceived value and the optimal use of each platform. The wireless carriers’ billing systems were often unable to support different charges for the “overhead” of login data vs. the “content” of sent and received messages. The carriers’ data plans and conditions of use were poorly understood by most of their subscribers.

In light of the phased approach and subsequent to the Conceptual Design process, we identified three or four different versions of the Mobile IM client that might be implemented.³⁶ The variations were due to differences in the ability of the mobile device platforms to support the same characteristics of IM that made the desktop experience effective and enjoyable.

Training and Help

Another challenge with advanced mobile applications was that most devices were not able to support much in the way of instruction. Severe limits on both storage and display space made the typical online training and help offerings ineffective. The 100-plus-page printed owner’s manuals were already packed with details, and reportedly were rarely read or referenced by users anyway. The economics of a new unproven service like Mobile IM discouraged the distribution of in-the-box informational flyers.

AOL’s [MyMobile](#) Web site became the fallback informational source for all of the mobile services. There another tradeoff soon emerged: whether to maintain very general descriptions and FAQs which didn’t directly apply to any handset on the market, or to create and revise phone-specific guides for a quickly-expanding collection of handset models from multiple manufacturers on multiple wireless networks.³⁷

The availability of information about a product can impact its success – whether via ads that set expectations or quick-start guides that help users over initial hurdles. “Technical and end-user communications should not be left until the last minute” was an often-repeated (but too-often-ignored) refrain by AOL Mobile’s technical and marketing communications team. They wanted the opportunity to review each proposed product, ideally by the time development began, to ensure that it could be explained to the customer through an effective (and cost-effective) channel.

Projects and Platforms

The carriers' push to monetize their new wireless data capability drove the Mobile IM product plan, which followed the phased approach surprisingly closely. Each carrier seemed to focus on a different platform for first deployment. The design team tried to apply the design improvement cycle as we progressed from project to project.

AOL Mobile eventually attempted or delivered at least one version of the Mobile IM service for each of the platforms below. How well did they come out? In describing and evaluating the resulting clients and/or services, I have considered them in terms of one or more of the following:

Context and/or Considerations – The unique situation created by the technology or carrier relationship that the product design had to account for.

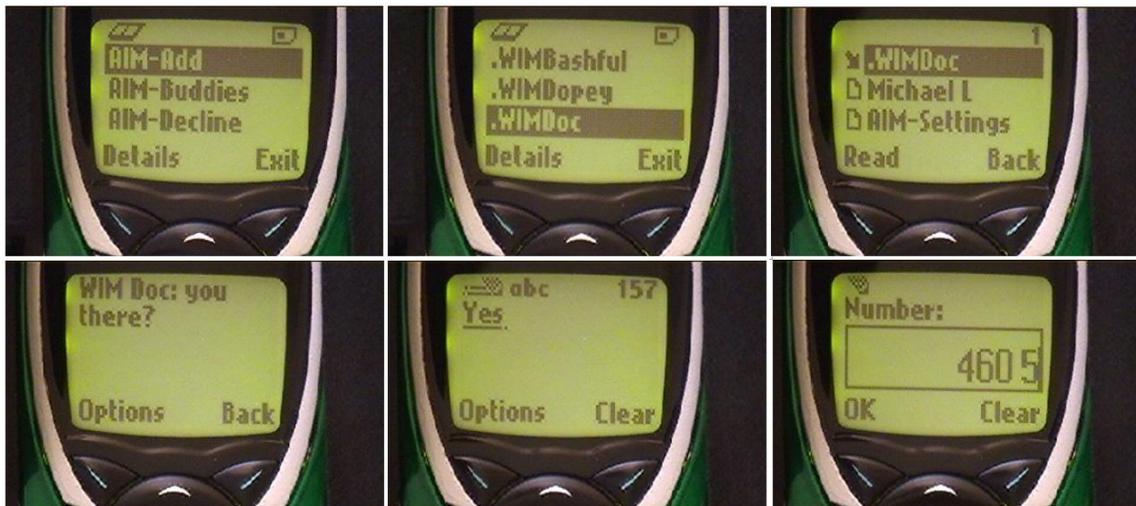
UI – Elements of the user interface that were particularly challenging or otherwise notable.

Usability – Key studies for the platform which confirmed the design or offered guidance for future revisions.

Successes and/or Surprises – What worked, what didn't, what was unexpected.

SMS

Millions of GSM³⁸ handsets across the US and Europe were already able to exchange text messages, a compelling installed base to target first. Tegic's proof-of-concept IM system³⁹ had already extended the Internet to SMS. So AOL Mobile engineered a scalable version for commercial deployment.



The product team assumed that the entire minimum feature set for this “Functional” product was needed by the savvy SMS user.⁴⁰ Each contact's screen name became a Phonebook entry with a unique short code⁴¹ so that a message could be addressed like any other SMS, and arriving messages were thus tagged with the sender's name.⁴² Additionally, each IM feature or command received a Phonebook entry (e.g., “AIM-SignOn”) to which the user would send the command's parameters in a text message, such as his screenname and password. Messages from another user could be simply forwarded to the Phonebook entry of the command (e.g., “AIM-Divide”) that you wanted to invoke on the other user's screen name.⁴³

Unfortunately, we soon discovered that none of the other carriers could fully support remote programming of the Phonebook; the process either didn't work on all phones or was too onerous.⁴⁴ For the carrier who was still willing to try offering Mobile IM via SMS, the user

might see a table of commands, contact names, and short codes on the carrier's Web site and be instructed to add them to the Phonebook as needed.

Even after configuration was taken care of, the number of operational steps needed to carry on an IM conversation via the text message Inbox made it tiring rather than "Fun". For example, a single message receive-and-reply might require these steps:

- Menu key / Messages / Inbox / select and read message
- Options / Reply / type response message
- Options / Send / confirm number
- Options / Delete / confirm delete (to prevent the limited Inbox from filling up).⁴⁵

We ran a couple of usability studies to ensure that the instructions on the carrier Web site would teach users how to configure their phones and exchange instant messages via SMS. We gathered user feedback on each major release via online surveys, usually during Beta programs or carrier "soft launches".⁴⁶ Some carriers also provided summaries of customer service calls that were related to instant messaging, though sometimes it was a concerned call from an executive who had an unexpected or unwelcome experience.⁴⁷

In later deployments, we reduced the number of commands that the user had to learn. For example, when a project extended IM for SMS to deliver picture messages,⁴⁸ system messages sent to the phone explained how to access and reply to the picture messages, rather than depending on new commands in the Phonebook. Hindsight also suggested letting the user explicitly add a few of his closest buddies to the mobile contact list.⁴⁹

Eventually the entire feature set was concealed when IM for SMS became two simple services:

IM Forwarding – Automatically rerouted messages from the desktop to the phone when the user stepped away from the PC

IM2SMS – Allowed sending messages from the desktop to any mobile phone number.

These services focused on convenience for the desktop user rather than full IM capability for the mobile user.⁵⁰

SIM Toolkit, Java

SIM Toolkit is a package that allows helper applets to be programmed into the SIM⁵¹ of a GSM phone. Most of the millions of GSM phones that could exchange text messages could also host a helper applet. The hope was that the multiple steps of view, reply, and delete could be reduced by a custom applet that the carrier would load onto all new customer SIMs.



But in spite of our best efforts to build a prototype, a Mobile IM applet was not feasible within the size and functionality constraints imposed by SIM Toolkit. The handset manufacturers also offered only minimal and inconsistent support for the runtime environment; for example, T9 Text Input was not enabled in SIM Toolkit entry fields on handsets of the time, effectively defeating the "instant" nature of the service.

The initial versions of Java on mobile phones also imposed too many limitations for an effective user experience or to reach its promise of portability. Though great for self-contained games, its "sandbox" model was almost incompatible with server-based instant messaging: the applet shut down when the user tried to use the phone for anything else, and startup and authentication was too slow; network connections automatically timed out and could only be initiated from the client side, so the user might not realize that a new IM was waiting on the server; and some handsets

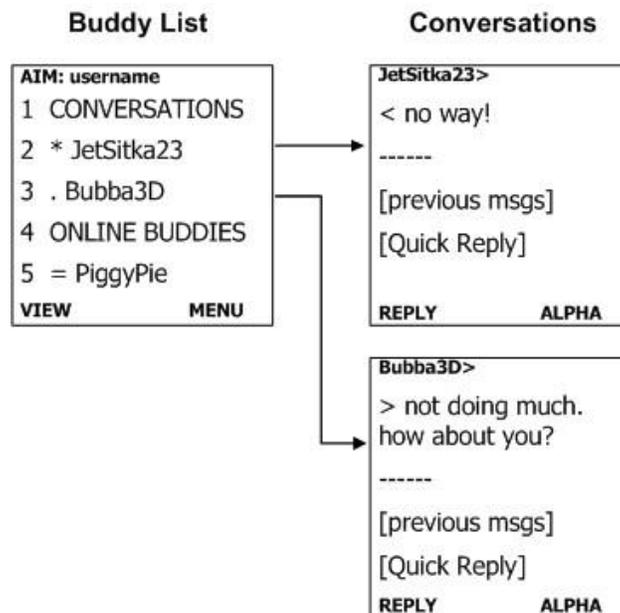
guarded against viruses by having the user confirm every SMS Send requested by the applet, multiplying the number of keypresses required to send a message. As a result, only one Java-based IM client was available prior to 2004.

Wireless Web

It was the end of the Millennium, and the Wireless Web was going to be the Next Big Thing. While our built-in IM client had to wait for manufacturers to integrate it into their handsets and ship them, WAP⁵² offered a server-only solution that promised to be as easy to deploy as any other Web site.⁵³

But in spite of the industry hype about WAP, it was immediately obvious to us that WAP was not ready for prime time. There was too much variation in implementations of the browser and related functions, especially how “push” notifications were handled.⁵⁴ And, like SIM Toolkit and early Java, there was not enough control over the entire user experience.

SprintPCS, for one, tried to realize WAP’s potential for its users, imposing some UI consistency across devices and creating an application style guide and approval process.⁵⁵ Inevitably, though, we had to use a lowest-common-denominator approach to deal with the variation in browser implementations. We iterated the Mobile IM design until the main functionality worked sufficiently on the first half-dozen handsets targeted by each carrier.



Our first usability study, run after development but before release on SprintPCS, evaluated whether users could complete the basic IM tasks. The study participants identified some issues with terminology, iconography, and navigation.⁵⁶

Fortunately, the server architecture was designed to handle frequent iterations and workarounds for individual handset problems, so we were able to address the most significant problems and re-test with users before the first public release. The single, sectioned CONVERSATIONS / ONLINE / OFFLINE list made its debut.⁵⁷ This improved design greatly simplified navigation with WAP and made it much clearer where to find new messages and which contacts were online, offline, or not listed at all.

But the system was barely revised after initial release. A year or so later, after net growth slowed, product management raised the question of whether the user interface – the one element over

which we had some control – was a significant factor in customers deciding to stop using the service after a while.⁵⁸ An online survey of current and recent AIM for WAP users provided the answer: Yes, along with text input speed, network reliability, and cost/billing issues.

According to the survey, users of AIM for WAP considered the service a fun way to quickly exchange text messages with anyone from anywhere. But many used it like a dial-up service, logging in just long enough to see who was online and to exchange a few IMs, apparently under the mistaken impression that their airtime minutes were being used as long as they were logged in.

Problems with text input was no surprise; T9 was poorly integrated with WAP browsers on many handsets and the few “quick reply” messages were not customizable. The survey indicated that we should try to reduce the frequency and intrusiveness of IM notifications, simplify sending/receiving messages and navigation between conversations, and offer user-customizable quick replies.

We were given the go-ahead for a redesign to improve what we could. The redesign capitalized on the cross-device consistency, new features, and visual appeal of the WAP 2.0 specification, which was now based on XHTML and supported by a new generation of color-screen phones.

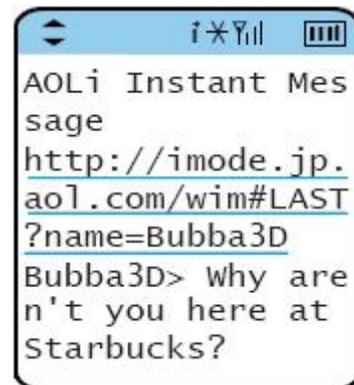
A follow-on user study confirmed the redesign’s improvements to the areas of concern and to the overall user experience, while identifying the side-effects of subordinating a few other features.

i-mode

In the meanwhile, the Japanese wireless carrier DoCoMo had built its own version of the Wireless Web called i-mode. It defined strict browser behavior and UI standards and offered a billing service so third-party developers could easily get paid. DoCoMo and AOL launched a joint venture to supply i-mode users with AOL email, IM, and other content.

But the only “push” notification available in i-mode was the “short mail” service. Employing it would have provided an ungainly user experience:

- Receiving a text message in the Inbox containing a long cryptic link to the IM conversation and a preview of the IM
- Following the link to the IM conversation displayed by the i-mode browser to be able to view and reply to the message
- Remembering to return to the Inbox to delete the notification.



Though I designed a complete IM solution within i-mode guidelines, DoCoMo AOL wisely chose not to develop and deploy it.

Built-in Client

PDA's were the first mobile devices to benefit from an open platform OS and publicly-available development tools. A team in the AOL Devices group designed and built AIM clients for Palm and PocketPC which mirrored the desktop client’s look and feel. Those PDA's had poor wireless connectivity, however, relying on dial-up modems over open-circuit connections; that defeated the desired IM-anywhere-anytime experience.

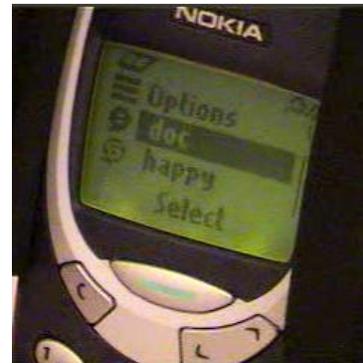
In a cooperative venture with RIM, the AOL Devices team also implemented custom e-mail and IM clients on a custom BlackBerry called the AOL Mobile Communicator. Though the software

was appropriately designed for the task, the pager-and-belt-clip styling didn't appeal to the young target market, and the economics⁵⁹ were ultimately unworkable.

A built-in, custom-designed application always holds the potential for the best user experience and platform integration; and, given the limitations of the other platforms, it offered the best opportunity to replicate instant messaging's conversational user interface (where, as often as not, new messages arrive and are viewed just in time for the user to change the reply she is currently composing). Unfortunately, each mobile phone manufacturer has had its own operating system and development tools. Tegic had dealt with that before, earning the trust of its partners through the delivery and support of a solid T9 product line and knowing how to create a small client core that could be integrated by a handset team. A Mobile IM client, however, presented new challenges as many more parts of the handset needed modification, and integration and testing among the manufacturer, wireless carrier, and service provider required extreme coordination skills. Instant messaging was also outside of the expertise, and personal experience, of many in the wireless industry.⁶⁰ As a result, we had to educate every new group we worked with, from product management to engineering to marketing communications, about the basics of IM and the expected mobile experience to ensure that it didn't get treated just like SMS.

When we started developing the built-in client, the next-generation high-speed wireless data networks had not been deployed. So the Mobile IM client-server protocol was tuned for transport via SMS with a future upgrade path to IP connectivity. Maintaining efficiency over SMS eliminated some desirable features, most notably near-real-time contact status updates.⁶¹

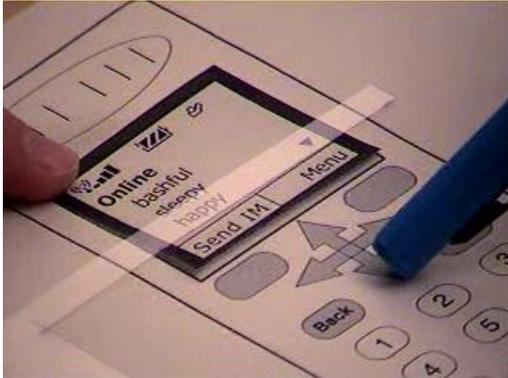
The first built-in client on the market implemented the UI specification straight from the original Conceptual Design proposal and didn't include the benefit of lessons learned from developing and testing AIM for WAP. So we ran a quick usability test to ensure that the typical AIM user could accomplish the basic tasks with minimal instruction or discovery time while avoiding major problems and that the overall user experience was acceptable. A post-test questionnaire confirmed that most participants would use that version of IM if it was on their phone and would also recommend it to their friend(s). Participants also made positive casual remarks about the handset and asked about its availability with their carrier. But although no "show-stoppers" were discovered, a number of suspected usability problems were confirmed during the test. Unfortunately, at that point, late in the development cycle, few of the problems could be fixed before release.⁶²



Those experiences prompted us to run a series of small studies examining the mental models and expectations of desktop AIM users approaching Mobile IM for the first time and the usability of key design elements.⁶³ Because the new design work had not yet been implemented on a handset, or even a PC simulator, we used paper prototypes to walk the study participants through the design. In many respects, paper prototyping helped keep our focus on issues that we could control⁶⁴ and ignore the issues that would differ for each handset anyway.⁶⁵

The results of the first paper prototype study indicated that users would be able to complete the basic IM tasks without too many problems, but the navigation structure needed to be improved, and the number of menus and options reduced, to keep the new user from getting lost or confused. The display and keypad limitations and the influence of hierarchical menu structures on most phones created real challenges, but a hybrid solution between desktop and phone UI conventions seemed to offer the best results. A follow-on study confirmed that the hybrid design

would increase usability without causing unacceptable side effects. Additionally, we gained more insight into potential problems with navigation, terminology, and privacy settings.



A third study closely paralleled the first, this time looking for problems an ICQ user might have completing the same basic tasks; navigation again posed the greatest difficulty. Support for a few desktop ICQ features⁶⁶ had been deferred until a later release, but the study participants didn't consider the features optional. Also, subtle differences in terminology and visual appearance between the desktop and the mobile client seemed to be more of a problem for these ICQ users.

Additional user feedback came in the form of online surveys mailed to current and former Mobile IM users, plus a set of focus groups. The habits and perceptions of six user segments⁶⁷ were evaluated. Not surprisingly, those users with built-in clients, and/or who were proficient with T9, were the most satisfied with their Mobile IM experience and would look for the product on their next phone. But there continued to be a lack of awareness about the product during service sign-up and handset selection and uncertainty about the cost and billing of wireless data.

The focus groups gathered user perceptions and behavior. The participants were teen girls, women, and men who regularly used Mobile IM on their phone. Their responses mostly echoed the findings of the surveys, and indicated that improvements in the marketing and education of the service could increase uptake and usage.⁶⁸

Even at this point in the evolution and propagation of Mobile IM, we continued to observe users signing on and then signing off again shortly thereafter. The promise of “always on” communications and “IM anywhere” seemed to us to be a compelling reason for users to sign on to the mobile phone and to stay online so that others could contact them. Users evidently had a number of different reasons for the contrary behavior. First, as noted earlier, following the dial-up or desktop mental model the user was just looking to see if anyone he knows, or a particular person, was online at that moment. Second, we theorized that since the typical mobile device only presents one menu or application at a time, some users assumed that they had to sign off the IM client / service to do anything else with the phone. Third, users expressed concern about receiving unwanted messages from spammers, or chatty friends, if they remained signed on.⁶⁹

Wireless Village

The eventual acceptance of Mobile IM among carriers and handset manufacturers may have hinged on leveling the playing field through open standards. Multiple handset models with different features for different carriers are added overhead and a management headache that manufacturers resist; but a few US carriers expected custom integration work on an IM client with a proprietary protocol. So, for Mobile IM clients to be cost-effective, the manufacturers pushed for an open protocol so that they could ship an IM-enabled handset model and have it work with whichever IM service the carriers arranged.⁷⁰

The open protocol was forged under the Wireless Village Initiative, later consolidated under the Open Mobile Alliance. As it looked forward to GPRS-level or greater wireless bandwidth, the proposed open protocol carried a much higher overhead and, as a result, an SMS-only transport was no longer practical. The protocol also did not initially support several IM features that we knew were important – a conventional SMS perspective undoubtedly influenced the WV system design.

It was yet another design iteration of the built-in IM client; we identified the constraints imposed by the open protocol and determined the best user experience for our AIM and ICQ users. New challenges included working within the generic feature set and appearance that resulted from the need for one IM client to support the different IM services, and finding a middle ground between our original server-centric point of view (where all user information is hosted on the service) and the handset manufacturer's phone-centric point of view (where all user information originates from and stays stored on the handset).



The AOL Mobile development team built a new proxy server, which understood the WV protocol and dealt with wireless connection and latency issues, but left the creation of IM clients to the handset manufacturers and third-party developers chosen by the carriers.

As the responsibility for designing and developing the clients moved further away from AOL Mobile, and its team members interacted less frequently with the engineering teams of the manufacturers and third-party developers, it became even more important to have clear and complete guidelines for designing the AIM and ICQ variants of the multi-branded clients. Our custom IM client UI specification evolved into a general UI Guidelines document for WV clients, which has since influenced the overall design of many of the mobile phone IM clients on the market today.

As with the custom client UI specification, we executed a few small studies to confirm the design presented in the UI Guidelines. In addition, we ran a usability study on one of the first WV clients completed by one of our partners. The potential problems uncovered there⁷¹ also reinforced the importance of the design elements we had specified in the UI Guidelines.

Epilogue

Mobile instant messaging services required a design approach that went beyond interaction design to consider everything that impacted the user experience. Constraints were imposed by the wireless data infrastructure, the existing IM service behaviors, the limited and varied handset user interfaces, and the needs and expectations of the users. The design for each platform had to accommodate the constraints while striving for an overall user experience that was fun as well as functional. Occasionally, a platform just couldn't deliver a service that was worthwhile.

As each constraint loosened over the years, with wireless data and mobile phone technology maturing and converging on new standards, the wireless carriers have recognized the market opportunity of mobile instant messaging. In 2006, GSM operators announced a "Personal IM" system that would work somewhat like SMS (sender pays) across carrier networks, and invited the major IM service providers to interoperate.

Though industry analysts both underestimated the timeframe and overestimated the global reach, Mobile IM has fulfilled its promise for millions of wireless subscribers in the U.S. and elsewhere.

⁷² Not bad for what started out as just some *idle chat*...

ENDNOTES

¹ Predecessors included the Unix “talk” program and Internet Relay Chat.

² Known variously as “contacts”, “buddies”, or “friends” depending on the IM service.

³ Elsewhere known as “mobile network operators”.

⁴ RIM’s BlackBerry addressed corporate executives’ demand for mobile email but supplied its own mini-QWERTY keyboard for input.

⁵ Initially MSN Messenger also connected to AOL’s messaging network, to leverage its installed base of users, but AOL fought that with server changes and warning messages until Microsoft gave up trying.

⁶ Shortly thereafter, Nokia and others began working on what would become the Wireless Village Initiative to claim that advantage; it is one of the **Projects and Platforms** described further on.

⁷ Issues with integration would require Tegic working with handset, SIM, and SMS-C vendors as well.

⁸ And the service-independent clients that handset manufacturers also preferred; they didn’t want to tie their devices to a particular third-party provider, or display its brand either.

⁹ Explained in **Projects and Platforms** below.

¹⁰ Also explained below.

¹¹ Such as games, dating networks, and local information queries.

¹² From the *Field of Dreams* management playbook (“If you build it, they will come”). After all, early adopters will try anything, and technology always improves in time to satisfy the rest, right?

¹³ Note that this could create a useful alias for a mobile phone number as well as for a person in a chat room, e.g., if the IM user didn’t want to share her number on a first date.

¹⁴ Configuration by the user: create/enter/change screen name and password, add/delete users of interest; these could be accomplished either from the mobile device or from a Web site. Configuration by the carrier: point-of-sale, customer support, and batch provisioning of mobile subscribers.

¹⁵ i.e., what the user actually wanted to do with the service after all the configuration was done – check the availability of selected contacts, and send/receive messages to/from listed contacts and unlisted users.

¹⁶ Voicestream (T-Mobile) led the way with an SMS-based project, and then the other US carriers followed.

¹⁷ For example, what did it mean to be “Away” when the mobile client always accompanied the user?

¹⁸ As defined by Kathy Potosnak of [Interface Concepts](#).

¹⁹ It was also at this point in time when the “wireless IM” business concept started to become a real “Mobile IM” service, which is reflected in the product names preceding vs. following this point in the article.

²⁰ The ultimate *unwilling subjects* to have their father watching over their shoulder!

²¹ Personas are descriptions of specific-but-fictional people who represent segments of the expected user base.

²² In contrast to *feature* requirements, these are the aspects of “real life” that the users of a system need it to recognize and support.

²³ Which was common in Palm PDA applications of that era.

²⁴ Set Alert, for example, notifies a user if a selected contact has come online. For the mobile user, this was potentially far more valuable since his IM session could be active all day and the contact could come online at a moment the user was not attending to the IM client.

²⁵ A hearing-impaired user was also considered during the Conceptual Design process but that target demographic was not singled out by marketing. AOL eventually fulfilled this promise with an AIM interface for national Telecommunications Relay Services providers. And it is gratifying to learn how many in the hard-of-hearing/deaf community adopted the T-Mobile Sidekick that featured our AIM client.

²⁶ The design team kept them in mind, but the project teams did not. A few years later, we finally had the time and authority to improve AOL Mobile’s software development lifecycle by including personas and other user-centered design milestones and artifacts.

²⁷ “Functional” included sufficient features to use it in a mobile context; “Intuitive” included consistency with the desktop application; and “Fun” recognized what made instant messaging a compelling experience.

²⁸ Sometimes as late as post-launch, though eventually working backward to the design phase with paper prototypes.

²⁹ Academic research papers for mobile were only beginning to show up at CHI/HCI conferences. The leading handset manufacturers and wireless carriers may have developed some techniques, but they weren’t sharing much. We experimented with lab setups, device simulations on a touchscreen PC, etc., as our

limited budget allowed. (The office microwave did double duty as a Faraday cage to evaluate how well the Mobile IM clients handled communication failures! See other examples in the **User Research** section of my [T9 article](#).)

³⁰ Members of our design team (primarily Keith Hullfish, Logan Jaeren, and David Schultz) were participants on each IM project team, collaborating with the product managers and developers. I gave my team a broad view of its design responsibilities; not just “user interface” or “usability” but considering feature sets from both the product manager’s and users’ points of view, assessing the impact of technical constraints on the overall user experience, etc.

³¹ Server-initiated alerts sent to the mobile phone, usually via SMS.

³² Until the advent of Smartphones, most handsets could barely execute one program at a time, which was a problem if an instant message was sent while the user was doing something else on the phone – like talking on it!

³³ Softkeys are the (usually left and right) programmable function keys just below the screen.

³⁴ And, later, the system displayed an explanatory sentence or two in the IM conversation window about interacting with a mobile user, after the first message was sent.

³⁵ e.g., how to fit long translations of desktop keywords and phrases into short menu items and softkey labels.

³⁶ The three main versions were termed “Functional” (offering minimal support for features that differentiate IM from SMS), “Integrated” (providing most aspects of the IM experience in spite of some device limitations), and “GUI” (with a presentation much like the desktop PC application and limited only by the speed of message transfer and mobile text input). A fourth version, termed “Portable” (similar to either the Functional or Integrated versions, but intended to be downloadable to different devices), was also considered.

³⁷ AOL Mobile eventually tried out every variation of this tradeoff as the responsibility for the site passed from one product manager to another.

³⁸ English equivalent: Global System for Mobile communications. The wireless communications standard for Europe is available in most other countries.

³⁹ The first prototype consisted of a cell phone tethered to a laptop running an IM interface!

⁴⁰ The complexity was manageable, in the first release at least, because remote / over-the-air-programming (OTAP) of the user’s handset permitted its Phonebook to be automatically configured.

⁴¹ Phone numbers with fewer than 7 digits that are controlled by each wireless carrier.

⁴² The text message Inbox looks in the Phonebook to match the sending number and displays the associated name.

⁴³ Not the sort of thing you’d want to explain to your mother, but many of the early users were tech enthusiasts. The typical SMS application required too many steps itself, so efficiency trumped simplicity.

⁴⁴ On one network, for example, each user had to manually acknowledge the receipt of every IM command and contact name sent to the phone, up to 50 in all.

⁴⁵ In the meanwhile, the other user may have sent 2-4 more instant messages from her PC!

⁴⁶ A soft launch typically offered full operation of the service but no advertising of its existence to customers.

⁴⁷ A notable misstep was the inclusion of some new learn-as-you-go system messages in a mobile service upgrade. It was deployed the same weekend that the AIM back-end began allowing multiple concurrent logins and echoed all messages to both the desktop and mobile clients. Some users were overwhelmed with instructional and echoed messages they didn’t expect. The problem was compounded by a lack of explanatory materials which had not yet been added to the AOL Mobile Web site nor fully disseminated to the carrier’s customer service department. The executives were *not* happy.

⁴⁸ Via Multimedia Messaging Service (MMS), or “picture messaging”, that allows images and audio clips to be included in text messages.

⁴⁹ The initial proxy architecture set a limit of 30 contacts that the mobile device would keep track of. In 1999, the average number of contacts in a Buddy List was less than 30, so all of them would be included automatically; just a few years later, however, the average was closer to 100, and the system wasn’t very good at guessing which 30 to select.

⁵⁰ Since then, a command line interface called AIM TXT has also been added. (Why? Apparently because it’s how all the other ISPs deployed IM over SMS, and nothing succeeds like mimicking the competition.)

⁵¹ Subscriber Identity Module, or “smart card”. It’s *smart* because it contains a microprocessor, algorithms, and secure data.

⁵² Wireless Application Protocol, a markup language and infrastructure for micro-browsers.

⁵³ A mobile device Web browser by Unwired Planet (soon renamed Phone.com, then Openwave) was already available on most new phones in the US.

⁵⁴ One handset refused to alert the user about a new IM notification until she went back and deleted the previous notification from the WAP browser’s Inbox. On another handset, each new IM notification popped up an alert dialog forcing the user to choose either “View” to follow a link or “Skip” to dismiss the alert. Paradoxically, the better response was usually *not* to choose “View” so that the new IM could be viewed in the current conversation!

⁵⁵ The SprintPCS experience highlighted the Catch-22 situation that occurs with many UI guidelines, and particularly with a new platform: Is consistency with the style guide always warranted, such as when the application is service is the first, or only, experience many users have with the platform? What “violations” would actually benefit an application’s users while risking little in the way of perceived inconsistency, at least until the platform gains a following?

⁵⁶ Terminology – e.g., “refresh” vs. “get new msgs”; iconography - we had too many “ASCII art” status icons; navigation - some users were unable to find new messages, and we had too many different lists and menus.

⁵⁷ The initial design paralleled the desktop client’s Online and List Setup tabs – *but it lacked the tabs*. One of many instances where the desktop experience couldn’t be directly replicated and made its design untenable.

⁵⁸ Often as soon as the free trial ended.

⁵⁹ Data subscription revenues shared among the hardware, service, and wireless providers.

⁶⁰ It was difficult to get carriers to even consider it, at first, especially with 3G and camera phones holding their attention.

⁶¹ A sly incremental update mechanism reduced its absence somewhat, but few of our proposed visual cues to indicate status obsolescence were ever implemented.

⁶² An unexpected outcome of this first built-in client was the strong precedent that it set with product management, even when it was at odds with our own UI specification. For example, a new status icon was created to satisfy a requirement particular to that handset, but our design team subsequently had to work to prevent that icon from becoming part of the UI standard for the carrier’s subsequent mobile IM clients; it even kept reappearing in AOL’s own customer presentations!

⁶³ These design elements were advocated in an updated UI specification which included guiding principles for mobile app design. (Visit David Schultz’s [Mobile IxD site](#) for further insights on guiding principles.)

⁶⁴ e.g., the navigation model, feature set, and basic terminology.

⁶⁵ e.g., its final visual appearance, the event notification scheme, system responsiveness, and overall efficiency.

⁶⁶ e.g., responding to authorization requests.

⁶⁷ A 2x3 matrix of AIM users / AOL members vs. non- / low / frequent usage.

⁶⁸ The participant comments also helped validate or refine the early personas I had developed.

⁶⁹ We subsequently enabled mobile-specific privacy settings and the “Appear Offline” or “Invisible” state, but users may need to be educated about their availability and use.

⁷⁰ I suspect they also wanted the opportunity for their own wireless infrastructure divisions to compete in the “Instant Messaging and Presence Services” provider market.

⁷¹ e.g., lacking a persistent visual cue after the user missed the auditory alert for a new message.

⁷² In contrast to the early estimate by Mobile Insights of 200 million users by 2002, Strategy Analytics predicted that the 12 million mobile IM users in 2006 would grow to 189 million by 2009. For their part, the GSM Association estimated that 33 million people per month used “Personal IM” services in 2009.