

J2ME Experience Reports: Early MIDP Adopters Prove the Business Value

Carl Zetie

Giga Position

Despite the early stage of the technology and the highly constrained nature of the platform, small applications built in J2ME that extend larger enterprise applications can unlock significant, measurable improvements in productivity and accuracy while lowering both operating and capital costs. While many of the benefits of J2ME applications can equally apply to other mobile devices, the fact that Mobile Information Device Profile (MIDP) exploits mass-market, off-the-shelf technologies, including relatively cheap mobile phones and widely available cellular phone networks, means that these benefits are available at a very low price point. These economics are further sustained by the ease of acquisition and deployment of J2ME technologies as well as the high degree of portability across devices and carriers. No other platform offers the same combination of rich client capabilities, low-cost devices, wide availability across carriers and manufacturers, and broadly endorsed standards. The richer capabilities of MIDP should allow it to avoid the pitfalls that undermined Wireless Application Protocol (WAP). Consequently, Giga expects these early projects to be joined by a rapidly growing list of MIDP success stories.

Proof/Notes

Since the MIDP specification was finalized earlier this year, the MIDP within J2ME has been implemented on a rapidly growing number of handsets and deployed on various carriers across Europe, the United States and Asia. Already, a number of real-world experiences are available from which valuable conclusions can be drawn. These early experiences show that MIDP can provide significant business value under the right choice of application. MIDP is most valuable to the enterprise when a small client can extend and leverage an enterprise back end, providing “last mile” access to key enterprise data and business logic. MIDP is most appropriate when the limited user interaction of handsets is not a major obstacle or can be circumvented by other means. It is less appropriate for applications requiring complex local functionality or rich interaction, due to the constraints of both the platform and the devices it is implemented on. The following case studies highlight a list of MIDP benefits as well as important lessons learned that should be applied to other enterprise applications that exploit this technology.

Case Study: Digital Dispatcher

This is a J2ME application that exhibits many of the cardinal virtues of a good mobile/wireless application, coupled with the simplicity and cost-effectiveness of a J2ME front end.

Business Challenges

Digital Dispatcher describes itself as “a wireless productivity enhancing solution for routing, tracking and managing your vehicles, inventory and personnel.” It belongs to the class of “dispatch” applications that aim to reduce costs — both personnel and capital — increase productivity and improve customer service by more effectively allocating tasks to a field service while improving the front office’s awareness of conditions and status in the field.

A typical customer will already have a dispatching application that links to back-office systems. In the pre-wireless scenario, dispatchers take tasks from the system and dispatch them to drivers by phone. This process

can be highly inefficient or even chaotic: On a winter day, it could take four or five interactions on the phone to complete one dispatch. Dispatchers lacked information about what drivers were currently doing and their availability for the next job, what level of oil they had on the truck, etc.

As well as a stressful work environment and inefficient scheduling, this semi-manual system could result in catastrophic errors such as pumping oil in the wrong place, leading to an expensive cleanup.

Use of mobile/wireless devices to dynamically and continuously revise the schedules of field workers as incoming tasks and priorities change during the day, as well as in response to input from the field such as an early finish on a task or an unanticipated hold-up, can improve efficiency by anywhere from 5 percent to 25 percent compared to static scheduling (worksheet issued at the beginning of the day, perhaps revised in response to emergencies). Many companies already use conventional phones/pagers to reschedule workers, but this has its own problems (see below), and companies are finding that even greater efficiency as well as greater accuracy can be achieved with data applications.

Digital Dispatcher's initial market is short-haul bulk materials delivery, e.g., domestic delivery of fuel oil/propane, but the concept and the technology are applicable to a wider range of markets.

Implementation

Information goes into the accounting back-end system; from there it goes to the dispatch board, and a dispatcher examines it and sends it to the nearest driver (based on last known delivery location). The dispatch is pushed to the driver's J2ME phone. This process eliminates the back-and-forth communication, as well as averting errors in communicating the address or other instructions.

As drivers complete each delivery, they punch some simple information (account number, number of gallons, means of payment) into the J2ME client, and send it back. The completed task gets posted to the delivery board with an automatic time stamp. Using this information, dispatchers know where the trucks are (to an accuracy close enough for this business purpose). These reports get reviewed once again and are then posted to the accounting system.

The system also provides monitoring of when messages reach the driver so the company can be sure that the message arrived on time in case there is any question later.

In some mobile implementations Giga has looked at, acceptance of an automated system (especially one with connotations of "looking over the worker's shoulder") has met resistance. In this case, drivers have incentives to use the phone and be more effective/efficient in their job. The fact that the application is extremely easy to use aids adoption.

Business benefits of this system are immediately measurable. Kelly Energy in Pennsylvania and Delaware is able to operate with fewer staff in the dispatch operation; this savings alone paid for the system. Improved efficiency allows drivers to make one more delivery on a 20-delivery schedule, equivalent to eliminating one full truck from the fleet and a six-figure saving. Other less direct benefits were also discovered. The dispatch center is quiet and efficient, and stress levels are down; the system is also more resilient to bad weather and does not descend into chaos when customer calls begin to flood in. Digital Dispatcher's customers can also tell when trucks are out of calibration if there are discrepancies between what goes in and what comes out. Discrepancies can be discovered in a day instead of weeks, leading to significant savings. Finally, because dispatchers have better and more accurate information — they can more accurately tell customers when to expect a truck, leading to higher customer satisfaction.

Other unanticipated benefits can also emerge. In one case, a company's insurer offered a 10 percent or more reduction on premiums because of elimination of accidents. This alone could pay for the system.

Lessons Learned

In theory, a system like this could be built on a variety of platforms, and in fact systems that are similar in principle have existed for a number of years on larger and/or proprietary hardware. J2ME brings a number of unique advantages to this application:

- **Cost:** J2ME handsets are extremely cheap compared to other terminals. Prior to developing the Digital Dispatcher approach, staff had been working on more sophisticated systems, e.g., using a ruggedized laptop in-truck. But the likely cost of \$6,000 to \$10,000 per truck was too much for many customers. MIDP phones bring the hardware costs down to a level that is affordable for many more businesses and justifiable for many more purposes.
- **Simplicity:** For this particular application, the data entry requirements are simple and well suited to a numeric keypad. The limited capabilities of the phone become a benefit rather than a burden since they ensure simplicity of operation. This also translates into lower training costs and fewer operator errors.
- **Ease of deployment:** The solution exploits off-the-shelf technology in the form of MIDP phones and conventional wireless networks (in this case, **Nextel**). No specialized network knowledge or complex wireless setup or provisioning is required of the customer.

Summary

As noted above, this is a cardinal example of a good mobile/wireless application. It takes an existing back-office/front-office system and extends its reach to where the workers actually perform their tasks, thus “closing the loop.” It ensures that events requiring changes to activities are delivered promptly, for greater productivity, and also gathers data at the point that it is generated in the field, eliminating errors and improving accuracy. This particular example also highlights the ability of J2ME to deliver these benefits at a cost point that sets a new low in the barrier to entry and economic feasibility of mobile technology.

Case Study: AirClic for Boston Public Schools

A phone-based application does not have to be limited by the constrained interactivity of a keypad. Low-cost bar-code technology provides the interface to encode actions as well as data in an archetypal “paper replacement” application.

Business Challenges

Boston public schools’ attendance offices spend much of their time on the streets of Boston looking for those who should be in class. If they find a youth who should be in class, they refer to a list of students to check where this student should be, who to contact and other information. Previously, the list was a computer printout that the officer carried around. Not only was the list rapidly out of date, but at around 63,000 students and several phone books in size, it was also cumbersome and inefficient.

AirClic has implemented an application that replaces the printout with a **Motorola** phone on the Nextel network, which the attendance officers have been trialing since May 1. An officer types a student’s name into the phone that connects to Boston Public Schools’ back-end systems and returns identifying information including age, address, attendance record and parents’ phone numbers. If an uncooperative student gives a false name instead of his or her own, the attendance officer can use this information to check the identity. Enhancements to the system will show a student’s bus routes and medical information, such as allergies. With access to the student’s schedule, the officer can even see what specific class a student is cutting. As the school system rolls out the bar-coded school ID cards, officers will be able to use small bar-code scanners attached to the phones to identify a student’s file even more quickly.

Several other school systems have expressed interest in the solution. Some school systems are also interested

in using the same low-cost, easily-deployed bar-code technology for other purposes, such as tracking students on and off buses (among other things, federal subsidy dollars depend on reliably auditing bus usage) as well as more conventional bar-code applications, such as tracking inventory of supplies.

Implementation

The hardware for this solution is a Motorola i85s on the Nextel network, integrated with a bar-code scanner in the form of a small sleeve that attaches to the phone. (In the future, the cameras that some handset manufacturers are building into phones could be used as bar-code readers as the quality of the camera improves.) The bar code is used not only for data capture but also for “action capture”: The user is provided with a printed sheet of actions. Scanning the bar code associated with an action, such as “search for student,” causes the associate action to be carried out. Data can also be input more conventionally using the phone’s keypad if necessary.

The core of the technology solution is AirClic’s Mobile Information Platform. The platform provides key services, including synchronization of record sets between client and server, associating “smartcodes” (bar codes that define functions) with actions as well as management of devices and end users. When codes are scanned, they are stored in a standard J2ME record store (RMS) along with a time stamp, allowing offline usage. When the data is synchronized with the server, a unique user ID (for the user) and optionally a device ID (from the scanner) along with the data is transmitted to the server, where it gets unpacked as an Extensible Markup Language (XML) structure. For efficiency, data is sent across the air in a simplified, tokenized form of XML. The platform is responsible for ensuring completion of the transfer, for example, allowing an interrupted transfer to continue where it broke off rather than restarting from scratch.

Once the data is received at the server, the platform interprets it as necessary, for example, as a request for a student’s details. Currently, enterprise integration uses a SOAP-like interface (reflecting the fact that development started before SOAP standards were finalized) and is evolving to SOAP and WSDL. If the data includes records that need to be processed by an enterprise back end, the data is cached by the platform until the back end requests it from the platform, decoupling the user in the field from being dependent on the availability of the backend.

The Mobile Information Platform also includes a rendering system that allows the user interface to be generated dynamically. The rendering engine receives a WSDL document that contains all the student details in a structured form, which allows the specific details displayed to be changed with relative ease (for example, to add allergy or other medical information to the display). The rendering engine interprets and slims down the document, then passes it to the device, which does the final rendering. The device combines the data with a templated form (similar in concept to an XSLT template) to create the final display. Although it may appear complex, in practice the time taken to render a student’s file from the time the student’s ID is dispatched to displaying the file on the screen is typically under two seconds, assuming of course that the user is within network coverage.

Lessons Learned

As well as the common benefits of the other case studies included here, this solution illustrates a number of interesting benefits:

- **Simplicity of operation:** In conjunction with the AirClic platform the scanner also provides for innovative action-driven solutions that circumvent the usability limitations of a phone keypad.
- **Simple, low-cost wireless scanning solution:** In addition, a scanner integrated with a wireless J2ME phone provides a low-cost solution for many conventional scanning requirements such as inventory. In combination with a wireless phone, this may greatly expand the economic feasibility of barcode solutions
- **Flexible and dynamic user interface:** The combination of templated forms and structured (XML)

data allows the application to be updated and enhanced without the deployment headaches of a conventional static client application.

Summary

A J2ME mobile phone can become a handheld scanner at relatively low cost. Not only does this provide an effective workaround to the data entry limitations of a keypad, it can also provide a simple way to specify actions and ultimately to integrate with enterprise services. A flexible user interface allows an application to meet evolving requirements without deployment headaches.

Case Study: mPortal for Xerox Corporation

A classic field-service application demonstrates the importance of usability and in particular task analysis when working with a small device.

Business Challenges

mPortal offers wireless solutions to both carriers and enterprises and its offerings include both a platform (Enterprise Everywhere) and a professional services staff that offers design, strategy and implementation services.

Engineers work in the field servicing **Xerox** equipment. Previously they were using laptops to upload a static schedule defining the calls for the day. In the old process, the engineers had laptops but relied on phone lines for connection along with other traditional data gathering and entry mechanisms. mPortal created a J2ME application for i50 and i85 **Motorola** phones. This application allows service engineers to work offline, synchronizing to a central server as network coverage permits. Consequently they can close more work orders a day. They no longer need to call into the call center to get information such as location of a client or details of the job. As noted above, moving to dynamic scheduling of service calls from a static model can deliver productivity increases in the range of 5 percent to 25 percent, depending on the nature of the tasks and the degree of unpredictability and urgency in service calls, although the specific improvement in this case has yet to be formally measured.

Implementation

The key to this implementation was the emphasis mPortal and Xerox placed on the front-end stages, analyzing and understanding the business case as well as the work tasks of the engineers. They spent time studying the business case with Xerox, including estimating the projected return on investment before attempting to design an application. Xerox itself spent considerable time “riding along” with the service engineers who would use and give feedback on the wireless application prototypes to refine the application and ensure the end result would cover all business processes.

Once the requirements were thoroughly understood, focus moved to the user interface, both in terms of user performance and user acceptance. It was important to persuade users that the task could reasonably be done with a small screen and keypad instead of the familiar laptop. Designers focused on what the core activities of the engineers on site were, and then studied how to make those activities as simple as possible. Some of the techniques used included pick lists and numeric codes, as well as reviving some ideas long used to optimize the performance of expert users of “green screen” terminal applications, such as “jump” screens. To achieve high user performance, the team went through three iterations of development and user feedback once the interface was implemented on the device. The application was put through user testing in the field, then a larger pilot, before being rolled out to a user population that may eventually number several thousand.

Although the requirements and task analysis were critical elements to success, the technical architecture of this project was also important. On the client, the application uses the MIDP RMS for data storage. (Note that raw RMS performance varies widely on different handsets. According to the TaylorBench tests (see References), the i85s is surprisingly slow to write RMS memory compared to its read times). The client

communicates using an architecture that is rapidly becoming the “classic” J2ME/J2EE model, namely HTTP to a servlet (see References). The payload in this case is a set of name-value pairs. (The platform is capable of supporting a form of lightweight XML, but this capability was not used on this project). All communication is synchronous, which was the business process requirement in this case. (Again, asynchronous is possible if the process requires it). On the server side, mPortal’s Enterprise Everywhere platform provides many common components, including session management, logging, device management, and network management, and is implemented within a standard application server.

Xerox also continues to support a laptop-based solution for its customer service engineers but has found that for many users the phones are much more convenient to carry and a better fit to requirements.

Lessons Learned

This project illustrates that the importance of requirements gathering and task analysis grows rather than shrinks with a small, constrained device for user interaction. Key lessons are:

- Focus on requirements and tasks: The necessary understanding of what field-based workers actually need may only come from accompanying them in the field. Even the best lab-based studies or focus groups in meeting rooms may not expose the essential context of use.
- Three iterations of design: As Giga has observed in previous research, it is not surprising to see designers work through three iterations of working prototype before being satisfied with the interface design if they are truly focused on making the application acceptable to the users.

Summary

Usability in the broadest sense — incorporating requirements, task analysis and iterative prototyping of the user interface — are essential to creating a truly successful design regardless of the deployment technology. These factors become even more critical with a small, constrained device such as a MIDP-enabled handset.

Case Study: Agea for Metrowerks

For applications where precise control of layout of the user interface is not essential, MIDP can provide a very high degree of portability with careful application design.

Business Challenges

Metrowerks wanted a sales force automation (SFA) application for its own use. One of the key requirements was that the application be able to run on a wide variety of devices, including both phones and PDAs, according to the preferences of the salesperson. Metrowerks worked with **Agea** to create an SFA client that has been ported to a variety of phones, PDAs and communicators and integrates with an existing **Siebel** backend. The sales force now has the ability to work offline against local data or to synchronize at will (coverage permitting) with Siebel.

Implementation

The client application comprises about 50,000 bytes of compiled code, including an XML parser, local database (implemented over RMS) and messaging. On the server side, session management is handled by the Agea platform, with additional custom integration work to connect to Seibel. (The Agea platform includes a notification engine that can raise events to escalate issues and manage workflow, e.g., to get a sales exception approved, but this capability is not yet used in this particular application.)

The application provides offline capability through a local database, essentially an abstraction of RMS that provides a recordset interface. Data is synchronized with Seibel via the middleware on demand. The synchronization is implemented using a messaging layer to provide reliable connectivity. The messaging API is based on JMS (although JMS is part of the J2EE standard there is as yet no JMS standard or proposal

within J2ME) and is a custom implementation; existing JMS products were considered to be too large for the smallest handsets, but can be plugged in on a sufficiently powerful device such as a PDA. The payload of the messages is a compressed XML, again a custom implementation. Agea and Metrowerks experimented with standard XML in an earlier prototype but found it to be too verbose for acceptable performance. The intention is to adopt the Web services standards defined by JSR 172 when they become available. The messaging layer can synchronize around 20 to 25 accounts in about a minute on CDPD using binary data, faster performance would be expected over 2.5G connections such as GPRS or 1xRTT.

The two companies have demonstrated that the application runs acceptably on a variety of MIDP platforms. Porting required almost no coding; the major task was testing to verify compatibility. An explicit decision was made to port with minimal code changes, keeping the user interface intentionally basic, rather than optimizing it for each device.

Lessons Learned

- Portability is possible: Portability is one of the great promises of MIDP, and this application proves that it is possible with very few compromises given suitable attention to design. The largest decision is whether to adopt a generic user interface that will work across all devices or to optimize for each one.
- Integration requires effort: The absence from MIDP of better interfaces in areas such as data access and reliable messaging means that solutions are being reinvented in many places. Eventually these efforts will make their way into community best practices and off-the-shelf products.

Summary

If user choice of delivery platform is an important requirement, MIDP can provide highly portable applications (within its inherent constraints). Efficient enterprise integration requires careful design and some custom work, however.

Alternative View

The most likely scenario is that these early successes portend even wider adoption within the enterprise. The next most likely scenario is that the adoption seen to date represents the “low hanging fruit” — small, comparatively simple applications that fit well within the constraints of MIDP. After an initial rush of deployments, further suitable opportunities will slow to a trickle, and MIDP will remain a small factor in mobile enterprise application deployments. The majority of enterprise deployments will be found to demand the power of a laptop or high-end PDA. We regard this outcome as a relatively low probability at this point, subject to revision during the next six to 12 months as more proof points emerge.

Findings

These early proof points demonstrate that even applications as small as a MIDlet (typically, tens of kilobytes of code) running on a phone-size device can deliver significant value to the enterprise when they leverage existing enterprise systems, providing “last mile” access to field-based employees. Key lessons demonstrated by these early initiatives include:

- MIDP promises extremely low-cost mobile solutions: J2ME-capable hardware is very cheap compared to other terminals, even PDAs and particularly laptops. A phone-based deployment can bring costs down to a level that is affordable for many more businesses and justifiable for many more purposes.
- Simplicity of use is key: Keep data entry requirements simple and well suited to a numeric keypad. Minimize data entry by careful design or by exploiting other means of entry such as a barcode scanner.

- Portability is possible: Portability is one of the great promises of MIDP, and experience proves that it is possible to achieve with very few compromises given suitable attention to design. The largest decision is whether to adopt a generic user interface that will cross all devices or to optimize for each one.
- Integration requires effort: The absence from MIDP of better interfaces in areas such as data access and reliable messaging means that solutions are being reinvented in many places. Eventually these efforts will make their way into community best practices and off-the-shelf products.
- Focus on requirements and tasks: The necessary understanding of what field-based workers actually need may only come from accompanying them in the field. Even the best lab-based studies or focus groups in meeting rooms may not expose the essential context of use.
- Three iterations of design: As Giga has observed in previous research, it is not surprising to see designers work through three iterations of working prototype before being satisfied with the interface design if they are truly focused on making the application acceptable to the users.

Recommendations

MIDP is ready for adoption for enterprise applications. Consider candidate applications where a small, simple user interface can exploit heavyweight processing on the back end. In such applications, a relatively modest incremental investment in hardware and software can return significant gains in productivity, accuracy and responsiveness. If you have set aside projects in the past because of the cost of equipment or deployment, or the difficulty of wireless connectivity, reconsider those projects in the light of the economics of MIDP handsets.

Emphasize usability in the broadest sense, with a major emphasis on requirements gathering and task analysis. With the limited interactivity of a handset, the application has to match extremely well to the needs and activities of the user. Expect to spend time in the field observing workers in action, and don't be surprised to need three iterations of prototype to optimize the design.

Don't underestimate the challenges of enterprise integration. Raw MIDP offers only a basic HTTP connection — anything more sophisticated must be bought or built on top of that byte stream. Despite this, experience proves that it is possible to build robust, efficient and flexible integrations, in many cases using variants of XML protocols optimized for efficiency. In the future, expect to see more Web services-like integrations, although still optimized for the bandwidth and latency characteristics of the wireless WAN.

References

Related Giga Research

Planning Assumptions

[Mobile Application Architectures From Laptops to PDAs: Balancing the Tensions Between Function and Data](#), Mike Gilpin and Carl Zetie

[Market Overview: Communicators and Other Converged Mobile Devices](#), Carl Zetie and Ken Smiley

[Web Services on Mobile Devices: Experiment Now, Adopt Later](#), Carl Zetie

IdeaBytes

[IT Trends 2003: Mobile and Wireless Devices](#), Ken Smiley

[IT Trends 2003: Mobile Enterprise Application Development](#), Carl Zetie

[IT Trends 2003: Mobile Enterprise Application Architectures](#), Carl Zetie

[J2ME vs. .NET CF: Compare Platforms to Platforms](#), Carl Zetie

[When Time Is of the Essence: A Business Case for Deploying Mobile Applications](#), Carl Zetie

[Mobile Information Device Profile Implementations Will Diverge Now, Converge Later](#), Carl Zetie

Relevant Links and Other Sources

TaylorBench J2ME MIDP performance benchmarks, www.pogit.com/midp/bench/index.html

Primary Sun site for information about J2ME, including best practices and technology details, <http://wireless.java.sun.com>