Energizing a Mobile Workforce:

Real World Solutions

Ken Deh-Lee, Gary Loob, and Deborah Caswell

Hewlett-Packard Company
100 Mayfield Ave, MS 36UO
Mountain View, CA 94043
Phone: (650) 691-5083
Fax: (650) 691-5576
Ken_Deh-Lee@hp.com

HP has extensive experience in providing field service delivery worldwide. HP's Mobile Solutions Lab creates, deploys, and supports tools that are used by our service repair professionals, called "Customer Engineers" (CEs). The tools increase a CE's efficiency, with the end result of improved customer satisfaction. In a geographically distributed, international company, it's not only the sales force and service force that are mobile. Professionals in all positions at HP require access to applications while away from the office, whether telecommuting or traveling on business. HP's tools allow mobile professionals to access the latest product information and to provide real-time work status.

When designing a mobile solution, one needs to consider several factors, including the amount of data transmitted, the required richness of the user interface, and the rate of change experienced by data being distributed. This paper describes the problems to be solved and discusses the technology and hardware platform choices made for deploying these tools.

Finally, discussion will address future technologies/hardware platforms that can be utilized to further assist a mobile workforce.

1 Introduction

The Mobile Solutions Lab (MSL) is one of the R & D Labs in HP's Customer Support Organization. MSL's mission is to develop solutions that can be used by Mobile Professionals providing customer support. We strive to keep abreast of emerging mobile technologies and seek opportunities to incorporate these technologies into our solutions where advantageous.

The solutions we develop enable HP Customer Support to provide increased customer satisfaction by reducing problem resolution time without sacrificing quality. These goals are obtained by

automating knowledge delivery to the field and providing constant work status information back to the service call center. Work status reporting enables HP to make better logistical decisions allocate resources to resolve the problem within the committed timeframe. Providing quick access to latest information allows HP engineers to concentrate on the problem instead of gathering and navigating through lots of information.

Our solutions use new technology in a compelling way to provide value to our customer and business partners. Aiding our mobile workforce with these solutions is one key step in meeting or exceeding the customer's expectations.

The purpose of this paper is to share some of our knowledge and learning experiences gained from deploying real solutions. To that end, we present two such solutions. The first solution, M-Web, is a mobile technical reference application using push technology. The second solution, SmartPhone, is a Wireless Application Protocol (WAP)-based application that is used to update and retrieve up to the minute information on support calls via mobile phone. For each solution we examine the problem being addressed, the underlying technology utilized, and our solution. The appendix contains additional reference information as well.

2 Mobile Web (M-Web)

M-Web is a complete**,** mobile environment for our CEs, providing them with quick access to the information and tools they need to deliver better hardware support for Hewlett Packard. It runs on a laptop that connects to the HP intranet while in the office and stores data locally for disconnected use while on the road.

2.1 Problem Statement

Customer engineers are required to meet contractual agreements, to reduce costs by completing tasks quicker, to support an ever-growing list of products, and to increase customer satisfaction by exceeding their expectations. That's a tall order! The M-Web project was created to aid CEs to better meet these requirements. M-Web attempts to resolve the following common field issues:

1. The service information is not integrated, nor is it organized. There is an ever expanding number of information sources: manuals, web sites, computer–based training, and CD subscriptions. Each uses a different technology and a different interface; each stores content in different formats.

   Without M-Web, needed information can be hard to find. M-Web focuses on providing easy access to the most important information needed by CE's: parts information, self-test information, configuration data, and failure codes.

2. Network-based tools, while extremely useful, are difficult for CEs to access. In our rush to the web, we have lost sight of one important fact. Although CEs have always been mobile, they spend less time in the office and more time on the road than before. As a result, their access to a high speed local area network (LAN) connection to information within HP is

reduced. Tools that rely on tethered access are very problematic. Phone lines are often difficult to use at a customer's site (especially if they are digital connections) and current wireless access technology is either unavailable or much too slow to be effective for large content.

3. CEs are constantly confronted with an ever-expanding product base. Information must be current**.** Unfortunately, the information that is necessary to support these products was not always available or easily found.

4. Information locations change and new locations become available without notification. Even if one knew the locations, it was difficult to know which information is the most recent and accurate. The proliferation of web sites only makes the situation worse.

5. There was no easy way to integrate local and/or region specific content into the available tools. Special Operations Plans, district phone and pager numbers are examples of local info.

6. Traditional communication media are not effective. Because email is overused, it can become overwhelming, and thus, it is not the most effective communication medium, especially for very important, time critical messages. We are constantly reminded that CE's cannot read all of the email they are bombarded with.

2.2 Our Solution

The M-Web solution uses push technology to maintain a local product reference repository on a laptop with the latest subscribed information. When using M-Web, CEs do not require a network connection out in the field to access reference materials. Having a repository of product information can be particularly important on reactive calls when a CE needs to do some investigation while away from the office. After installing M-Web, users populate the repositories with the newest reference materials to which they have subscribed which usually include the parts list, error codes, configuration and diagnostic guides, service notes, and firmware reports.

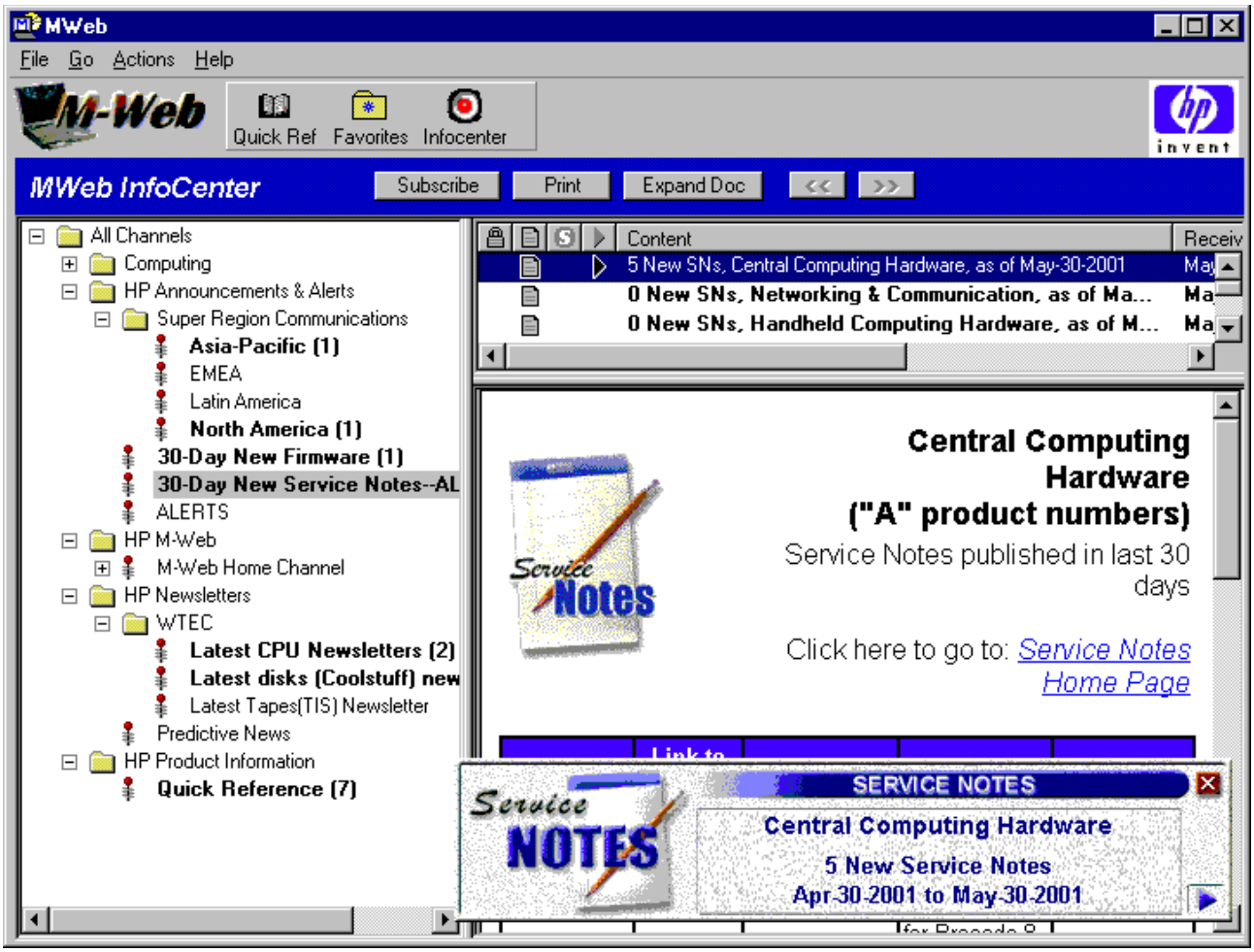

Figure 1 M-Web InfoCenter View

The M-Web application is a self-contained executable. It provides the M-Web user interface and contains an embedded web browser. The web browser gives M-Web multi-document-type display features with the use of plug-ins to support formats such as excel, word, powerpoint as well as pdf files. By using an embedded browser, M-Web can retain the benefits of executable applications such as the ability to store application-specific state, more programming flexibility for features, and multi-document handling features of a web browser. Figure 1 shows the M-Web Infocenter View which gives users access to newsletters, alerts etc. A service note notification (flash) is also visible. Figure 2 shows the M-Web Product Reference View. The pane on the left represents the product tree; the pane on the upper right is the document list for the hi-lighted product. The pane on the bottom right is a web browser object used to display a product reference document

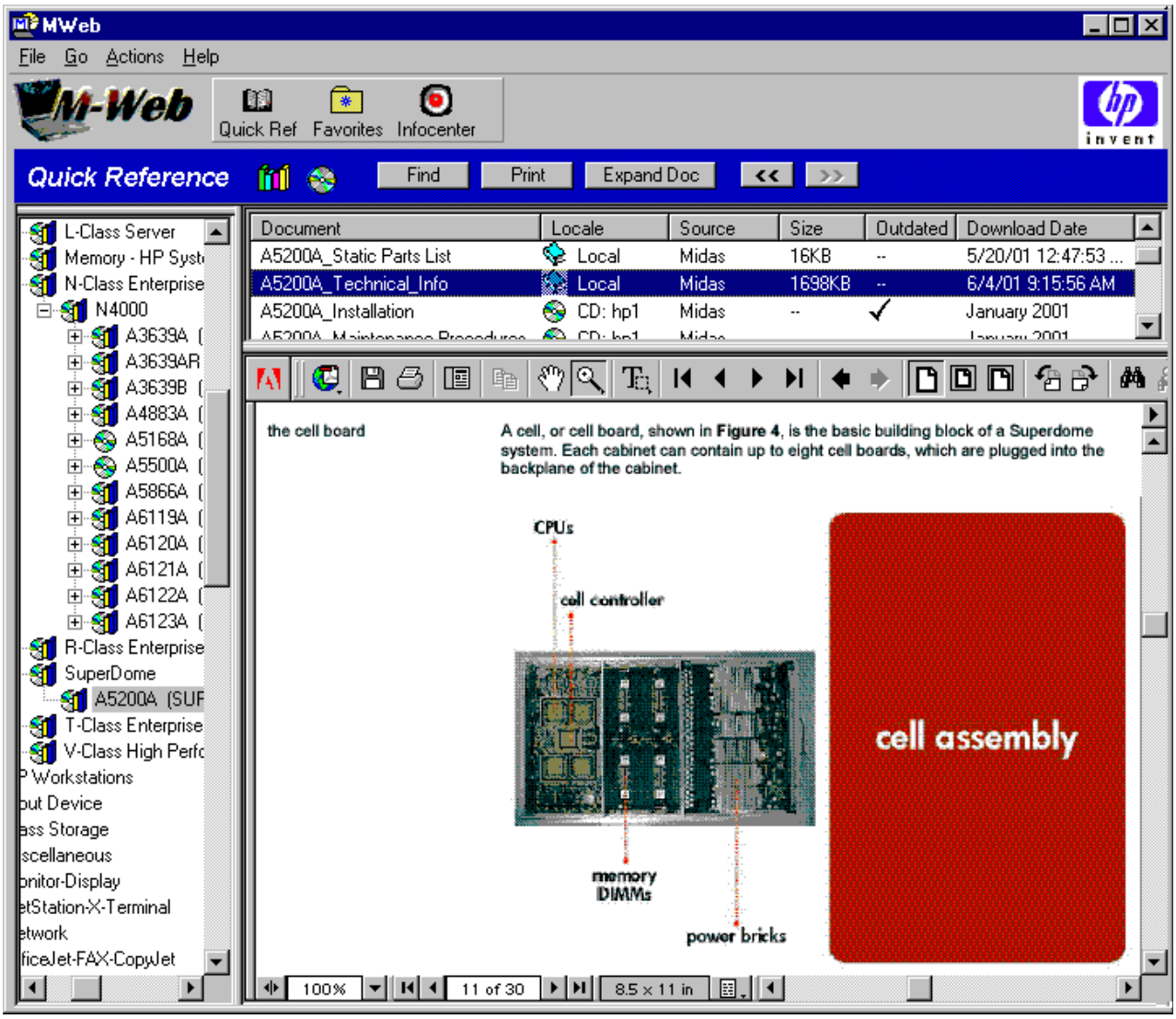

Figure 2 M-Web Product Reference View

As much as is practical, the philosophy of M-Web is one of selective local data storage. Thus it provides CEs with quick and immediate access to the knowledge they use most frequently without requiring a network connection. For information delivery, M-Web relies heavily on a supporting intranet push infrastructure to update information stored locally as well as to provide a means to distribute targeted bits of information. Much emphasis is given to the backend staging process of collecting information and integrating it into a single, standard product hierarchy (tree). It also features publishing regional and organization-related news and alerts.

M-Web benefits to our customers include reduction in call closure time (by providing quicker and easier access to important and often-used repair information) and increased CE efficiency. Our engineers are always kept informed with up-to-date technical support information from support organizations and product divisions within HP. In the end, customer satisfaction is improved by aiding CEs to meet and exceed customer expectations.

The push technology infrastructure makes much of the above possible and is leveragable to other business solutions.

2.3 Understanding Push Technology

Push technology in its simplest form refers to a way of delivering tailored information to a user's client device (laptop, personal digital assistant (PDA), phone, *etc*.) from a server. When we talk about push technology, it is important to keep in mind that it can refer in most general terms to a technology that supports a subscription process. The term, *push technology*, is not well defined. It is not a standard or a protocol. Software vendors that sell syndication solutions sometimes use the term to describe the technology they offer. Note that the word push refers to the fact that information is proactively delivered to subscribers. It does not refer to the implementation of the software. Whereas one might assume that push technology would be implemented with a server managing the distribution process to many receivers, in fact the implementation used by M-Web today puts the distribution control in the client. Thus, it is a pull implementation. For today's use model of nomadic (sporadically connected) clients, putting the distribution control in the server would be too much of a burden. It would have to poll every subscriber continuously to determine whether or not they were connected, and maintain a queue of pending deliveries for each one that is not connected. This would severely limit the scalability of the solution. However, in a world where the norm is continuous mobile connectivity, reliable multicast infrastructure could be used to instantaneously deliver updated information. This is the holy grail of push technology.

A user of push technology must subscribe to information. The category that is selected for subscription is called a *channel*. Once subscribed, pertinent information packages are delivered to the user for that channel. In most implementations, an agent must be installed on the user's client device. The agent is usually started up as a background process that polls the server in periodic intervals to request new information. After a request is received from the agent, the server will determine the user's validity, the set of information that must be delivered, and any new subscription channels that need to be made available.

The information is delivered in a compressed package (like a zip file) and can be encrypted. HyperText Transport Protocol (HTTP) and proprietary communication protocols can often be selected. Proprietary protocols have extended features to support optimized background processing such as non-intrusive downloads. HTTP protocols are supported to enable use of existing corporate firewall schemes. Note that users will not have access to information delivered until the download and unpackaging occurs. Some of these constraints must be addressed when determining the total size of the information being published to a channel. Users with low bandwidth connections may require more time before information is viewable. The proprietary protocols support partial downloads such that upon the next network connection, the download can continue where it was interrupted.

It is important to understand the impact of multiple channels with fine-grained content (small nuggets of information) versus one large channel with coarse grained content (large nuggets). The trade-off will be in the amount of work required to subdivide information and channel creation/maintenance against usability for the user. Having finer grain channels allows for quicker downloads due to smaller content size, and it supports the user with limited disk space. Finer grain channels offer users more configurability and ability to better target information. Other issues related to channel granularity include the affect of the Push Technology software implementation and its limitations. There may be performance implications as well as supportability issues. On software upgrades, having more channels may require more work.

Sophisticated push technology solutions consist of a custom client application and a backend infrastructure portion (Figure 3). The client application contains value added features such as the ability to search, file placement and custom (intuitive) presentation for the user. The backend infrastructure is usually powered by software from third party companies that specialize in push technology. Push technology can deliver various formats/types of files such as a setup package, an executable or database files and therefore, can handle most solution requirements. Uses in more advanced solutions can include automatic delivery of software and knowledge repository updates.

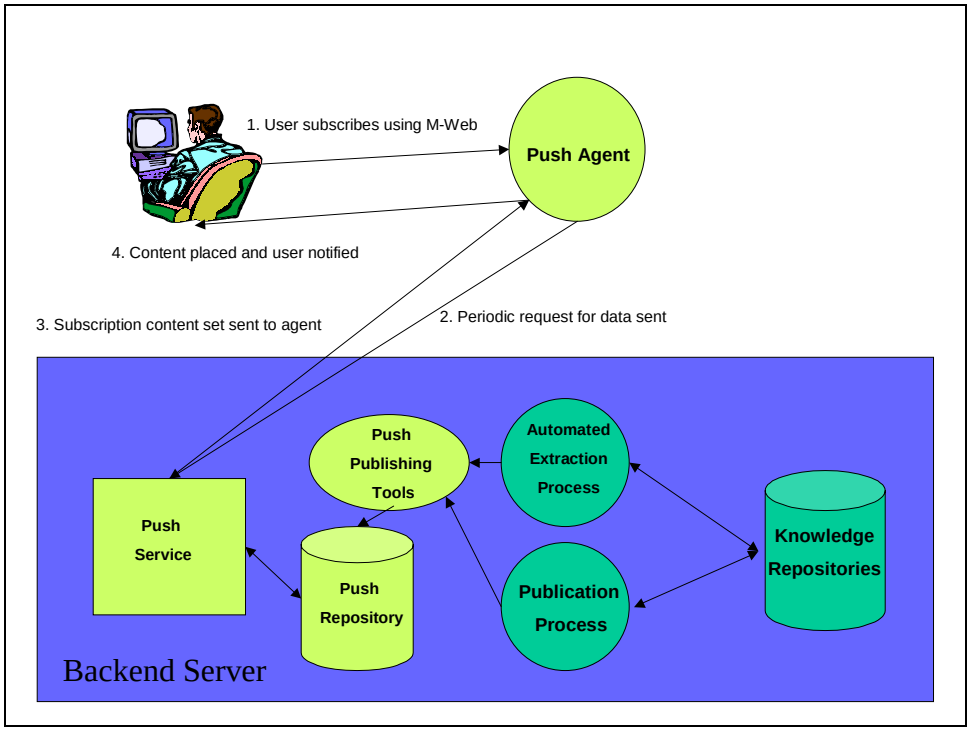

Figure 3 General Subscription/Knowledge Flow

Simple solutions can be implemented right out-of-the-box meaning that the push technology software provider has packaged a generic client for use with limited configurations. The out-of-the-box client will allow browsing of pushed information in the server's predefined hierarchy. In general, these packages are tailored for delivering news type information which may include new

product announcements, corporate news etc. Note that an email subscription solution can be thought of as a push technology application.

Push technology software, used correctly, can offer great advantages over other information delivery/access methods such as ftp, direct web access or email:

- Automated Delivery of targeted Information – Once a user has subscribed to information, new content can be delivered whenever a network connection is present. Note that for an always-connected user, this can be looked upon as "immediate" delivery whenever new content is available.

- Notifications – Users can be notified of new information arrival. A pop-up dialog, called a *flash*, can be displayed. Some vendors also display content for a channel via ticker window. Notifications are required because the agent that collects the information is always running and can display messages when they arrive without waiting for the M-Web user interface to be started.

- Publishing Tools - Push Technology infrastructure also accommodates publishers by providing tools which allow information submittal as well as information extraction from another source. In most cases, information is already published on a website but users may not know of its existence or may simply be too busy to keep up with new content. In these situations, publishing tools to *scrape* the web page can be of great use.

- Statistics - Data gathering and statistics reporting can be embedded into the tools/infrastructure. This can aid in determining which groups of information are being downloaded more than others. These statistics can then be used by the content generation team to make better decisions on content investments.

- Remote Processing – Aside from just dropping data onto the client, scripts or executables can be launched. Uses can include automated patching and upgrades, metrics collecting etc. Deployments of new clients or fixes can become quite transparent to users. Alternatively, a flash may be displayed, notifying a user to begin an upgrade. The design of the client application must be thought out in advance to reduce the dependency of having to redistribute all executable components. However, an update of a help file can be easily accomplished.

2.4 How M-Web Uses Push Technology

M-Web uses a 3rd party implementation of push technology to deliver information from many disparate sources and offers subscriptions through a single user interface. Delivery and organization of technical product reference information is handled differently from newsletters and other periodic publications. The product reference area is subdivided into multiple channels representing product lines. Within each channel, additional add-on content such as service notes can be selected. The information from the product reference channels are integrated into a single product tree after delivery (will be discussed below). The quick reference view (Figure 2) displayed

by the M-Web user interface offers access to the technical reference documents. More dynamic information such as alerts, newsletters and periodic reports are deposited into folders representing the channels for viewing and is accessible through the infocenter view (Figure 1). The publishing process for product reference information is also quite different from the dynamic information (Figure 4). New channels can be created at will on the server and distributed to users for subscription. Automatic subscription is even possible. For instance, during the Y2K sensitive months, a Y2K alerts channel was created. All M-Web users were automatically subscribed to it creating a tailored communication path to CEs. This knowledge area is open to partners who can benefit from publishing to the field workforce.

Push technology tools/wizards are used heavily to populate data. New M-Web application releases can be delivered as well as agent upgrade packages. Configuration and support files are automatically updated via the M-Web channel, invisible to the user. Surveys can also be sent through these channels to solicit feedback from users. Notifications (flashes) are generated for any time sensitive deliveries of information such as alerts and service notes reports.

Another important user issue of content organization/presentation was resolved by using a product tree hierarchy. The product tree represents HP products in a tree structure. It allows users to navigate quickly and intuitively through information. Once at a leaf folder, which represents a specific product (product number), all information pertaining to that product is available regardless of its origin ("one-stop-shopping" approach). A backend extract and merging process provides the data integration into the product tree. After the backend process has completed and is validated, the appropriate portions of the reference materials are pushed to the users.

To reduce the amount of download overhead for the product repository, a set of product reference CDs was created containing older, and therefore static, product information. The use of CDs posed a content integration problem between local hard-drive content and the CD content. M-Web allows seamless access of both sets of information by providing a single product tree view. An index is deposited on the hard-drive that contains product hierarchy information for the CD. During product tree navigation, information from the local repository and CD index is combined to provide a merged view. Using the CDs, disk space usage was reduced along with network usage, and users were not penalized with the download time. Other features to reduce download impact include using a protocol provided by the push technology software that only processes pushed data with available CPU cycles and network bandwidth. Performance is also improved by sending and applying only byte-differences within a changed file.

M-Web users periodically connect to the network when they are at the office to get pushed information updates. The usage model is nomadic. In some areas, a wireless modem may provide more frequent access. M-Web is deployed worldwide.

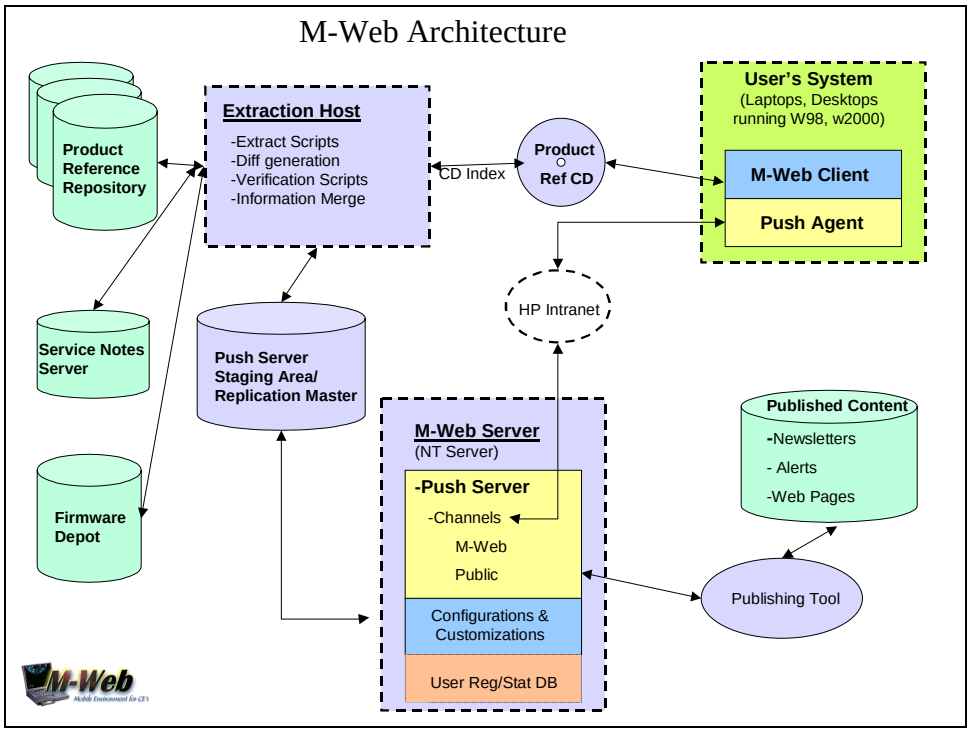

Figure 4 M-Web Architecture/Process Flow

2.4.1 M-Web Architecture/Process Flow

Figure 4 shows the M-Web architecture and data flow within the system. Here are the steps taken to prepare and deliver informational updates to a nomadic workforce:

1. Information is extracted from several repositories on a set interval and merged onto the extraction host. The extraction host is used to run data validation steps as well as to determine the differences between the current and previous extracts. The validation checks are important to minimize the chance of delivering bad data or redundant information and to determine the download impact to the user. The CD index is also kept on this system to manage product tree changes.

2. Once the data has been validated, it is moved to the staging area on the push server to integrate it into a structure for delivery. The content merges take place while in this staging area as well. The push server software uses the information provided in the staging area to create deliverable packages. Other non-scheduled, publishing content is created by content providers using a publishing tool. An example of such a publishing tool is one that scrapes a web page that already exists.

3. The information from the repositories as well as published content is downloaded to the client whenever the agent determines that new content for a subscribed channel exists.

2.5 Experience is the Best Teacher

We've learned much in the process of creating the M-Web solution that was not obvious from the start. Here we share our advice for the benefit of someone considering implementing a similar solution.

*Understand how to use Push Technology effectively***.** Understand performance and feature usage trade-offs as well as the impact to the user. For instance, frequent notifications can be a hindrance to users. Most of these issues revolve around content publishing. Content publishers should be trained to understand performance and feature usage implications.

Keep the content flowing. Once a solution is deployed, users will ultimately judge your application on the usefulness of content it offers. Subscription information can quickly become outdated if new content is not generated periodically. If Uniform Resource Locator (URL) hyperlinks are published, make sure that they are validated routinely and verify that the owner understands the consequences of changes. Don't underestimate the amount of on-going content related work. A knowledge/content strategy is a must.

Know your end user's mobile environment and workflow process. This can be quite a challenge when deploying worldwide. Regional representatives and user groups can validate your solution's integration into local/regional processes.

Ensure that the organization of information and channels is simple and intuitive. Time to find relevant information after content delivery is just as important as the delivery automation. Users often have difficulty subscribing to content. Simple and intuitively organized channel layouts can improve a new user's experience and enable users to find information after it is delivered. The relationship between channel layout and the actual data-browsing model must be clearly presented and defined.

A critical mass of features can be key to user acceptance. Look for improvements in the total solution space. For instance, keywording information after content delivery may add value in a knowledge repository. Provide integration with other applications or useful desktop features.

For more detailed advice on how to evaluate a push technology offering relative to the needs of your application see Appendix A: Push Technology Evaluation Criteria

3 SmartPhone

SmartPhone is a WAP-based application that delivers work assignments, customer information, and parts information to a mobile phone and allows the status of a work order (referred to as a Customer Service Order (CSO)) to be updated through the phone. SmartPhone is currently deployed in Europe and Asia.

3.1 Problem Statement

As mentioned earlier, CE's spend much less time in the office and more time on customer sites or in transit from one customer to the next. The service call dispatch process requires that CE's have network connectivity to the HP intranet to receive their work assignments. CE's need up-to-the-minute information about these service calls. One way to achieve this is to deliver this information to their mobile phones. CE's not only need view to service call data in real time, they also need the ability to update portions of this real-time data.

Providing a wireless mobile dispatch and reporting process increases the efficiency and productivity of our field service engineers. This translates into our customers experiencing a faster turn around time on any given support call.

3.2 Our Solution

A user with a mobile phone starts the application by selecting the SmartPhone Application link that is book-marked in their microbrowser. The home page allows users to authenticate themselves in order to be authorized to use the system. (**Figure 5**A) Once authorized, the user is given a list of options that can be delivered to the mobile phone.(**Figure 5**B) Currently, the primary application supported is workflow, and all the example screen shots in **Figure 5** show workflow. A CE can request the list of service calls that have been assigned to him/her (**Figure 5**D), view the details of a particular service call (**Figure 5**E,F), update their status indicating whether they are available to be take a new service call, in transit, on site, etc. (**Figure 5**C). The timely update of this information is important for efficient resource management. Finally, service call information can be entered through the phone. (**Figure 5**G,H,I)

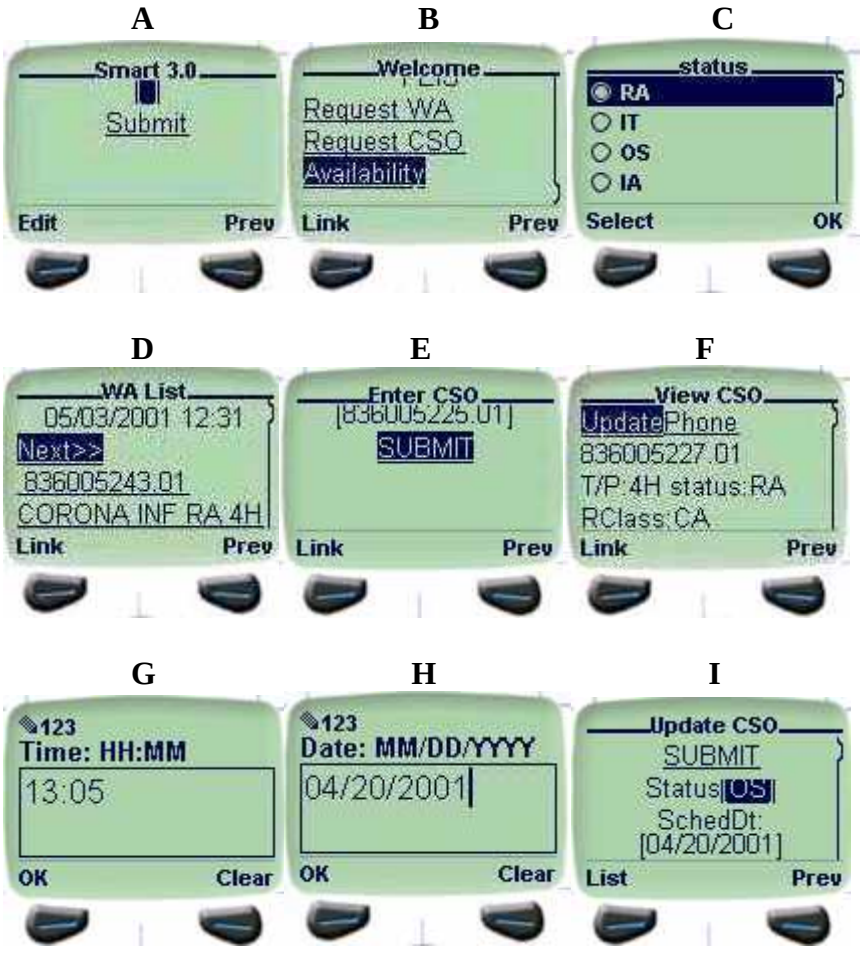

Figure 5 SmartPhone User Screens

The SmartPhone application provides several benefits to HP. One benefit is the ability to automate the workflow process and receive current information both at the CE's phone and at the HP call processing center. SmartPhone realizes all the benefits of any automated workflow system but has the additional benefit of delivering timely workflow information to a mobile professional much sooner than was possible before.

Receiving real-time work assignments allows the dispatch center to reprioritize calls in real time. Thus, the CE always knows which is the most important call to take when it is time to plan the next service call. Similarly, CE's can get the most up-to-date status on the service call, making the repair more efficient (they do not have to duplicate effort already spent or replicate information already known). This reduces the time to fix the problem, which increases customer satisfaction and enables the CE to handle more service calls in a day. Not having to visit the office just to establish a network connection for retrieving work assignments also saves time.

Allowing CE's to update service call information immediately upon finishing a service call, rather than waiting until the end of the day when network connectivity can be obtained in the office, means that the information will be more accurate, especially for documenting what was done and how much time was spent. Having this data updated in the backend workflow repository gives the call processing center a more immediate view of the status of the service call and of the CE.

The above benefits were all about the immediacy of workflow information, but SmartPhone's benefits are not limited to workflow. We will use SmartPhone to deliver up-to-the minute information on parts including current inventory, status of monitored customer sites, or any other time critical or real-time information that would make the customer support process more efficient.

3.3 Understanding WAP

Before delving into how SmartPhone is implemented, we first share tutorial information on WAP. Those already familiar with WAP can skip this section and continue with Section 3.4.

WAP is a worldwide standard, providing Internet access to an ever-increasing array of wireless devices. The WAP forum [3] defined and released the WAP specifications, and there are currently three versions: 1.0, 1.1, and 1.2. Because WAP is both a protocol for communications and an application environment, implementations exist on a host of different operating systems including Windows CE, JavaOS, EPOC, and PalmOS. In addition, WAP has been designed to work with many different wireless networks including but not limited to CDMA, CDPD, DataTAC, GSM, PDC, PHS, TDMA, FLEX, ReFLEX, TETRA, DECT, and Mobitex.

The WAP standard is similar to HTTP in many ways, and in fact, it was based on several other standards including HTTP as well as HTML, TCP/IP, SSL, and Mime. As a result, we find that WAP and HTTP transactions behave similarly. In Figure 6, we see the data flow of a WAP request. The Client User Device, which is the microbrowser running on the mobile phone, creates a WAP request and sends it to the WAP gateway. This request is compressed and specially encoded to decrease bandwidth requirements and increase performance on the wireless network. This binary encoding is one of the main differences between WAP and HTTP.[1]

[1]The protocol part of the WAP standard, called WSP (wireless session protocol), is a compact binary form of http.

The WAP gateway is a protocol converter between WAP and HTTP. It uncompresses and decodes the WAP request, then translates the request into HTTP. It sends the HTTP request on to the intended web server. The web server responds to the HTTP request by returning either static or, in the case of SmartPhone, dynamic Wireless Markup Language (WML) web pages. Using WML a subset of the Extensible Markup Language (XML), requires the web server to support the WML MIME types. The HTTP response is received by the WAP gateway and translates the response into a WAP response, encoding it and compressing it appropriately. When the microbrowser receives the WAP response, it defines the formatting that will take place and presents the data to the user.

The similarities between WAP and HTTP behavior enables those that currently understand web technology to apply this understanding within the WAP environment. Existing tools can also be used to build WAP applications.

Why use WAP rather than HTTP you ask? Within the wireless world we find that HTTP and TCP/IP, on which HTTP depends, are much too complex for meeting the needs of the wireless network. A large amount of overhead exists within these standards that is just not required when considering the wireless environment. For example when using TCP/IP, numerous messages must be sent in order to establish a connection, this consumes additional bandwidth, which is a very limited resource in the wireless network. TCP/IP places additional demands on the processor as well; when considering the limited processor capability of wireless devices today this becomes a real performance concern. WAP was designed to provide a low overhead protocol so that the needed protocol engines could run in small devices with limited hardware resources such as a mobile phone.

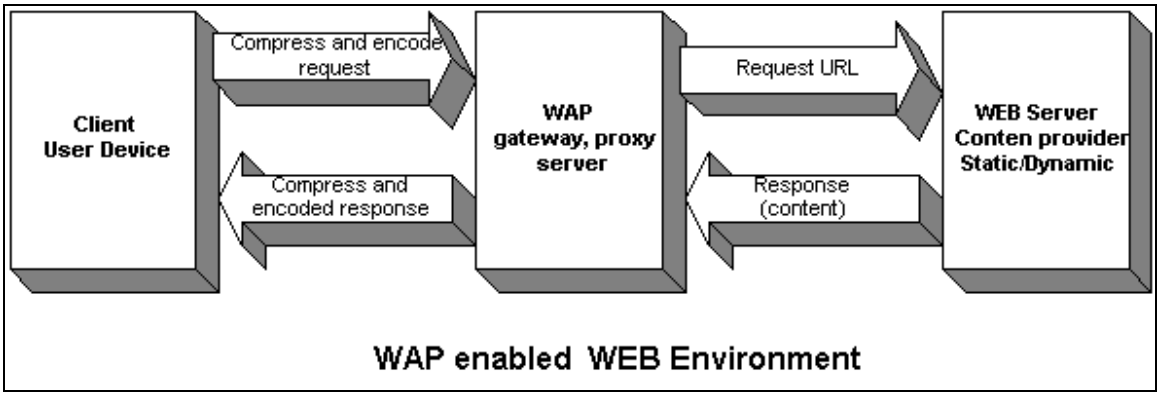

Figure 6 WAP enabled web environment

WAP, however, is not without its limitations. First, as mentioned above, the WAP gateway introduces latency as it compress/uncompresses and encodes/decodes to translate the WAP protocol into HTTP and vice versa. An HTTP web proxy, where http is used on both sides and no protocol conversion is needed, does not introduce this latency. Second, it should be noted that WAP has not been optimized for best performance or reliability with any of the networks on which it has been implemented. The reliance of WAP on these underlying wireless networks renders WAP applications slow and limited in coverage. Many of these limitations and constraints are being addressed and as technology moves forward these improvements will make the wireless

environment even more attractive. Please see the following web site for information about these standards http://www.wapforum.org/what/technical.htm.

3.4 How SmartPhone Works

SmartPhone uses WAP to communicate between a user's mobile phone and the web server inside the HP firewall where the SmartPhone application runs. Internet Service Providers local to the countries where SmartPhone is deployed provide a Global Systems for Mobile Communications (GSM) network over which WAP runs. All the application-specific parts of the application run on the internet side of the WAP gateway. Figure 7 provides an overview of how the end-to-end SmartPhone application operates. The overall data flow is presented along with the various components. SmartPhone is implemented based on the WAP 1.1 standard.

At the phone, one configures a bookmark, which is the URL of the SmartPhone home page. This homepage is implemented using Active Server Pages (ASP) pages located on a web server. The ASP pages create dynamic content in the WML format using information obtained from the application server.

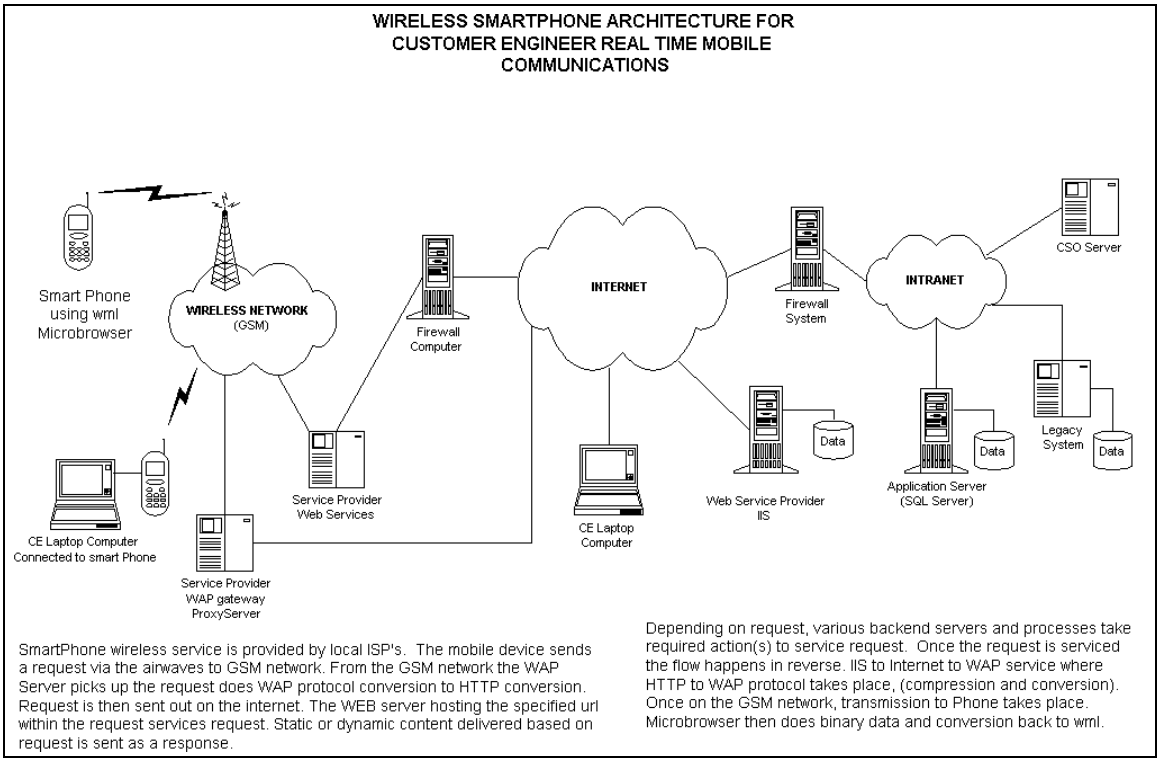

Figure 7 SmartPhone Architecture

The application server runs a database management system (DBMS). This server hosts the database management system and associated databases (one per country), consisting of stored procedures,

triggers, and tables comprised of both static and dynamic data relating to each country. The ASP pages issue SQL commands against the appropriate country's database. Based on the type of request, stored procedures are executed. From here the formatted transactions are passed along to the CSO Server, which is responsible for getting or setting the desired information to/from the legacy database system.[2] The CSO Server sends the formatted data back to the country database where it is cached. If successful, this updated, cached data contained within the database for which the request was originally made is returned to the web server via stored procedure(s).

The ASP pages use the information obtained from the database to generate a WML page that is returned through the WAP gateway, back across the GSM network, to where it finally arrives on the phone where the microbrowser renders and displays the WML content for the user.

Figure 8 shows a more detailed view of the backend processing that comprises the SmartPhone application.

[2] It should be noted that access to both legacy and non-legacy systems is possible.

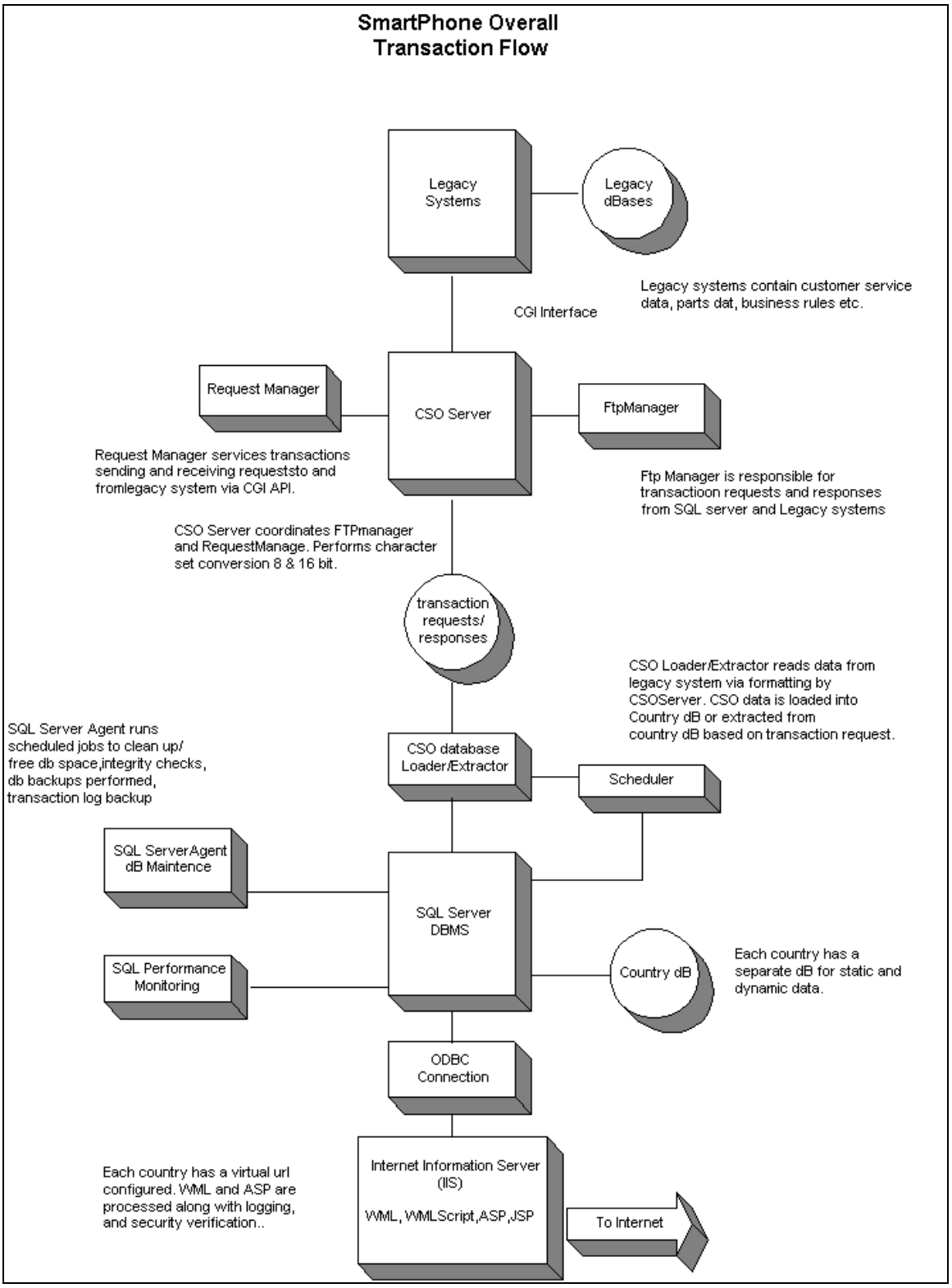

Figure 8 SmartPhone Transaction Flow

3.5 What We've Learned

As with any worldwide deployment many concerns and issued must be addressed. This section presents some lessons we've learned that might help someone getting started with their first international wireless application.

Internationalization is crucial for any application targeted for worldwide deployment. This requirement isn't unique to wireless applications, but given the number of systems and infrastructure software involved in an end-to-end wireless solution, internationalization can be a real challenge. Every piece of software that touches the transaction must be able to appropriately handle international character sets, local time/date and currency conventions, *etc*.

Each phone's microbrowser renders a WML page in its own way. Thus the same WML page can look very different depending on which microbrowser is displaying it. Similarly, an application that behaves properly on a phone simulator will not necessarily work as expected on the real phone that was being simulated. Allocate enough testing time on the actual handset that will display the application user interface. Different firmware revisions on the same model phone can produce inconsistent application behavior as well.

Working with a mobile phone as a client is more constraining than other clients such as laptops or PC desktops. Designing a usable application interface is challenging given the small screen size and lack of a standard keyboard for text input.

.

Independent Service Provider (ISP) infrastructures are different; these differences can produce surprising and unexpected results. For example, configuring the WAP gateway properly can be difficult, and typically the application developer does not have access to this configuration. Some ISP's even outsource portions of their wireless infrastructure, sometimes including the WAP gateway! Thus, it might be confusing and difficult to troubleshoot when an application works fine with one service provider but doesn't work with another. The difference could be in the WAP gateway configuration.

4 Future

We plan on, and actively work towards, taking advantage of emerging technologies to enhance our currently deployed tools. In this section, we discuss several such enhancements:

- Consolidate the multitude of appliances that mobile professionals must carry.
- Reduce the dependency on keyboard input.
- Applications should tolerate sporadic connections, where necessary, and take advantage of connectivity where possible. When the state of connectivity changes, the application should handle this without requiring manual intervention by the user.

4.1 Device Consolidation

Many mobile professionals today carry a mobile phone, PDA, pager, and laptop. The SmartPhone user interface is offered on the mobile phone that CEs already carry for voice communication purposes. M-Web runs on a laptop that CE's already carry for running diagnostics and interacting with the full workflow user interface.

As these appliances become more powerful and multipurpose, fewer separate devices will need to be carried. PDAs are becoming available that integrate telephony, paging, and computing in a light, handheld-size package. Thus, if we can get all the needed mobile applications to provide a user interface on this new device, it will be much simpler for the mobile worker (especially when you consider the separate charging and/or batteries for each device as well as separate phone numbers and user identifiers).

The issue is whether a highly portable, functionally integrated device such as a PDA can reasonably provide the user interface for all tools including SmartPhone and M-Web. SmartPhone can take advantage of the larger screen size on the PDA relative to the mobile phone screen it currently uses. The challenge to SmartPhone is to move the application to an infrastructure that can support a variety of handheld devices, not just WAP-enabled mobile phones.[3] Most modern application servers provide the ability to separate the application logic from the presentation so that the same content can be delivered to a large set of devices each with a different set of browser capabilities. Modern mobile portals provide wireless connectivity over a variety of wireless protocols to the application without having to recode the application. Moving to an infrastructure including a mobile portal and an application server will not only enable SmartPhone to be used by all different kinds of mobile devices, but will enable the easier development of new mobile applications as well.

There are at least two important challenges to making a PDA a viable M-Web client: lack of storage and screen real estate. Eventually, PDAs will have integrated minidisks that will store a large portion of the needed technical reference information. As a result, in the longer term, the storage provisions on the PDA will approach that of laptops running M-Web today. However, today's PDAs don't provide as much storage as a laptop, and thus, less reference information can be carried around. A medium term solution that combines the portability benefits of the PDA with the local

[3] PDAs tend to have HTML rather than WML browsers.

storage space of the laptop is to use a hybrid solution. Rather than replacing the laptop completely with the PDA, the PDA becomes an adjunct to the laptop obviating the need to carry a laptop onto a customer's site. The laptop becomes a mobile caching server. Only the information needed on the current service call needs to be downloaded into the PDA, and this can happen at high speeds using short-range wireless technology such as Bluetooth or 802.11. The other challenge is that the PDA screen size is smaller than the laptop's, making it more difficult to read manuals from a PDA screen. We can compensate for the small screen by a combination of search mechanisms to find the fine-grained nuggets of needed information and the ability to print out just the needed pages on a nearby printer. This ability to print content sent from an appliance over a wireless link without having to install device drivers or in any way configure the printer *a priori* has existed with jetsend in the past. A new print-by-reference specification is under development in the Bluetooth forum that would allow an appliance to send a hyperlink to content that exists on the web. The internet-enabled printer would first get the content, direct the rendering, and finally print the rendered output.

There are other capabilities that come integrated or as add-ons to a PDA that are useful to our mobile workers. First, a barcode scanner can be used to automate the interface to parts. There are many applications, including workflow and parts logistics, that require the input of a part number and/or serial number. This information can be entered without typing, by scanning the barcode on the part itself. Second, a digital camera can send an image back to the factory for consultation. Eventually, biometric and environmental sensing devices will be commonly available that will ensure that the device can only be used by an authorized person for an authorized purpose.

So far, we've only mentioned the use of the PDA as a data device, but with integrated voice capability, it also offers the possibility of offering voice-based user interfaces, a feature not currently exploited by SmartPhone.

4.2 Voice-Based Interfaces

Voice-based interfaces can greatly benefit a mobile professional. Spoken commands can enable interaction with an application in hands-free mode. This is useful not only while driving but also takes away the pain of keyboard input on small keypads. Voxml is a voice markup language that allows an application's content to be presented as synthesized voice. An application server supporting voxml can deliver an application's content to a device with a voxml browser that renders the output as speech.

The challenge of entering text on a small or non-existent keyboard is especially profound when needing to enter free-form text fields such as the cause of a problem or the description of the repair. It is possible to simply digitize a voice clip describing the cause or fix and attach the audio file to the CSO for processing in the backend system. Although today's PDAs are not powerful enough to do really good speech to text translation, the backend systems are powerful enough to do a reasonable job at transcription (though it might need to be manually edited later on when there is access to a full sized keyboard).

In addition to using voice for data entry, voice synthesis also can be used to render email or other messaging communication as speech. Whereas many mobile professionals are already quite accustomed to listening and responding to voice mail while driving, the same can be done with email. Therefore, the same device that provides a user interface into workflow, parts availability, and technical reference information can also be used for mobile communication.

4.3 Wireless Connectivity

Today, mobile phone connectivity is not uniform worldwide. The GSM phones that work in Europe don't work in many places in the United States. The US CDPD phones don't work in Europe. Even nationwide US mobile carriers don't reach every area, especially rural communities. In the short term, before such connectivity is ubiquitous, strategies are needed for supporting sporadically disconnected use. For workflow applications, it is important to update status as soon as a connection can be established.

However, we look to the future when mobile phone connectivity is ubiquitously available worldwide and offers a single standard. GPRS is likely to be the first such widely supported protocol. The fact that GPRS is packet switched rather than circuit switched is important for several reasons. First, GPRS users always have a channel available for them. The billing is not time based, but rather, traffic based. In the past, since it was too expensive to keep a phone line open all day, the mobile worker has to dial in and authenticate each time there was a need to communicate with the application. With traffic based billing, it becomes feasible (and cost effective) to authenticate once at the beginning of the day, and have continuous connectivity with the application throughout the day.

Because there is a continuous virtual connection between the mobile appliance and the application, it also becomes feasible to push timely information to the appliance rather than waiting for the appliance to request or pull information. This supports our desire to have real-time messaging, and can be leveraged by M-Web to realize the original vision of push technology.

Another benefit of continuous connectivity is that it provides a safety net for requesting technical reference information on demand. We expect the common case to be that M-Web will provide needed information locally on the PDA, but we must address the corner case that there will be unanticipated needs for information. With constant connectivity, the needed document can be downloaded on demand. Furthermore, constant connectivity will mean that flash notifications can be delivered instantaneously rather than having to wait until the next time a network connection might be established.

One final benefit (or curse) of using GPRS is that it can track the location of the user. There are obvious privacy concerns with this, but there are corporate benefits of the availability of location information. First, many companies already install GPS systems in their fleet vehicles for the purpose of locating them. This information is useful for logistics, especially for dispatching. Most customers would prefer to have the dispatched CE be the one that is already closest to the customer’s location so they don’t have to wait for the one across town to travel to their site. With

GPRS appliances, is it not necessary to equip each vehicle with GPS hardware. The vehicle operator will be carrying the locator device in the vehicle. The second benefit of having location information is the ability to provide location-based services. It allows the mobile professionals in an unfamiliar area to find the service establishments closest to where they are.

4.4 Summary

We live in exciting times. Appliances, software, and wireless networks are becoming smaller, faster, more functional, and more ubiquitous. This allows the development of very sophisticated mobile applications. Our experience in deploying present day mobile applications enables us to quickly take advantage of these opportunities to make our mobile workforce even more effective and efficient, thus reducing a customer's downtime.

5 Conclusion

Mobile solutions can offer a wide range of benefits and uses for businesses. Understanding the technology and its current issues as well as the problem set is a requirement for successful development and deployment of these solutions. We have documented our experience with building two mobile applications currently in use within the Hewlett-Packard Company.

Most mobile solutions will evolve to integrate with new technology, services and information systems. An attempt to foresee the future impact of new technologies and services will allow for better decision making today in designing, changing processes and managing information for the future. Organizations that are prepared to take advantage of these advances will benefit from new opportunities as well as improvements in efficiency.

Our solutions, M-Web and SmartPhone, have proven to be valuable to our mobile support workforce by increasing their productivity, reducing call closure time, and improving field logistics scheduling. All of this helps in quickly resolving our customers' support calls.

6 Acknowledgements

Everyone in our lab contributed to the content of this paper in some way. We'd also like to acknowledge the help we received from our many reviewers who made this paper much better through their constructive suggestions: Roy Clifton, Jamie El Fattal, Jan Marks, Robert O'Dell, Jim Privat, and Linda Tarby. In addition, thanks to Zahid Khan for supplying us with the user screen shots of the SmartPhone application. We owe you *all* big time!

7 Resources

[1] http://www.backweb.com

Push Technology Vendor

[2] http://www.marimba.com

Push Technology Vendor

[3] http://www.wapforum.org

Site of WAP forum whose members developed the de facto standard for wireless devices.

[4] http://www.openware.com

Phone.com joined with Software.com to form openware.com.

[5] http://www.wirelessinanutshell.com/wap

Great site focused on wireless.

[6] http://www.ericsson.com

[7] http://wapsight.com

Site covering latest breaking news in wap/wireless.

[8] http://www.nokia.com/corporate/wap

Great site for information and solutions regarding wireless.

[9] "Nokia WAP Toolkit 2.1" Nokia Corporation

[10] "Nokia WAP Toolkit 2.1 User's Guide", Nokia Corporation

[11] "Nokia WAP Toolkit 2.1 Developer's Guide", Nokia Corporation

[12] "Deploying Microsoft SQL Server 7.0 Notes from the Field", Microsoft Press 1999

[13] http://www.gsmworld.com

Site devoted to GSM.

[14] http://mayweb1.mayfield.hp.com/bist/security

HP security standards and information (info on HP-UX Servers and NT).

[15] http://www.microsoft.com/sql

Product information on SQL Server7.0

[16] Keith Neuhofer, “CE Communications”, Revision 2.3, March 1, 1998

[17] “Enabling The Wireless Internet”, February 1999, Unwired planet

[18] Jeffrey McManus, “Create a Database App for the web”, Visual Basic Programmer’s Journal 6/98 , Volume 8, Issue 7, Fawcette Technical Publications

8 Appendix A: Push Technology Evaluation Criteria

Because self-developed push technology infrastructure and tools will be too costly, most application developers will choose to use a third party software package for this infrastructure. It is especially important to choose wisely among the available software packages. There are companies that only specialize in Push Technology. There are other companies that provide Web Application Servers that are trying to incorporate Push Technology into their feature set for increased product differentiation and well as to fulfill customer needs in this ever-growing mobile market.

Here is a list of categorized questions that should be addressed when evaluating for push technology products for suitability for your application.

Features	Questions that need to be addressed
Operating Platform for Server and Client	• Does the server software run on NT, W2000 or a flavor of UNIX or LINUX? • Are there client agents available for Windows CE, PALM OS, Windows OS systems?
Publishing Tools, Features, Process	• How flexible are the publishing tools? Can it support automated monitoring and publishing if a web page is being pushed and it changes? Are there publishing wizards? • Can publishing responsibility be distributed to a group of publishers? • Is there a content approval method integrated into the product? • Can information be targeted to a group of end users? • Aside from publishing information, can publishers create a new

	subscription category?
Notifications	• How are users notified of new information? • Can notifications via pop-up dialog ("flash") be configured in terms of display interval, number of times to display, deletable, etc.
Programmatic Hooks/APIs	• Custom solutions will require these interfaces. How much control does the software allow to place information outside of its infrastructure, especially on the client end? • What kinds of processing are allowed on the user end? • Can executables be delivered and invoked? • Can data be collected and delivered to the server?
Out of the Box Configurability	• For simple solutions, how easily can an out of the box client be customized – look and feel, removal of unnecessary features, adding a company logo etc.
Additional Software Requirements	• Does the push software require a database package? If so, what does it support? • What about a web application server?
Performance	• How is server load balancing handled?

	• Can servers be replicated for large deployments? • How intrusive are the desktop agents when information is delivered? • When published information changes, how much of it is re-delivered? • Understand the performance factors such as number of users, number and size of published information nuggets, number of services per channel, etc. • What types of communication protocols are supported?
Content Related Questions	• Are there limitations to size, type and number of information nuggets? • What is the correlation between content grouping and channels? • How granular do you need to make the subscription? For instance, in our case, it is by product line, product number, etc.
Cost	• Is there a license charge for agents? • What's the price of the server software?
Product Updates	• How are product updates handled? • Can the agent updates be pushed and automatically installed?

	• How difficult are server upgrades?
Recovery	• What is their disaster recovery plan? How would end users get impacted?
Reporting Features	• What kinds of statistics are being tracked? This could be extremely useful when determining areas for improvement and focus.
Support	• Is support available? • What does it cost? • How responsive is their support? • Is customer consulting available if needed?
Training	• Is training available? • How much does it cost?

Other factors that will determine successful use of Push Technologies depend on the business itself. Answers to the following question will determine the amount of effort required to successfully create and deploy a solution.

Business Knowledge Repository	• Are the information nuggets to be pushed readily available and in the proper format? • Are there many similar knowledge repositories that need to be merged? • How much data is there? • Can information be maintained and updated after solution deployment? • How do structural and information changes affect the solution and users? • Will the current business process get impacted with this solution?

User Skill Set and Operating Environment	• Are the mobile devices currently being used sufficient to handle another tool? • Is the skill set of the users addressed for the solution offered? • What is the medium for internet/network connection? • Does the solution fit the user's process in terms of information needs, when it is required, and what is required in order to receive and utilize the information?
General	• What is the impact of future changes to the repository, backend infrastructure, etc? • How does the push technology infrastructure fit in with other systems/infrastructure such as a web application server? • What's the cost of being dependent on a particular software vendor? Will it be around for long? • What's the gain? Cost savings, time to market savings, improved service, etc? • Is push technology part of the answer for your problem?

9 Appendix B : GLOSSARY

Dual Band – In the wireless world normally refers to handset that is capable of operating on two different frequencies.

GPRS-General Packet Radio Service provides tenfold increase in bandwidth, users are connected and always on. When compared to GSM this service only uses the network when data is sent.

GPS – Global Positioning System, satellite based radio-positioning system.

GSM-Global Systems for Mobile Communications international digital standard used in more than 120 countries.

GSMS-1800 -Europe standard for 1800Mhz and 900Mhz cellular operation.
US PCS operates at 1900MHz

Microbrowser –Enables handheld devices to access the Internet.

PDA – Personal Data Assistant

SIM – Subscriber Identification Module, which contains authentication information and GSM encryption routines.

WAP- Wireless Application Protocol is a standard for the presentation and delivery of wireless information and telephony services on mobile wireless devices.

WAP Portal – web page with many links to applications that are organized by category.

WML – Wireless Markup Language developed for WAP, tag-based language which is a subset of XML and similar to HTML.

WMLScript – scripting language that extends the power of WML, providing programming structures such as *If then else*, looping etc., and is used for programming mobile devices.

WSP – Wireless Session Protocol that manages the session within the WAP protocol stack.