The importance of embedded software on mobile handsets is growing

The functionality of mobile phones has expanded dramatically in recent years. No longer constrained by small monochrome displays and rudimentary textual browsers, most mobile phones have become sophisticated computing devices, with power and memory rivaling that of personal computers of the late 1990s.

Along with high-resolution color screens, mega-pixel cameras and other advances in hardware, today’s mobile handsets are equipped with powerful software, enabling rich user interfaces and sophisticated features and applications. Many features have been tightly integrated with the embedded software on mobile handsets, including: browsers, multimedia messaging (MMS) and instant messaging clients, Java virtual machines, digital rights management, e-wallet, 3D graphics, video telephony, video streaming, location/GPS, Bluetooth/WiFi, mobile TV, push-to-talk clients and music players. As a result, the amount of embedded software in mobile handsets has grown tremendously (see Figure 1).

Integrated mobile software is critical to reliability

With the heightened sophistication of mobile devices and increased software on the handset, operators are facing the challenge of ensuring reliability to their subscribers. To ensure that their services perform as specified, operators require that handset manufacturers deliver a compelling suite of optimized, well-integrated and built-in handset features. This is typically captured in embedded firmware, stored as files in a read-only file system, a monolithic firmware image or as a combination of the above. Core handset applications and middleware components are carefully integrated into this embedded firmware. Embedding the core feature set ensures a level of reliability, security and performance that can only be achieved through native, close-to-the-chipset implementation, secured in a read-only memory area. The embedded firmware image also provides for

Figure 1: Exponential Growth of Embedded Software, Source: Red Bend, industry data

1. Operators are taking on more responsibility for the design and development of handsets. They are mandating handset requirements to the handset manufacturers, building custom applications and even designing their own handsets using white label original design manufacturers (ODMs). This trend is especially apparent in Japan, where operators mandate that their handset suppliers deliver phones with a specific set of features. Along these same lines, the Open Mobile Terminal Platform (OMTP) organization consists of a group of operators trying to create commonality in handset software platforms.
tight integration across software layers and dependency control needed to provide the high degree of interconnectivity and interoperability between functions that are required.

An embedded feature set provides operators with better control over the produced handset and thus allows them to better control the user experience. This makes it easier for operators to ensure a consistent set of services while projecting their unique brand identity and differentiating themselves from competitors.

**Integrated embedded firmware presents challenges**

Despite its advantages, integrated embedded firmware, especially in the case of a monolithic image structure, also presents challenges to operators. Once the phones are in the hands of consumers, it is difficult to change the firmware. As operators develop new services, the firmware in their subscribers’ handsets may not support those new services. Operators may be unable to “break open” the monolithic firmware entity to fix defects or to upgrade, modify or add software service enablers to many of their installed base of handsets.

The fact that there are many variants of the firmware for each handset model creates a logistical maintenance problem for handset manufacturers. OEMs produce many different firmware versions in order to meet their customers’ diverse needs. In some cases, there are as many as 200 different versions – from a generic version to an operator custom version to a country-specific version.

The need for integrated embedded firmware, in its many variants, also presents a time-to-market problem and can delay new handset introductions. Design, integration, test and build cycles involve a large number of interdependent third-party components and operators’ custom feature requirements. The lead time from developing requirements to releasing a handset can take up to 18 months.

**New requirements for managing embedded software components**

Today, operators and handset makers can make post-factory modifications to the core handset software on a mass-scale through firmware over-the-air (FOTA) updating. FOTA is widely used by leading handset makers and operators, primarily to deliver software improvements and to fix defects that are discovered after handset shipment.

FOTA is the most efficient and convenient solution for updating mobile firmware in the field. However, the technology was designed to handle the firmware as a monolithic entity. In order to repair a specific defect or add a feature, the entire firmware version must be updated. This requires that handset makers re-create every handset variant’s firmware version in order to incorporate the modification, and then conduct extensive testing on each of these versions before an update is generated and delivered. The result is a versioning and management bottleneck (see Figure 2). Although this process may be appropriate for critical maintenance releases or major enhancements, it is not an effective solution for more frequent, smaller updates.

![Figure 2: Monolithic (FOTA-Based) Firmware Updating Process](image)

2. The FOTA updating process is based upon applying “diff” or “delta” algorithms, which compute the difference between old and new firmware versions. This in-turn, allows a small update to be sent over bandwidth-constrained networks and securely re-flash the handset. OEMs and operators use FOTA to reduce service costs, avoid product recalls and increase customer satisfaction and loyalty by providing consumers with the latest software improvements and features.
Managing embedded firmware with Software Components Over-the-Air

A new firmware update technology is being evolved to enable manufacturers and operators to manage individual software components over-the-air (SCOTA). SCOTA allows deployment of discrete software components, either updated or new, to the embedded firmware of a mobile device. A modification or feature change in a software component requires only a single updated version of the component to be deployed to all relevant handset configurations in the field. A completely new feature can be deployed and installed to applicable handsets, and previously installed features can be removed to make space for other features. With SCOTA, operators can maintain the reliability and security of tightly integrated firmware, but they gain more flexibility to efficiently update devices post-sale.

SCOTA focuses on the handset’s built-in software components and applications

SCOTA differs from standard application/content delivery, such as ring tones or J2ME applications, in that it enables the modification of the handset’s built-in, embedded software. The main difference between SCOTA and FOTA is that while FOTA treats the entire firmware as a single monolithic entity, SCOTA treats the firmware as a set of individual components, so that modifications can be made at a component level (see Figure 3).

SCOTA delivers benefits to smartphones as well as feature phones

The software maintenance flexibility created by SCOTA is particularly valuable to mid-range feature phones with a “closed” operating system (OS) whose entire core feature set is embedded within a monolithic firmware image. However, it can also benefit open OS smartphones, such as Symbian-based handsets. Smartphone software is designed and built in an advanced, modular fashion and provides capabilities for post-sale installation of fully featured native add-on applications. However, its core functionality, critical service-enabling applications, middleware components and the OS itself are still embedded into read-only flash memory. SCOTA enables the efficient and modular management of core smartphone software.

SCOTA achieves new flexibility in Mobile Software Management

SCOTA is an important enabler for Mobile Software Management (MSM). The ability to remotely provision and manage device software applications is being addressed by standards organizations such as the Open Mobile Alliance (OMA) and industry initiatives such as Open Mobile Terminal Platform (OMTP). These groups are working to create frameworks and protocols to support software management and related issues. One key enabler for SCOTA is a developing standard called Software Component Management Object (SCOMO) [see Sidebar].

While previous software management technologies focused on high-level Java or open OS smartphone applications, SCOTA
extends management scope, flexibility and control to the core, embedded part of the handset software, which contains all the critical applications and middleware. SCOTA uses the same abstraction mechanisms and protocols used for managing high-level software.

SCOTA benefits handset manufacturers and software platform providers

For handset manufacturers and software platform providers, SCOTA makes software maintenance more efficient. Component-level software maintenance resolves the problem of managing the wide variety of firmware versions. Using SCOTA, there is no need to link, test and deploy a large number of firmware variants for every software release or feature change. A single component can be deployed to all compatible handsets regardless of their actual underlying software configuration.

For example, if a defect is identified in an embedded component such as the terminal XML parser, that defect will likely exist in many different handset model variants and even within different OEM models based on the same software platform. In order to repair those handsets in the field using FOTA, the handset maker would build multiple new firmware versions and generate multiple firmware update “delta” packages, one for every firmware variant that contains the defect.

Using a SCOTA-based process, the fixed component is handled as an individual entity. This makes development, testing and deployment much easier and quicker. A single update package can then be deployed to all handset model variants shipped with the defective component.

In addition to saving time and money in issuing maintenance releases, handset makers can also use SCOTA to deliver new or upgraded features to mobile devices. They may, in fact, choose to charge operators for a feature upgrade service for its supplied handsets. Or they may share revenues with the operators from feature upgrades and new feature sales.

SCOTA benefits independent software vendors (ISVs)

Embedded software components, such as applications [browser, messaging client] and middleware components [Flash Lite, Java Virtual Machine] are often provided to handset manufacturers and platform vendors by third party software suppliers. The manufacturers then integrate these components at handset production. This traditional mode of operation demands long lead times and is strictly determined by the handset release schedule. At the same time, software suppliers constantly evolve their software products and develop enhancements and new features. Many ISVs see a benefit in being able to update their embedded software on handsets in the field. Unfortunately, the ability to perform such updates is very limited. The manufacturer may periodically release a new firmware version that incorporates the software update along with other firm-

Enabling Standards

A new proposed extension for the standard OMA-DM protocol for managing individual software components in mobile phones is launching a new generation of mobile software management capabilities. The proposed standard, called Software Component Management Object — or SCOMO for short, is being developed through the Open Mobile Alliance Device Management (OMA-DM) Working Group.

SCOMO defines new management functionality to be performed by DM systems and mobile phones. The result will provide device manufacturers and mobile operators with greater flexibility and control over managing a device’s software throughout its lifecycle. With this new management capability, DM systems will be able to:

- Deliver and install new software components on the device
- Update existing software components
- Remove existing software components
- Query the inventory of software components, and
- Activate or deactivate software components

The standardization of SCOMO, expected by the end of 2007, is part of several significant advancements in mobile software management that will address the growing need to manage all of a device’s software after the phone has shipped. For most mobile phones, it is not possible to update or change individual embedded software components post-sale, unless performed as part of a complete firmware update, known as FOTA (Firmware Over-the-Air). The technology to manage embedded software components on the devices combined with server-client interaction standardization provided by OMA-DM and SCOMO will open up the entire device assets for individual customization and management.

For more information on SCOMO and other device and software management capabilities, visit the OMA DM web site: www.openmobilealliance.org/tech/wg_committees/dm.html
SCOTA provides increased flexibility for software suppliers. With SCOTA, ISVs can realize the following benefits:

- Shorter lead times for handset integration as enhancements can be made post-factory
- Better service to their OEM customers and eventually to consumers
- Periodic software feature updates to their embedded software component that enable ISVs to achieve recurring revenue
- In some cases, and with appropriate security processes in place, ISVs can deploy a completely new component over-the-air, which may provide them with new revenue streams

SCOTA benefits mobile operators

For mobile operators, having greater flexibility to modify the core features of the handset after it has been shipped opens up many valuable opportunities. SCOTA can be used to improve the reliability of handsets on their network, deploy custom features, customize phones at the point-of-sale and sell software features directly to customers.

Some examples illustrate the value:

- Operators can enable consumers to download more sophisticated games and applications by upgrading a middleware component such as a Java API or upgrading to a higher J2ME MIDP version
- By downloading and installing a media codec, consumers can subscribe to a new music service, view a better quality video clip or view received images using a previously unsupported content format
- Camera applications can be upgraded to include a new digital zoom function
- Operators can upgrade the phones on their network to include a custom client portal or a newer Flash engine, which could support a new set of services
- Handsets can be upgraded to support Voice-over-IP (VoIP) or Session Initiation Protocol (SIP) stacks or to repair defects encountered upon launching IMS based services
- Operators can use SCOTA to customize language support at the point-of-sale by downloading and installing appropriate language packs

While in principle most of these activities can be accomplished using the FOTA-based firmware updating process, SCOTA provides a faster update, the user experience is better, and frequent updating and installation is both possible and manageable at a mass market scale.

SCOTA allows operators to enhance revenue

SCOTA offers operators opportunities to enhance service revenues. Features can be offered to consumers for a one-time fee or through a subscription service. Or, by upgrading capabilities within handsets, operators may then market new services for which they could charge a fee.

In fact, the flexibility enabled by SCOTA allows operators to target personalized offerings of features to selected segments of their subscriber base. For example, operators can promote a new or upgraded service or feature through targeted SMS/MMS campaigns, and interested consumers can install it on demand.

Summary

Software has become a key focus of competitive differentiation in the mobile industry and has vastly transformed the economics of the handset market. As the volume and sophistication of mobile handset software has grown, the process for effectively managing this software has become complex and costly, limiting operators’ flexibility and delaying the introduction of new handsets. SCOTA offers a new, more efficient embedded software management process that gives operators and consumers new ways to enhance the capabilities of their devices and mobile services.
About Red Bend Software

Red Bend Software, the market leader in Mobile Software Management and Firmware Over-the-Air (FOTA) updating solutions for mobile devices, is the first to launch a new solution for software component management that addresses the embedded firmware.

Red Bend’s vRapid Mobile™ offers unmatched capabilities to enhance and customize the growing software assets of mobile devices. vRapid Mobile is the first comprehensive solution to give the mobile value chain complete flexibility and control over updating, adding, changing, and removing embedded software components on any feature phone or smartphone at any point during the device lifetime. The solution is compliant with the emerging OMA-DM SCOMO standards enabler.

Red Bend Software helps mobile phone manufacturers and network operators to accelerate the adoption of new services and features, respond rapidly to customer needs, and reduce support costs through mobile software management solutions. LG Electronics, Motorola, NEC, Sharp, Sony Ericsson and other large handset manufacturers use Red Bend’s firmware over-the-air (FOTA) mobile client software to quickly and reliably deliver compact firmware updates to more than 200 million mobile phones in the hands of consumers. Founded in 1999, Red Bend Software is a privately held, venture capital-financed company with offices in China, Israel, Japan, Korea, the U.K and the U.S. More information is available at www.redbend.com.