

Why is a different operating system needed?

Summary

This white paper describes the key characteristics required of an operating system designed for mobile phones and explains why Symbian OS is the best-in-class mobile operating system.

1. Addressing specific needs

Small devices come in many shapes and sizes, each addressing distinct target markets that have different requirements. The market segment we are interested in is that of the mobile phone. The primary requirement of this market segment is that all products are great phones. This segment spans voice-centric phones with information capability (such as Series 60 phones) to information-centric devices with voice capability (such as UIQ and Series 80 phones). These advanced mobile phones integrate fully-featured personal digital assistant (PDA) capabilities with those of a traditional mobile phone in a single unit. In this article we'll be looking at the critical factors for operating systems in this market.

It is important to look at the mobile phone market in isolation. It has specific needs that make it unlike markets for PCs or fixed domestic appliances. Scaling down a PC operating system, or bolting communication capabilities onto a small and basic operating system, results in too many fundamental compromises. Symbian believes that the mobile phone market has five key characteristics that make it unique, and result in the need for a specifically designed operating system:

- mobile phones are both small and mobile
- mobile phones are ubiquitous – they target a mass-market of consumer, enterprise and professional users
- mobile phones are occasionally connectedconnected – they can be used when connected to the wireless phone network, locally to other devices, or on their own
- manufacturers need to differentiate their products in order to innovate and compete in a fast-evolving market
- the platform has to be open to enable independent technology and software vendors to develop third-party applications, technologies and services.

The way to grow the mobile phone market is to create good products – and the only way to create good products is to address each of these characteristics and ensure that technology doesn't limit functionality. Meeting the impressive growth forecast by analysts in a reasonable time frame is only possible with the right operating system.

2. Small and mobile, but always available

Mobile phones are both small and, by definition, mobile. This creates high user expectations. For instance, if you have your agenda on a phone that you also use to make calls and exchange data, you expect to be able to carry it with you at all times and to be instantly available whenever you want to use it.

Fulfilling these expectations makes considerable demands on power management. The device needs to be responsive in all situations, and cannot afford to go through a long boot sequence when it is turned on. In fact, the device should never be powered down completely since it needs to activate timed alarms or handle incoming calls. At the same time, a mobile phone must provide many hours of operation on a single charge or set of batteries. Meeting these contradictory requirements can only be done if the whole operating system is designed for efficiency.

3. Addressing the mass-market

Reliability is a major issue for mass-market phones. Data loss in a personal mobile phone causes a loss of trust between the user and the phone. A mobile phone therefore must be at least as resilient as paper diaries and agendas. Recalling phones to install service packs is a commercial and practical last resort – a mobile phone should never lock up or come with a major software defect. In fact, to use a PC term, it should never ever need a “reboot”! This is a far cry from desktop computers where bugs, crashes and reboots are expected.

It may come as a surprise to many computer users that a robust and reliable operating system is perfectly achievable. Even though nobody can guarantee bug-free software, a good operating system can make it much easier to write robust and reliable applications. Reliability requires good software engineering (including object orientation) and a good error-handling framework. Engineering best practice greatly helps reduce the number and severity of bugs while the error-handling framework enables graceful recovery from run-time errors such as running out of memory, low battery power or dropping a communication link.

Reducing the possibility of user code making the whole system unstable goes a long way towards achieving robustness. Ideally, the kernel, with its privileged code, should be small. System servers running without special privilege should handle much of the functionality conventionally handled by device drivers.

An effective memory management system is needed to prevent memory leaks. System resources should be released as soon as they are no longer needed and an effective, easy-to-use error-handling framework should manage out-of-memory errors properly. For systems that are never completely shut down and cannot be rebooted, keeping an accurate track of resources makes the difference between peak performance at all times and slow degradation to partial, or total, lack of usability. Applications and system modules that allocate blocks of memory should cater for the possibility that none might be available. Defensive programming has to be applied from the operating system through to the application level.

However, reliability alone is not enough to make good products. Sound consumer design is also necessary, where:

- product applications take advantage of the mobile phone's unique characteristics as well as its environment
- products should be designed to meet current usability and future developments in wireless technology
- consistency of style is paramount – if a feature is too complex to use, then it cannot justify either the time it took to develop or the space it takes in the device.

An operating system targeted at mobile phones must support these design principles by offering a high-level of integration with communication and personal information management (PIM) functionality. Symbian OS combines high functionality middleware with superior wireless communications through an integrated mailbox and the integration of Java and PIM functionality (agenda and contacts).

4. Handling occasional connectivity

Accessing remote data, sending email or synchronizing calendars requires some type of connection. Mobility constraints generally make a wireless connection preferable – whether wide area (using wireless telephony) or personal area (such as infrared or Bluetooth links).

Wireless connectivity is patchy, with different protocols around the world, fade-outs while moving and incomplete coverage – especially in remote areas, in some buildings or while airborne. It is unwise to rely on a permanent mobile connection – it is very frustrating for the user if such a connection is assumed. Wide area wireless networks are – and always will be – much slower than wired networks. An operating system must take this into account by delivering rich applications that are designed to manipulate the user's data while it is on the phone even when no connection is established. In short, the mobile phone must function as an advanced client, not a thin client, and the operating system must support this. There must be a smooth transition between being a window on the network and a self-sufficient device.

Connectivity requires an operating system with genuine multi-tasking, communications-capable real-time performance and a rich suite of communications protocols. In addition to the real-time requirements to maintain connections, the operating system must provide mechanisms to handle dropped connections gracefully and inform the user appropriately. To provide a smooth transition to the user and to be able to support forthcoming standards (such as third generation WCDMA and its evolution), network stacks must be abstracted in such a way that the application-level interface remains consistent no matter what type of protocol stack is used. The operating system has to provide a rich set of APIs to ensure that applications can benefit fully from current connectivity possibilities and be easily adapted to take advantage of new protocols as they are implemented.

5. Product diversity

There is an apparent contradiction between software developers who want to develop for just one popular platform and manufacturers who each want to have a range of distinctive and innovative products. The circle can be squared by separating the user interface from the core operating system.

Advanced mobile phones or “smartphones” come in all sorts of shapes – from traditional designs resembling today’s mobile phones with main input via the phone keypad, to a tablet form factor operated with a stylus, to phones with larger screens and small keyboards.

The different input mechanisms and form factors strongly influence the intended primary use of devices. With a very small screen and just a keypad, the main use tends to be voice calls. With pen input, browsing is quite convenient, but data entry is not. A keyboard is obviously the most practical mechanism to enter a large amount of data. These distinctions imply that user interfaces are ultimately both device and market dependent.

Product differentiation isn’t just a matter of operating system design. The operating system vendor must allow its licensees freedom to innovate and develop new product lines. Whether or not a vendor allows this is a key feature of its commercial model.

To support distinct phone families and yet maximize code reuse, Symbian focuses on the common code: Symbian OS, which includes a multi-tasking multithreaded core, a user interface framework, data services enablers, application engines and integrated PIM functionality and wireless communications. Phone manufacturers are active participants in software development, creating a large development organization to extend Symbian OS. This results in thousands of developers among licensees and partners having access to source code and ensuring that Symbian OS remains an ‘open standard’ – open and advanced.

This strategy ensures that Symbian OS phone manufacturers can create highly differentiated products while sharing a technology platform and keeping the learning curve to a minimum.

6. Open platform

An operating system for the mass-market must be open for third-party development – by independent software vendors, enterprise IT departments, network operators and Symbian OS licensees. In turn, this implies a manageable learning curve, standard languages such as C++ and Java, along with SDKs, tools, documentation, books, technical support and training. Symbian OS has a rich set of APIs for independent software developers, partners and licensees to write their applications.

Even though mobile phones are small and mobile, they can offer facilities as rich as those on desktop computers, in addition to basic functions such as voice and data communication. The operating system has to support both conventional and mobile computing paradigms, and developers need knowledge of both.

To reduce the time-to-market, developers should become proficient in as short a time as possible. Supporting standards that they may already know or can easily learn from a multitude of sources is necessary. Standards also make the platform more open and hence attract more developers.

Traditional standards such as Unicode for internationalization, a POSIX API, and Java are a must, but for an operating system to take its place in the connected world, open standards such as TCP/IP, POP3, IMAP4, SMTP, SMS, MMS, Bluetooth, OBEX, WAP, i-mode, Java and SyncML should also be supported.

Symbian has trusted leading partners in the mobile phone market and actively participates in standards organizations (such as the Open Mobile Alliance and the Java Community Process). Through these, Symbian has advance knowledge of future technologies and can test Symbian OS with many different phone systems. This ensures the stability and the future place of Symbian OS. Furthermore, a user interface framework, data service enablers and application engines provide a solid base for application developers to target.

7. A different operating system?

Symbian and its licensees aim to create a mass market for advanced open mobile phones. To deliver products that satisfy mobile phone users, an operating system must be engineered to take into account key functional demands of advanced communications on 2.5G and 3G networks.

To fit into the limited amount of memory a mobile phone may have, the operating system must be compact. However, as we have seen, it must still provide a rich set of functionality. What is needed to power a mobile phone is not a mini-operating system but a different operating system – one that is tailored. Symbian is dedicated to mobile phones and Symbian OS has been designed to meet the sophisticated requirements of the mobile phone market that mini-operating systems can't. They simply run out of steam.

8. Why Symbian OS

The five key points – small mobile devices, mass market, intermittent wireless connectivity, diversity of products and an open platform for independent software developers – are the premises on which Symbian OS was designed and developed. This makes it distinct from any desktop, workstation or server operating system. This also makes Symbian OS different from embedded operating systems, or any of its competitors, which weren't designed with all these key points in mind.

Symbian is committed to open standards and is actively working with emerging standards, such as J2ME, Bluetooth, MMS, SyncML, IPv6 and WCDMA. As well as its own developer support organization, books, papers and courses, Symbian delivers a global network of third-party competency and training centers – the Symbian Competence Centers and Symbian Training Centers. These are specifically directed at enabling other organizations and developers to take part in this new economy.

Symbian has announced and implemented a strategy that will see Symbian OS running on many advanced open mobile phones. Products in the market ^[1] show the diversity of mobile phones that can be created with Symbian OS. We can look forward to an even wider range of Symbian OS mobile phones.

Bibliography

- ^[1] See the up-to-date list of Symbian OS phones at <http://www.symbian.com/technology/symbos-phones.html>

Trademarks, copyright, disclaimer

Symbian licenses, develops and supports Symbian OS, the platform for next-generation data-enabled mobile phones. Symbian is headquartered in London, with offices worldwide. For more information see the Symbian website, <http://www.symbian.com>. 'Symbian', 'Symbian OS' and other associated Symbian marks are all trademarks of Symbian Software Ltd. Symbian acknowledges the trademark rights of all third parties referred to in this material. © Copyright Symbian Software Ltd 2005. All rights reserved. No part of this material may be reproduced without the express written permission of Symbian Software Ltd. Symbian Software Ltd makes no warranty or guarantee about the suitability or accuracy of the information contained in this document. The information contained in this document is for general information purposes only and should not be used or relied upon for any other purpose whatsoever. WP01-001-B-2005