Opinion Paper

The Multi-tasking Java™ Virtual Machine: Transforming the Mobile User Experience

By

Alain Blancquart

Chairman & CEO

Esmertec AG

July 2005

Contents

Headquarters:
Lagerstrasse 14
CH-8600 Duebendorf
Switzerland
Tel +41 44 823 8900
Fax +41 44 823 8999

Further information about Esmertec and its worldwide offices is available at
www.esmertec.com

Transforming the mobile user experience

In the second half of 2005, the wireless device market will reach an important milestone: for the first time mobile feature phones will hit the market sporting the next-generation Java™ Virtual Machine (JVM): the multi-tasking Java Virtual Machine. As I will show in this paper, the evolution of the JVM to the multi-tasking JVM promises a revolution in the mobile user experience. I see this revolution as further evidence of the shift in the wireless device market from a technology market, driven by functionality, to a consumer market, driven by convenience and lifestyle. This paper sets out the driving forces behind the multi-tasking JVM and explores the implications for users, device manufacturers and network operators.

Today's mobile feature phones are noticeably out of synch with the mindset of mobile users these days: they can do only one thing at a time. But in our high-tech world multi-tasking has become the norm in people's daily lives. People optimize their time by accomplishing multiple tasks simultaneously and the next generation of mobile devices must target this mindset and satisfy this desire. Mobile feature phone users, who grew up in the Internet age, are used to the capabilities and functionalities of the personal computer (PC) and smartphones, and want the same multi-tasking flexibility. They want to be able to play games and receive messages or listen to music and read news alerts—all at the same time. Their attention spans have shortened and they demand instant gratification.

Changing the life of the user

The multi-tasking JVM promises to change the life of the mobile user, first by breaking down barriers to accomplishing multiple tasks simultaneously on feature phones, and second by creating multiple opportunities to utilize feature phones in exciting new ways. In this section, I will present a portfolio of use cases that constitute a world of new opportunities that promise to make the mobile experience dynamic and fun.

Breaking down barriers

Until now it has been impossible to run more than one application on feature phones at the same time. Let us say that you are playing a game on your handset and want to check the results of your favorite football team. To do this, you must exit the game and start another application to retrieve the results. Once you have checked the sports scores, you must then exit the application and restart the game - and in some cases start playing the game all over again from the beginning. This can quickly become tiresome and most users today simply will not tolerate this constant restarting and stopping of applications in order to run multiple applications on feature phones. Simply put, they do less with their phones than they wish to do.

With the multi-tasking JVM, the user in the example above could easily switch between the game and the football results, without having to exit and restart the applications, because both applications would be running simultaneously. In fact, the user could be notified of the football results *while* playing the game, without requiring him to leave the game in the first place.

Here is an example for power users. Let us say you are reading the latest news ticker on your handset, and you are wondering if you have received an important email you have been waiting for. To check, you must go to your email inbox, which requires exiting the news ticker. Once you have checked your inbox, you must restart the news ticker and wait for it to update to the current news. Switching back to the news ticker requires exiting the email program. The process must, of course, be repeated whenever you want to switch between your email and the news ticker. Feature phones allow you to run only one user application at a time, which significantly limits their usefulness.

With multi-tasking, you can easily switch between the news ticker and the email program, with both programs running on the device simultaneously. This means that once you have checked your email, you can switch back to your news ticker, which has been running and continuously updating in the background. Moreover, since the email program continues to run in the background, you can even receive an alert sound or notification when new mail has arrived, saving you time in checking your email unnecessarily. The process is simple, easy to use, and with only a few button clicks you can be sure you are kept up-to-date all the time!

Creating multiple opportunities

Since the emergence of Java applications developed for mobile phones, the wireless industry has just scratched the surface of the potential of Java technology on wireless devices.

Today, while Java software is being deployed on an increasing number of phone models, Java technology is used primarily for games. In many cases users who do not play games on their mobile phones don not use the Java software at all. Many are probably not even aware that their phone is Java-enabled.

The multi-tasking JVM unlocks the potential of Java applications on mobile phones by offering users opportunities to interact with their devices in ways that fit with their lifestyles. The multi-tasking JVM enables a dynamic user environment wherein users can stay connected while exploring an exciting array of new Java applications:

Media Players – While listening to your favorite songs using a media player written in Java language you can use your phone as usual. You may wish to use one or more office tools, also written in Java, in parallel. Similarly, using the media player need not block other Java functionality like handling incoming SMS messages.

Downloads – When we are download a large file on our PC, most of us do something else in parallel. Why should we not be able to do the same with a mobile phone? If you download the latest MP3-track with your Java media player you could save the world in a mobile game in the meantime and be notified when your download is finished.

News, Stock Tickers, and RSS Feeds – Your customized content can be constantly and automatically updated on your phone in the background, while you accomplish other tasks.

Chat, IM, and Push-to-talk – Game players can utilize chat, IM or push-to-talk services that enhance a Java game between multiple remote players. For example, when you have lost one round in a multiplayer game and are waiting to join your team again you could use one of these services in parallel to have a chat with the other waiting players.

Push and Messaging Applications – Push email, instant messaging and other messaging applications can run in the background and notify you when messages are received - all while you are doing something else with the device.

Seamless Integration of Applications – The multi-tasking JVM offers you the possibility to interrupt a mail or text message during composition to look into your contact manager or the web browser and then continue to write without losing your work.

Role-playing Games – Imagine that you just spent an hour as a novice dragon slayer, only to be interrupted and have to switch from the game to another application. With the multi-tasking JVM, you will not lose your placing or score: once you have time to switch back to the game, you can continue where you left off without having to slay the same dragons all over again.

Other Leisure Applications – With the multi-tasking JVM you can be alerted to events of personal interest to you while continuing to use your device as usual: bus and train notifications to tell you if your ride to work is running on time; soccer or hockey results of your favorite teams delivered to your mobile phone; and even horserace results to let you know if your long-shot came in.

Enterprise applications – For business users, the multi-tasking JVM enables a constant update of information, for example, inventory, product prices, appointments, emails, etc., in the background without any interruption to the phone operation. This ensures that the person will always have access to the latest information.

Mobile marketing applications – Java multi-tasking opens new possibilities for mobile marketing on mobile devices: permission advertising, for example, can be delivered to mobile phones in the background and can be viewed when the user opens the specific content or application associated with the advertisement.

M-commerce applications – With the multi-tasking JVM, m-commerce applications can be conducted with fewer intrusions to the normal operations of the phone. For example, while browsing for tickets, they can be ordered by email without exiting the browser and if confirmation notification can be received while the user continues browsing.

These examples are only the beginning. The multi-tasking JVM creates a world of new opportunities for mobile feature phone users, offering them convenience, time-savings and more entertainment at the same time.

A consumer-driven technology

The multi-tasking JVM is being driven by the multi-tasking mindset of users and as we have already seen, it creates exciting new possibilities for mobile feature phones. In this section, I move beyond the end user to consider why handset manufacturers are eager to embrace the multi-tasking JVM and the important benefits of the multi-tasking JVM for network operators. I will also discuss some of the technical challenges in implementing the multi-tasking JVM, and how one company – Esmertec™ - has succeeded in overcoming these hurdles to be the first company, to our knowledge, to bring the multi-tasking JVM to the market.

What is the multi-tasking Java virtual machine?

The multi-tasking JVM is a general-purpose virtual machine for running multiple applications written in the Java programming language concurrently on a mobile device. Without multi-tasking, an active Java application stops running when another Java application is launched. Multi-tasking enables applications such as downloads, media players and push services to continue operating while the user interacts with another application - such as messaging or a game. By giving users the opportunity to do more than one thing at a time, the multi-tasking JVM enables a much more dynamic user experience.

With multi-tasking JVM, a task manager - much like that on a PC - is used to switch between running applications or to stop them. This type of leading-edge platform provides a robust, adaptable and dynamic environment for Java applications that can be leveraged to meet the latest demands of the evolving mobile market.

Why handset manufacturers are embracing the multi-tasking JVM

I see intense interest among handset manufacturers for running multiple Java applications in next-generation handsets. The key reasons for this interest include:

- The portability offered by Java technology. An increasing number of applications that were once developed and deployed to mobile phones in native code are being written in Java language. This enables manufacturers to develop an application only once and to deploy the same program on their entire product range where each device might have different hardware configurations and operating systems.

- The security offered by Java technology. The security features of Java technology prevent rogue applications from being used as hacking tools.

- Better optimization of resources on the device. The majority of user applications that typically run on today's handsets such as email, instant messaging (IM), calendar, address book and call management are provided as native applications. Using such user applications programmed in Java language can enable a better use of resources on the device since the Java applications provided by the same supplier would have been optimized to work well together.

- Minimize handset recalls and repair costs. Today, if there were a software problem discovered on a mobile handset model, the device is recalled or being sent in for repair. With a multi-tasking JVM, it will be possible to send bug fixes and software updates directly to the devices (with the user's permission) for seamless fixing. This would result in huge cost savings.

- Enhancing wireless services. With multi-tasking feature, handset manufacturers can enable their customers – the network operators - to offer a lot more wireless services and applications both in the consumer and enterprise space.

Simply put, the multi-tasking JVM is attractive to handset manufacturers because it enables them to produce more innovative handsets while reusing the existing hardware configurations and operating systems they employ for their mobile devices. This significantly increases the value of their product offering to their customers.

Benefits to operators

For network operators, the multi-tasking JVM promises to spur increased consumption and use of Java applications, significantly boosting data traffic over their networks, thereby increasing average revenue per user. In enhancing their services to consumers, this could also lead to reduction in churn when consumers change providers.

Operators may also harness the power of the multi-tasking JVM to enable remote device management applications that aid in the branding, personalization and ongoing monitoring of devices deployed in the field. Branding and personalization applications can come in the form of screen savers, wallpapers, icons, tickers or notifications. Applications delivered in the background under the multi-tasking JVM are invisible to the user and can enhance the overall service quality and delivery.

In a similar way, the multi-tasking JVM could allow operators to distribute regular software updates to devices already deployed in the field - either for free or for a fee.

Simple in concept; difficult to implement

Multi-tasking may sound simple in concept, but there are a variety of problems that need to be solved to run multiple Java applications in resource-constrained devices such as feature phones:

- Running multiple Java applications in parallel without utilizing a large amount of additional phone resources such as memory, processor, and battery.

- Backwards compatibility. The multi-tasking JVM must accommodate the large number of Java applications that have already been deployed, without requiring them to be rewritten, and must not impose design constraints on new applications.

- Robustness and reliability. Currently, a JVM needs to run only as long as a single Java application is running. Each time the application quits, the JVM quits along with it and is restarted each time a Java application is launched. In contrast, the multi-tasking JVM must run continuously, and therefore robustness requirements are correspondingly high. One challenge facing the multi-tasking JVM parallels a common operating systems problem: a bug in an application brings the application down, but a bug in an operating system brings all applications down.

- Efficient allocation of memory among applications. The multi-tasking JVM must clean up allocated memory properly after each application is stopped, and ensure fairness between running applications.

- Working within limitations of today's operating systems. Feature phones simply do not have enough processor, or memory resources to run a full-scale operating system such as Linux, Windows CE, or Symbian. The multi-tasking JVM must work within the resource constraints of these devices, while optimizing battery life.

Overcoming these technical hurdles has proven difficult for the industry. In fact, as of this writing, only one company to our knowledge – Esmertec - has succeeded in bringing the multi-tasking JVM to market.

Bringing multi-tasking Java to mass-market phones

In mid-2004, Esmertec was first-to-market with its multi-tasking JVM when it launched Jbed™ Advanced. Jbed Advanced is a solution that enables Java multi-tasking on mass-market mobile phones, bringing functionalities normally found on smartphones to feature phones. A high-performance modular Java execution platform, Jbed Advanced allows multiple applications to run concurrently on a mobile device while optimizing memory, processor and battery resources.

Our customers were quick to adopt Jbed Advanced, and in the second half of 2005, we expect see the first devices with Esmertec's multi-tasking JVM hit the market. Other companies have made their own multi-tasking JVM product announcements, but as of this writing, Esmertec's multi-tasking JVM solution stands alone in the market.

Conclusion

So what does the multi-tasking JVM change? In this paper I have shown how it transforms the mobile user experience through a richer user environment. It enables handset manufacturers to increase the functionality and value of next-generation handsets. For telecom operators, the multi-tasking JVM promises to expand the content services market, thereby increasing the average revenue per user. A multi-tasking JVM will facilitate new and exciting applications to fill the large new color screen displays that are being incorporated into next-generation handsets.

Moreover, the multi-tasking JVM can be seen as a huge step towards ultimately providing a complete operating environment for Java programs on mobile devices. On its current course, the Java platform could one day subsume the role of modern operating systems as well as middleware platforms.

While the multi-tasking JVM is itself an innovative technology, it is crucial to remember that it is being driven by the multi-tasking mindset of users. By targeting this multi-tasking mindset, Esmertec has been first to deliver breakthrough products that enable mobile devices to be designed and deployed in a flexible way to meet diverse customer needs. With powerful new tools like Jbed Advanced in hand, there can be no doubt the multi-tasking JVM revolution is imminent. This is the future of our market and this future belongs to those who can envisage what it is going to be.