Chapter 15

Java ME and the Future

Up until this point in the book, you have looked at what is possible with Java ME today and at its current state of development. You have familiarized yourself with the kind of hardware you need to support, the APIs at your disposal, and the techniques best suited for developing for that hardware and APIs.

You examined the kinds of tricks and optimization needed to get the best performance out of current devices, at what users expect to find in current Java ME applications, and at how today’s Java ME applications stack up against the competition in terms of functionality and user interfaces.

You have also looked at the various pitfalls that Java ME developers can encounter today, ranging from fragmentation issues to simply approaching the development process with the wrong mindset.

Now it’s time to take a quick look at the future, and see what the Java ME world will look like in a few years’ time. In this chapter, you will examine what the future holds for Java ME in terms of its target audience, available hardware, available APIs, development techniques, and more.

NOTE: The opinions expressed within are my personal ones, but they are based on my experience with the platform and on current market trends.

Java ME Hardware Evolution

In the Java ME world, almost everything these days comes down to hardware. As many features and capabilities supported natively on other platforms are missing from Java ME’s default feature set (XML support, graphics capabilities, and so on), developers have to implement these features in Java code, which is both slower and a drain on available resources.

Furthermore, mid-level and low-end devices combined tend to dictate what kind of software you can write for any given platform, and the relatively large gap between them
in Java ME’s case means that developers have to walk a very thin line between underdelivering in terms of features, or overshooting in terms of requirements.

Looking at the growth that can be seen in terms of hardware performance in the past couple of years, especially the very impressive one in the low-end device segment, we can draw two very important conclusions.

The first one is that low-end devices have evolved, and still are evolving, rapidly in terms of hardware performance and are starting to catch up with mid-level devices. This is great news, as a very large part of the Java ME development effort stems from supporting these low-end devices. With their processing power increasing, developers will have to resort less and less to clever programming tricks and extreme optimizations, and they will be able to include more and more features as standard.

The second conclusion is that mid-level and high-end Java ME hardware isn’t evolving at the rate it used to. Although these segments are also increasing in processing power, their growth rate isn’t nearly as big as it used to be. This happens primarily because top hardware these days gets allocated to smartphones, not to feature phones.

Coupled with the rapid growth in the low-end segment, in the end this will lead to a “closing of the gap” between low-end, mid-level, and high-end Java ME devices. Differences will still exist between them, but not nearly as significant as those seen today. This is good news because Java ME will become a much more consistent platform to develop for, but also bad news because the slower growth in terms of hardware performance correlates to a slower growth in terms of innovation and new features.

That being said, processing power isn’t everything. Other hardware aspects are almost as important, if not more so. For example, more and more phones today have touchscreens, which leads to more-intuitive and far more enjoyable user interfaces. Furthermore, screen sizes are constantly increasing, while support for 3D graphics is also starting to become mainstream, as is support for sensors and global positioning.

What this means is that tomorrow’s Java ME applications will have much more visually pleasing and intuitive interfaces and will derive their appeal not necessarily from their processing abilities, but from the way in which users can interact with them. This trend can be observed even today, as top-notch Java ME user interfaces tend to rival those found on smartphones or on PCs, and advanced proof-of-concept applications, such as augmented reality, are starting to appear.

Also, as devices receive bigger and faster storage capabilities, and as they become more and more connected, Java ME applications will eventually be able to store, manipulate, and share very large amounts of data. Coupled with the growing trend of cloud computing, this will lead to Java ME applications that work together with server-based software to truly empower their users. Even offline applications will benefit greatly from this growth in storage capabilities. For example, they will be able to include more-detailed graphics and multimedia elements, and they will be able to store and work with more of the user’s data.