

# SOFTWARE TECHNOLOGY FOR WAP BASED M-COMMERCE - A COMPARATIVE STUDY OF TOOLKITS FOR THE DEVELOPMENT OF MOBILE APPLICATIONS

Karl Kurbel, Andrzej Dabkowski, Piotr Zajac  
*Europe University Viadrina  
Business Informatics  
D-15230 Frankfurt (Oder), Germany*

## ABSTRACT

As the technologies for mobile commerce are maturing, professional support for the development of mobile applications is becoming increasingly important. In the same way as integrated development environments (IDEs) were introduced and are being used for conventional application development, IDEs for mobile application development are becoming available nowadays. In this paper, the features of such IDEs are discussed. Three software packages which assist developers in the creation of mobile content are investigated. These are the packages by the technology and market leaders in the mobile solutions area, Nokia, Ericsson and Phone.com.

## KEYWORDS

WAP toolkits, mobile applications development, m-commerce.

## 1. INTRODUCTION AND STATE OF THE ART

The current standard for mobile Internet applications is defined by the Wireless Application Protocol (WAP [7]). WAP includes a set of communication protocols and an application environment including WML (Wireless Markup Language [7]).

The latest version, WML 2.0 [7], was developed by the W3C consortium [9] and adopted by the WAP Forum [5]. WML 2.0 is based on the XHTML Mobile Profile [10] and WAP CSS (Cascading Style Sheets) [6]. The full power of WML 2.0 will unfold with UMTS (Universal Mobile Telecommunication System [4]).

We are going to discuss three well-known SDKs offered for free by technology leaders Nokia, Ericsson (currently: SonyEricsson) and Unwired Planet (Phone.com). The SDKs are intended to present the content to the developer in almost the same way as a vendor-specific mobile browser will do.

## 2. BASIC ANALYSIS OF SOFTWARE DEVELOPMENT TOOLKITS

### *Ericsson IDE 3.1.1 Application designer*

The WapIDE toolkit by Ericsson consists of three not entirely integrated tools: *browser*, *application designer*, *push initiator*. The *browser* simulates a device equipped with WAP software from Ericsson. With the use of the browser WML decks and cards can be accessed. The browser has a facility of tracing different types of system events. This information can be observed in the browser trace view.

The *application designer* is the main part of the WapIDE toolkit. The WapIDE application designer is a WML and WMLScript [7] editor to design and test WAP applications. WML and WMLScript files are managed in WapIDE projects.

The design area of the WapIDE consists of several sections and scrollable panes. The major panes are the project and the source panes. The project pane presents the view of the current project with the help of a special generated tree view. It shows the files in the project and the structure of the WML files. The attribute pane shows the attributes of the current project, file or WML element in a file. The message pane indicates messages from compilation time.

The *push initiator* is another part of the WapIDE that is responsible for sending push messages to the browser or to a real terminal.

The Ericsson WapIDE has some shortcomings. Unfortunately the attribute pane indicates attributes of the currently selected WML element in the project tree view only, but it does not show the attributes of the currently edited element in the source pane. The toolkit provides a number of wizards. Those wizards sometimes do not enforce that the syntax is correct. As an immediate consequence of incomplete syntax the source code cannot be compiled successfully.

Another weakness regarding testing is that the toolkit does not support any debugging mechanisms like tracing variables, dynamic assignment or even displaying variables. The WMLScript editor does not distinguish keywords from other text, e.g. by using colors.

Apart from these small shortcomings, the Ericsson WapIDE provides some useful features. One of these features is storing mobile applications in projects.

*Nokia Mobile Internet Toolkit 3.0*

This toolkit provides a set of tools for preparing mobile Internet services and includes many features and components such as: device simulators; WML and WMLScript encoders; WML, WMLScript, WBMP [7], push, multipart content, XHTML [10], and CSS editors; WAP protocol stack, HTTP and file access modules; debugging views; WAP server simulator, a WAP gateway for local toolkit use only; Wireless Telephony Application Interface (WTAI) [8] features; Nokia SoftID [8], a Wireless Identity Module (WIM) [8] card simulation; basic WML and XHTML sample applications and source code.

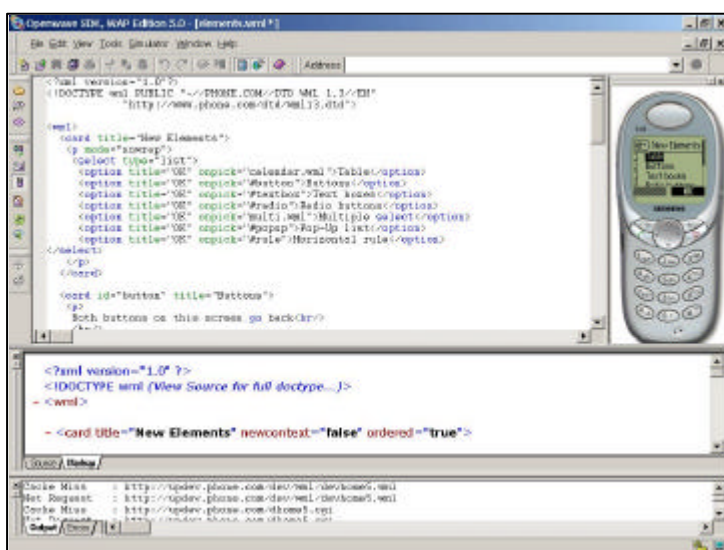
Content that users create is encoded if necessary and usually immediately displayed. Compilation status messages are displayed at the bottom of the editing window.

The toolkit is also designed to assist developers during the process of developing a push application. A push message has both a binary (encoded) and a text (source) form. There are facilities for helping the developer observe and debug the effects of various pushed content on the client system: displaying a push message on the simulator, effects on the cache, when URLs get loaded, and so on. The NMIT provides also a mechanism for simulation of phone calls. That mechanism is based on WTAI.

The Nokia Mobile Internet Toolkit offers a wide range of configuration and customization functions, which are easily accessible through numerous intuitive menus. These features allow customizing device settings, connections, toolkit preferences, etc.

*Openwave SDK WAP Edition 5.01.*

The user interface is very intuitive, typical for programs running under Windows operating systems. The



various windows displayed within the IDE can be moved and docked, or can be displayed outside the IDE window.

Figure 1. Openwave SDK WAP Edition 5.0 – Main Window

The two main windows are the editor window and the browser simulator window. There is also group of debugging windows: variables; cookies; history; HTTP response; browser output.

Together with the Openwave SDK program files basic documentation (Getting started guide, WML/WMLScript guides and references) can be installed. The information included is Figure sufficient even for absolute beginners

in this area. Experienced users can explore the Openwave Developer site.

The Openwave Software Development Kit WAP Edition 5.0 is a user-friendly application, easy to learn for beginners. The program itself is stable; it has no extraordinary hardware requirements and offers a huge range of customization possibilities.

All three toolkits can be downloaded for free. There are dedicated websites for network operators, developers, content providers, technology suppliers and others interested in mobile Internet services and applications, innovative mobility projects, etc. [1], [2], [3]. Additionally those sites offer rich choice of technical documentation and information on mobile technologies. Basic technical requirements are summed up in the table below.

Table 1: Technical requirements

	Ericsson IDE 3.1.1 Application Designer	Nokia Mobile Internet Toolkit 3.0	Openwave SDK WAP Edition 5.0
Operating system(s)	Win 98/NT/2000	Win 98/NT/2000	Win 98/Me/NT/2000
Hardware minimum	P II 266/128 MB RAM/20 MB HDD	P 300/128 MB RAM/50 MB HDD	Not stated by the manufacturer
Java platform	Required	Required	Not required

### 3. ASSESSMENT OF THE TOOLKITS

In discussing the toolkits we focus on aspects that facilitate the process of designing, developing and testing mobile applications. We consider the following features: software and hardware requirements, graphical user interface, design/development environment features, documentation, and general impression.

#### *Technical Issues and Requirements*

Requirements for the Nokia and Ericsson toolkits are specified in the documentation. Nokia recommends strongly more hardware resources, hard drive space, and RAM memory. Both toolkits use the Java platform, therefore they consume many resources. Moreover, the Nokia software seems to be mostly implemented in Java and sometimes the screens switch and refresh slowly.

The Openwave documentation does not state hardware requirements. We ran the Openwave toolkit on the same machines as the other two toolkits and found that it did not require more resources than those. It is necessary to note that for none of the toolkits installation versions on other platforms than Windows are available. Linux, for example, is not supported yet.

#### *Graphical User Interface*

Evaluation of graphical user interfaces (GUI) is not entirely free from subjective judgment. We found that Openwave provides the richest possibilities of GUI customization, shaping of windows, and design area layout. The Ericsson toolkit appears to be the most elementary on regarding GUI customization. The options the toolkits provide are easily accessible in all three toolkits, but the Nokia toolkit provides a large number of options that are clearly located and can be quickly accessed in the user interface.

#### *Application Development*

The three toolkits are not based on the the well-known drag-and-drop technique. The user, not the toolkit, is mostly responsible for source code production. All of the tools add appropriate headers to files automatically, but only Ericsson supports the user with mechanisms for source code automation. Those mechanisms have some shortcomings.

All toolkits recognize syntax elements but the Ericsson toolkit does not highlight the keywords of WMLScript files. The most extended functionality concerning testing and debugging of mobile applications is offered by Nokia. It is possible to add new and manage existing variables in the current context of the mobile device simulator. The weakest toolkit in this respect is the one by Ericsson. This toolkit provides neither views for debugging and variable tracing nor features for adding and manipulating variables.

All three toolkits provide useful comments when errors occur although Openwave does not go into the source code line after the message is clicked. The Ericsson toolkit's manner of storing and managing files is

remarkable. Files are maintained and grouped into projects. The concept of a project makes file management and content distribution much easier.

#### *General Impression*

During the investigation of the toolkits we did not encounter any serious faults. The Nokia toolkit seems to be the most complex one, requiring the largest resources. Sometimes stability problems arose, especially when the various simulators for mobile devices were tested and changed. In some cases the simulator just hung up and could not be initialized again. Ericsson and Openwave performed without problems with regard to stability. The elements of the toolkits are well integrated. However, the push initiator in the toolkit by Ericsson is not integrated with the other tools. It needs to be launched separately.

All toolkits provide information about what is done behind the curtain, like requests and responses info, sessions, cache info, stack history and other diagnostic information. However, the Nokia and Openwave toolkits provide more detailed data of this kind in comparison with the Ericsson toolkit. Among the considered toolkits, Nokia offers definitely more possibilities of customizing device settings, connections and toolkit preferences. The weakest toolkit in this field is the one by Ericsson, providing just basic customization features.

The basic functionality required for creating and testing mobile applications is available in all toolkits. However, Nokia carries some more sophisticated features, e.g. WTAI (simulation of making calls), although not all Nokia simulators support this feature. Limitations of the specific phone emulators are presented in detail in the toolkit's documentation.

Simulators of the mobile devices are a separate subject. The producers of mobile phones, like Ericsson or Nokia, included in their mobile toolkits simulators of their real mobile units. Openwave, the only one that is not a hardware supplier, offers Siemens S45 as one of its simulators. During the use of the Ericsson and Nokia tools, occasional problems with the cache mechanisms occurred, and with changing and adjusting the simulators. Simulators embedded in toolkits differ from each other like real mobile phones vary. It is a useful and interesting practice to test a mobile application under development on a variety of mobile simulators. In this way, one can investigate whether the mobile application works properly on different devices. Numerous offline examples for mobile development are given in the Nokia and Openwave toolkits. The latter one presents also online examples which are opened when the program starts.

## **4. CONCLUSIONS**

The SDKs discussed in this paper provide a relatively easy and comfortable entry into application development in the world of the mobile Internet. The toolkit offered by Openwave seems to be the one best suited for beginners but it lacks sophisticated features as available in the package by Nokia. The Ericsson IDE has good organizational support as it arranges files in projects, and user friendly drag-and-drop facilities.

Our investigation was done from an application developer's point of view. Being involved in a project where mobile front-ends for business information systems are under development, our intention is to share experiences with WAP toolkits with others who are or who are going to be in a similar situation.

## **REFERENCES**

- [1] Ericsson Mobility World: <http://www.ericsson.com/mobilityworld>
- [2] Forum Nokia: <http://www.forum.nokia.com>
- [3] Open Developer Site: <http://developer.openwave.com>
- [4] UMTS: [http://www.gsmworld.com/technology/3g\\_intro.html](http://www.gsmworld.com/technology/3g_intro.html)
- [5] WAP Forum: <http://www.wapforum.org>
- [6] WAP CSS: <http://www.forum.nokia.com>
- [7] WAP, WML, WMLScript, WBMP: <http://www.wapforum.org/what/technical>
- [8] WTAI, WIM, SoftID: <http://download.forum.nokia.com/download/WAP2000Overview.pdf>
- [9] W3C: <http://www.w3.org>
- [10] XHTML Mobile Profile: <http://www.forum.nokia.com>