A comparative study of mobile application development in Symbian and J2ME using example of a live football results service operating over GPRS

O. Rashid, R. Thompson, P. Coulton and R. Edwards

Abstract — *It is commonly acknowledged that the market for mobile phones has reached a level of saturation. The mobile operators have invested heavily in the provision of the Universal Mobile Telecommunications System (UMTS) and the operators such as Three (formerly Hutchinson 3G) have failed as such in the wide scale adoption of their services. There is therefore need for applications that will encourage adoption of new devices and services. In this paper we will compare the two different development options for mobile devices, namely Symbian and Java. These will be compared on the basis of a Live football updates application.*

Index Terms — GPRS , J2ME, Symbian.

I. Introduction

For many operators GPRS was initially viewed as merely a stepping stone in the evolution of their existing General System for Mobile (GSM) communication network to one compatible with the packet data requirements of UMTS. However, with operators such as Vodafone, delaying their deployment of UMTS until their GPRS provision is saturated. There is a strong argument that we should view GPRS as an opportunity to drive consumer demand, in a similar manner that NTTDoComo adopted with the provision of i-mode, through the provision of novel applications and services taking advantage of the always on line provision on a mobile device.

However application development for mobile phones have been complicated by the fact that until recently we have only been able to develop cross platform applications using a cut-down version of Java, the so called Java 2 micro edition (J2ME) [3] which is not ideally suited to the hardware architecture of mobile phones and in particular their use of DSP processors rather than general purpose microprocessors. We are now seeing the emergence of a number of operating systems that have been designed specifically to suit the phone architecture, the principal two being Symbian [2], [4] in Europe and Binary Runtime Environment of Wireless (BREW) [5] in the United States.

O. Rashid is a PhD student at Department of Communication Systems, Lancaster University, Lancaster UK (e-mail: m.rashid@lancaster.ac.uk)

R. Thompson is a 3rd year student at Department of Communication Systems, Lancaster University, Lancaster UK (email: r.j.thompson@lancaster.ac.uk)

P. Coulton is with the Department of Communication Systems, Lancaster University, Lancaster UK (email: p.coulton@lancaster.ac.uk)

R. Edwards is with the Department of Communication Systems, Lancaster University, Lancaster UK (email: r.edwards@lancaster.ac.uk)

In this paper we will compare a live football updates and fantasy football application created in J2ME and Symbian. In the following sections we will give a brief overview of the application and then discuss the major advantages/disadvantages of using J2ME/Symbian. Finally we will compare the two environments on the basis of the live football updates application in terms of architecture and user interface. Finally we shall provide a general guideline as to their advantages and disadvantages.

II. Application Overview

For any application to succeed it must address a particular market requirement or "need" and preferably offer something that either does not exist or improves on the existing provision. Furthermore, applications are often affected by the culture in which they operate, in other words what succeeds in one country may not necessarily succeed in another. This application address both aspects of a perceived need coupled with a strong cultural tradition.

The English premiere league is arguably one of the best football leagues in the world and people in the UK are famous for passion and support of this game. On any given match day supporters like to stay up to date with latest news and running scores, not only of the particular club they support but also the matches for the entire league. Current services offered to mobile users work over the Short Message Service (SMS) and are generally limited to goal alerts with the user paying by the number of SMS messages received, for the club or clubs he has subscribed to. The application presented in this paper uses GPRS as its data bearer, to both improve the timeliness of information and the range of information available without increasing the cost to the user. This application offers comprehensive updates from all the football matches on a given day providing such information as goals, red cards, yellow cards, substitutions, etc. All the information is displayed in an easy accessible format where a user can simply scroll through the event by event coverage of the entire league games on that day.

The application requires a constantly updated source to receive new updates. The client server architecture for receiving the live updates is shown in the following figure.

0-7803-8526-8/04/$20.00 ©2004 IEEE.

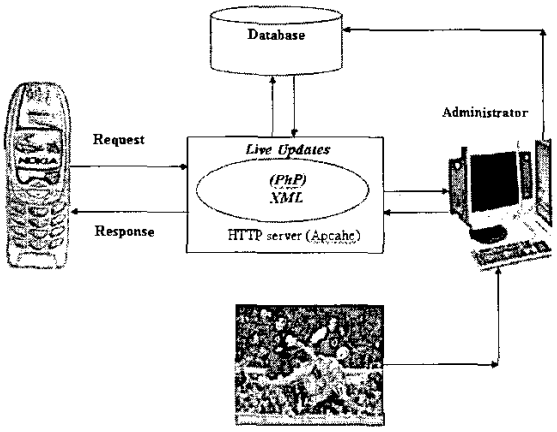

Fig. 1. Client server architecture for live updates [1]

The live update component, running in a Java or Symbian enabled phone, sends a request to the server with the update ID. The server has received its live information from a particular game via updates from the administrator. This update information is stored in an XML file with an update ID, all the details about the event (such as the type of the event e.g. goal, red card, yellow card or substitution), the name of the team and the name of the player involved in the event. All the updates, at the end of a day's play are stored in a database. The request coming from the client is received by a PHP script which checks if the update ID received from the phone matches with the one on the server. If the ID matches then the PHP script parses the XML file containing the event data and returns it to the client. Once the response has been received back at the phone it is interpreted and displayed [1].

This particular client/server communication model was implemented because Mobile Information Device Profile (MIDP) version 1.0 implementations are not required to support socket based connections, in fact the only connection protocol they are required to implement is Hypertext Transfer Protocol (HTTP) 1.1. Hence using HTTP as the connection protocol makes this application more versatile, giving it the ability to run on any MIDP 1.0 enabled mobile phone. Although MIDP 2.0 implementations support socket based connections HTTP was selected to have a similar communication model for both applications developed over Symbian and J2ME.

III. J2ME Vs Symbian

In order to make a comparison of this application, where information delivery to the client in the least possible time at an economical charge is of utmost importance, we will discuss the advantages and disadvantages of an application running over J2ME and Symbian. We shall analyze the effect of these particular implementations on the live football results service.

J2ME is essentially a subset of Java 2 Standard Edition (J2SE). Some pros and cons of developing on J2ME are

- Java enabled phones (including those running over Symbian) are much more common in the market hence giving a wider user base which can essentially affect the success of an application as it can reach a wider range of users.
- Java applications are very easy to install over the air (OTA), their limited size make them ideal for such an installation. Users can simply select the application they want to purchase and download it to their phone which is one of the factors that has seen Java based games and applications becoming more and more successful recently. Although, this means that developers are restricted to a limited application size of a few kilobytes (30K - 64K) compared to a Symbian application which can be in the size of multiple megabytes.
- J2ME provides and high level user interface which allows developers to create check boxes and text fields etc with a provision to program the soft keys.
- Persistent data storage on the phone is achieved through a Record Management System (RMS) which allows the application to record its data as individual entries which can be searched by entry number at a later time. Java Database Connectivity (JDBC) is an optional package if an embedded database is to be used.
- J2ME (MIDP 1.0) has no direct access to IrDa, Bluetooth, phone dialing, phone book and calendar which makes it harder for applications to interact with each other. One way to overcome this and share data between two J2ME applications (midlets) is to allow them to share the same record store. Although MIDP 2.0 provides support for Bluetooth in the form of JSR-82 the number of mobile phones in the market that are MIDP 2.0 are very few compared to MIDP 1.0, hence at this point Symbian has a distinct advantage over java which allows applications to access the generic features of the phone directly.
- Generally J2ME has no access to SMS but this can be overcome by vendor specific API's which allow MIDP applications to access SMS.
- Although J2ME is well documented and has a huge developer community one of the most critical issues is that it has no facility to automatically scale the drawing area to the screen size and this causes serious problems for the developers as they have to design different versions of the same application to suit the needs of phones of various screen sizes.

Symbian is an operating system for the new generation of smart phones that runs applications written in C++ and compiled to machine code. Smart phones are less common in market as compared to the java enabled phones but they are capable of running much complicated applications. Some pros and cons of application development on Symbian are:

- Since more memory is available for the applications more advanced and rich applications can be developed although it could be argued that it makes them difficult to install an application, spanning multiple megabytes, OTA.
- An application developed for Symbian OS can access IrDa, Bluetooth (if present on the phone), SMS, calendar, contacts and even dial the phone. Hence allowing multiple applications to communicate with each other creating a much enhanced experience for the user.
- Since the application code is compiled to machine code Symbian application run much faster as compared to their counterparts written in J2ME.
- Since Symbian applications can access the hardware of the phone it is often required to change some code of the application to make it compatible to a different phone but an application for instance developed for Nokia series 60 will run on all series 60 phones with out modification.
- Since a much larger amount of memory is available to store data, with a further provision for expansion, data storage is not a problem for a Symbian application. Data can easily be stored in applications own data files or an embedded database. This data can be then be shared between different applications.
- Symbian is a multitasking OS which means that more that one processes or servers [2] can be started to suit the needs of various applications. This provides a distinct advantage over java applications as the application can be made to run as a background process. Although it can be argued that a java application can take advantage of the multitasking nature of the Symbian OS, which is in fact true but the number Java enabled phones currently in the market running over Symbian are not that many and this is a factor that is going to be changing in the near future.

IV. Architecture Comparison

Figure 1 shows the overall communication model as to how the application will interact with server to send and receive information. In order to compare the two applications the same communication model was used [1]. In this section we will discuss how the two applications differ from each other. Figure 2 below shows the behavior and working structure of the application created in java.

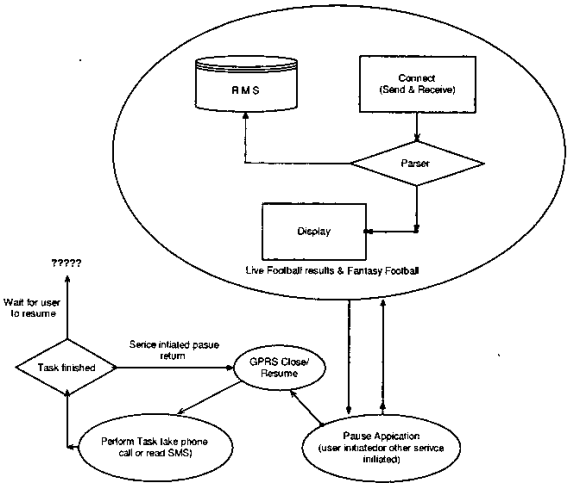

Fig. 2. Application Structure created in Java

Figure 2 shows the basic components of this application along with its behavior towards the rest of OS. If the phone encounters another service that has greater priority over this application, such as a phone call or SMS ,it will automatically pause the application and take the user to the main display, to either accept/reject the call or view SMS. During this transition the GPRS connection is suspended. Once the new task has been performed the application returns from its pause state and resumes the GPRS connection. If the pause in such an application is initiated by the user the application will simply save its state [7] allowing the user to access the other features of the phone. Even after the user has finished other tasks the application cannot resume itself unless the user goes back to the applications menu and selects to restart (resume the application) which will cause the application to wake up and resume from the last saved state. Now this is the scenario that will be seen most commonly on many java enabled phones which are not currently using Symbian OS. If this application was running as Java over Symbian the phone will use Symbian's multitasking capability and just spawn another process to run the other task.

Figure 3 shows the behavior and structure of the application created for Symbian (Nokia series 60).

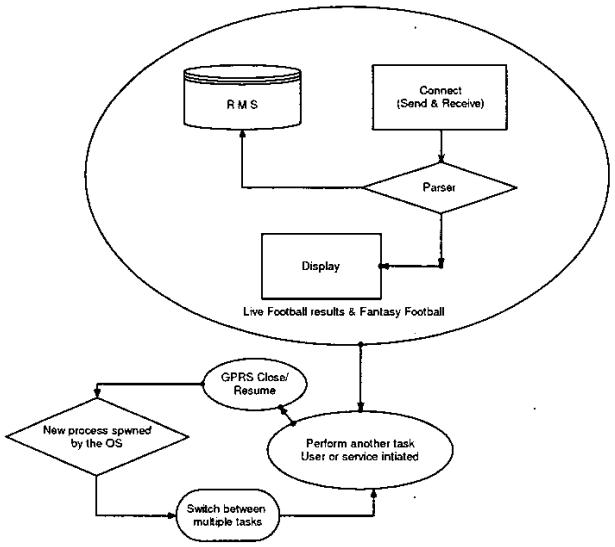

Fig. 3. Application Structure in Symbian (Nokia series 60)

As seen from figure 3 that the main application architecture was kept exactly the same to the one created in java. In this case a user or service initiated task is dealt with by the OS which creates a multitasking environment where the user can switch between different applications by simply holding down the menu key. GPRS connection can stop or resume depending if another application requires the connection.

This is an illustration of the advantages/disadvantages outlined earlier, where major difference comes in the accessibility of other features of the phone. However we can now consider the effect of the system running J2ME on a smart phone using Symbian. As previously discussed the only way for the user to access other services is to make use of the multitasking nature of the Symbian OS. However, this is a more complex procedure for the user than a straight Symbian implementation.

Figure 5 below shows a brief comparison of the two applications running on the same phone under Symbian.

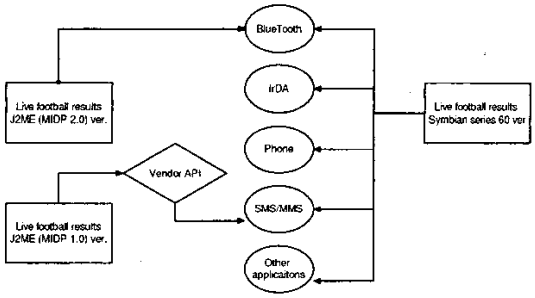

Fig. 4. Service access comparison

As seen the J2ME application has only access to Bluetooth directly (which is only possible in MIDP 2.0 enabled phones) and in order to share information between two midlets they have to be in the same midlet suit. Java based football application has no support for accessing these services from within the application whereas the same application designed for series 60 could access Bluetooth, IrDA, SMS, and MMS [4]. Having access to these other commands and features can be used to enhance the application to produce wider benefits to the user, such as:

- Invite a friend to try this application (SMS)
- Have a league of your own, compete with your friends (Bluetooth enabled fantasy football feature)
- Send this score or event to another friend (SMS enabled feature allowing user easy access to the SMS either to send the score or perhaps even another text)
- Make a phone call (for novice users)
- Read news about your club/league (access to browser)

V. User Interface Comparison

Although mobile phones have come a long way, in terms of being able to run software that was once only available on a PC, there are certain limitations that still affect application developments for instance, screen capabilities, processing power, and memory. For example an application may be required to operate on mobile phones with screen performance ranging from a 176*208 colour display to 100*80 mono displays. All these constraints coupled with a wide range of devices in the market make it a challenging task for the developers to create interfaces that will enhance the user experience. Consumers will definitely use an application that is easier and friendly to handle on the move. The live football results application being considered in this paper [1] has to have a unique, user friendly interface that can be used with no additional effort by a user on the move. This provides a distinctive opportunity to compare this application on the basis of user interfaces on both Symbian and J2ME.

The application was tested on various mobile phones with different screen sizes and resolution (Nokia 6310i, 8910i, N-Gage, 3650). The user interface for Java based application is shown in the following screen shots.

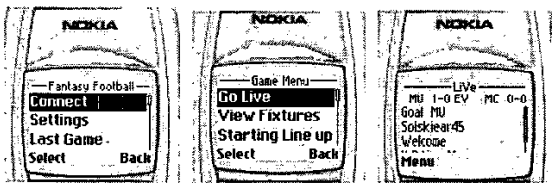

Fig. 5. User Interface on Nokia 6310i (J2ME)

The basic connection and setup options are shown along with live updates screen. The fantasy football team creation and scoring are all done through commands displayed on the screen to the user. The same application is shown running on a Nokia 8910i and Nokia N-Gage in the following figure.

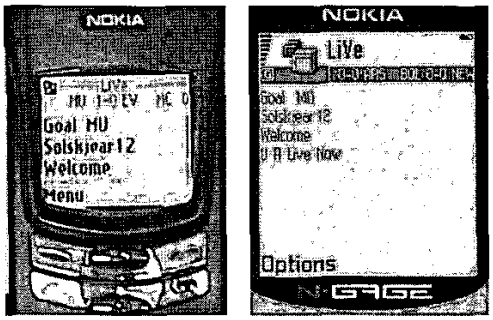

Fig. 6. User Interface on Nokia 8910i and N-Gage

As seen from figure 5 and 6 the application developed in J2ME adapts very well to varying screen sizes. The user interface seen was developed using J2ME's high level user interface [7]. The use of low level user interface was avoided because of the fact that the canvas would not adapt to utilize the maximum screen size available which is a limitation on applications based on MIDP 1.0. The user interface for the Symbian version is shown in the screen shots below.

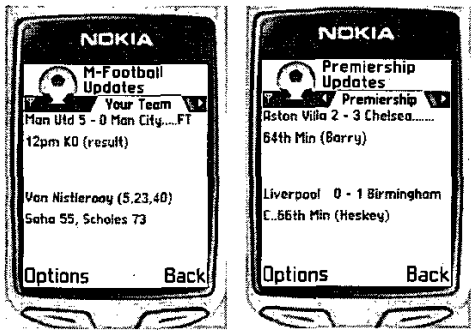

Fig. 7. User Interface for Live updates (Symbian)

The most important aspect of this user interface is the use of tabs which can be used very easily to switch between different screens which is not possible with high level user interface in J2ME. A further improvement to its J2ME counter part the Symbian version allows the user to select one particular club that he/she follows. This results in the selected club's updates to be displayed in separate screen so that the user does not have to switch between screens even though it is very easy. The fantasy football team creation becomes easier with the user of sub menu system which is not available in J2ME. This allows the user to select a particular striker by accessing the team creation menu and then the striker sub menu. The figure below contains screen shots of the fantasy football team setup and application menu to access other features of the phone without having switch between different applications.

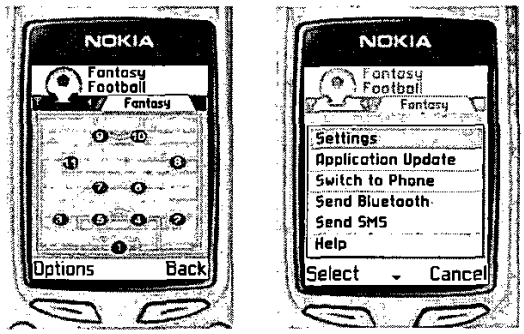

Fig. 8. Fantasy Football and Application menu on Nokia 3650 (Symbian)

The application menu provides an easy access not only to the M-Football application settings but also provides access to other services on the phone. This is not possible with an application created in J2ME even though it is running on Symbian OS. Better control over how the objects are positioned on the screen makes it easy to display and update the user's fantasy football team formation. When taking into account the fact that this fantasy football setup and layout provided on the move is very similar to the one the user is accustomed to on the internet, enhances the experience and ease of use.

VI. Conclusion

By developing the application in J2ME and Symbian we can give general developments guides to prospective developers. As a general guideline software engineers have to decide which route will suite their particular requirements, even though it can be argued, in general, that Symbian based applications provide much more than J2ME based applications it always comes down to what is required for example speed of production (time to market), memory usage, power usage, etc.

For example J2ME can be very useful when the application size needs to be small and it needs to be deployed OTA to mass market. On the other hand Symbian based applications can be very useful when the application requires more memory and higher processing power. Further Symbian can run much faster and will generally consume less device power than their J2ME counterparts. Overall both have a place in the market and present useful alternatives for applications developers.

References

[1] P. Coulton, O. Rashid and H. Ahmed "Live information update services using GPRS" proceedings of IEEE ICTTA 04, Damascus Syria, 19-21 May 2004.

[2] M. J. Jipping, *Symbian OS Communications Programming*, John Wiley & Sons Ltd: West Sussex, 2002

[3] V. Piroumian, *Wireless J2ME Platform Programming*, Sun Microsystems Press, California USA, April 2002.

[4] Digia Inc. *Programming for the Series 60 Platform and Symbian OS*, John Wiley & Sons Ltd: West Sussex, 2003, pp 24,32

[5] R. Barbagallo, *Wireless Game Development in C/C++ with BREW*, Worldware, 2003

[6] W. John, *Core J2ME Technology and MIDP*, Sun Microsystems Press, California USA, January 2002.