Product-focused Process Elicitation for Mobile Application Systems: A Wireless News Browser Case

Vesal Hakami and Ahamd Abdollahzadeh
Intelligent Systems Lab, School of IT and Computer Engineering
Amirkabir University of Technology
PO Box 15875-4413 No. 424 Hafez St., Tehran, Iran
{vhakami, ahmad}@ce.aut.ac.ir

Abstract

The heterogeneous nature of mobile application systems, immaturity of the associated technologies, and insufficient knowledge about the effects of different methodologies denote the need for scrupulous adoption of a life cycle process in developing m-commerce systems. This paper promotes project-specific solutions as a basis for wide-ranging process elicitation efforts. The concept is elaborated by presenting a case study highlighting the best practices for designing and building a wireless news browsing application using a specialized recipe derived from an iterative-based framework. We apply a subset of UML diagrams for modeling the system; moreover, with respect to the high availability of competitive products and close comparability in mobile applications market, we have adopted a highly iterative process model with parallel execution of iterations to speed up our development strategy. Our product incorporates many of the ideas in the already existing feed reader applications to deliver the right mix of features and functionality.

Keywords- M-commerce Systems, Process Elicitation, Unified Modeling Language, Wireless News Browser.

1. Introduction

The successful construction of wireless applications is challenged by several of software's nastiest development and deployment landmines: rapidly emerging standards, volatile platforms, intermittent connections, varied devices, and inconsistent input technologies [1]. Besides these infrastructural constraints, developers face increasing time-to-market pressure and higher quality expectations, with customers insisting on more features with each new release of a product [2]. In addition to failure in responding to the growing demand for accelerated delivery, most project breakdowns in mobile applications context are due to incorrect or misunderstood user's requirements which result in wrong architectural modeling or undesirable system behavior [1, 2].

In this context, the insufficient knowledge about the appropriateness of the existing development procedures contributes to the dilemma. There is no single definitive recipe for developing all variants of m-commerce systems and the project managers or software developers have to customize existing and known processes to suit their particular needs. Since a consensus within the business and industry of such applications is still in its infancy, one good solution would be to elicit a product-focused development approach from the suggested wireless applications frameworks which are mostly based on iterative process models.

This case study undertakes the task of modeling and developing a service-based mobile news browser which encompasses most of the features presented by the already-existing online as well as wireless feed reading applications. The implementation is based upon the technical-level criteria outlined in the framework offered by Kemper and Wolf [4] which provides initial guidelines on selecting an appropriate iterative process for developing a particular mobile system. The final version consists of two Java MIDlets called Service Manager and Explorer operating on the client side. The system makes use of the NewsIsFree [3] syndication service endpoint to find feeds matching best with a user-supplied query and retrieve their latest associated news items.

To address the existing time-to-market pressure effectively, we have adopted a highly iterative approach along with short development cycles and time-boxed execution of iterations. Also, to achieve a high level of system understanding and improve technical accuracy in requirements and interface specifications, we have applied UML diagramming practices in modeling the system. The Unified Modeling Language has proved handy in providing a

Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE
International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06)
0-7695-2511-3/06 $20.00 © 2006 **IEEE**

standardized notation for describing object-oriented systems, and our choice of J2ME as an object oriented Java platform causes the UML notations to be effectively applied.

The paper is organized as follows: Section 2 discusses the background information and related research in the areas of UML, iterative process models, and feed reading systems. The details of our development strategy are outlined in section 3. In the next part, the empirical study with regard to the analysis and design artifacts is described. Consequently, sections 5 and 6 lay out our implementation environment as well as the testing and deployment activities followed by section 7 which is dedicated to a brief analysis of results. The paper ends with conclusions.

2. Related work

Some background information on the adopted strategies and technologies are presented in this part.

2.1. Unified Modeling Language

The Unified Modeling Language incorporates ideas of a large group of software practitioners who decided to leave any agreement on a reference process until later and to confine the UML to being a modeling standard [5]. Booch (1999) has described UML as a language for: visualization, documentation, specification, and construction. Out of these uses of the UML, it is primarily the visualization which helps improve the description of a solution within a project domain [6]. UML has been successfully used with Java in many projects with diverse architectures; however, considering UML as part of mobile development strategy is still a new proposition [2].

In this paper, we have used unified modeling principles to increase formality and precision in requirements and interface specifications. With this end in mind, the use case, sequence, and component diagrams are selected to analyze the user's requirements, clarify the interactions between system objects, and specify the relationships among components. Moreover, a class diagram is used for documenting the static structure of classes.

2.2. Iterative development in mobile context

It is generally accepted that classical, sequential process models are not appropriate for the development of modern market-oriented software systems [4]. In order to catch up with the rapid pace and unceasing change inherent in the wireless industry, many of today's most successful mobile application vendors have adopted iterative and incremental processes as their development strategy [1]. However, considering the exceptionally heterogeneous nature of mobile application systems regarding their technical complexity, user's requirements, and functionality, it is not judicious to prescribe a single reference iterative-based process for the professional development of all variants of wireless systems.

Based on this practical viewpoint, Kemper and Wolf [4] have conducted an analysis of five predominant groups of mobile systems. On the basis of the results of this analysis, they have characterized the scope of m-commerce applications by three dimensions: degree of innovation, speed of development, and risk. Each dimension is rated with the values low and high that have to be checked situational in a project-specific way. Combining these dimensions, one gets eight different types of m-commerce applications ranging from highly innovative, time-critical, and high-risk implementations (type 8) to systems with lower degree of novelty, long life cycles, and lower degree of risks (type 1). With reference to these three dimensions, a framework is developed which allows the typification for a planned mobile software system according to the eight defined types. Finally, there are some recommendations and guidelines that can be adopted in an application-specific manner.

In the context of our implementation, the already-existing online news services and RSS aggregators on the one hand, and the close comparability to the current wireless versions of feed readers on the other, would imply a lower degree of innovation but a higher speed of development and the low level of investment would also imply a low value on the risk dimension (type 2). Table 1 elaborates the indicators of type 2 systems along with their two technical level criteria.

Table 1. Indicators of type 2 applications

Dimension		Technical Level Criteria	
Degree of innovation	Low	T3	Concurrency of development blocks
Speed of Development	High		
Risk	Low	T4	Precise specification of interfaces

The initial guidelines offered by this framework have been followed to adopt and then customize an appropriate development process.

2.3. News browsers

With the advent of RSS[1] technology, a wide variety of clients can easily obtain news summaries from various feeds around the Internet and present them in different ways to the user. Syndicated content includes such data as events listings, news stories, project updates, excerpts from discussion forums, or even corporate information [7]. Porting this technology to

[1] RSS stands for Rich Site Summary or Really Simple Syndication and is an XML-based standard for syndicating headlines and article abstracts.

handheld devices enables the users to keep up with the latest headlines concerning their favorite sports teams, weather forecasts [8], and news channels whilst on the go. Using RSS removes the need to directly visit a weblog or a news site. It also keeps the user's mail box from getting clogged up by spammers and guarantees virus-free contents on delivery to the subscriber [9]. Today, many websites offer RSS feeds for their content and there are also some online services designed to provide access to thousands of news sources on a wide variety of topics with a powerful and flexible indexing mechanism. Examples of such services include Syndic8 [10], Meerkat [11], and NewsIsFree [3] which act as RSS search engines and keep a reference list of validated sources.

There are many online as well as mobile news aggregators available in the market [12~14] and hence the extent to which the requirements for a new version of a feed reader can be specified in advance is relatively high; perceptively, we have based our implementation on the ideas presented by the earlier systems and in order to achieve competitiveness, support for reading all feeds in UTF-8 and UTF-16 encoding schemes is included; further, it provides users with a remote subscription capability to the syndication web service offered by NewsIsFree, currently featuring some 20,000 news channels, ranging from profile news sources, to very specific topic sites, to weblogs [3].

3. Development strategy

The technical-level criteria offered for type 2 systems suggest working iteratively and at the same time concurrently on the development blocks. It also points out the criticality of a precise specification for interfaces. In order to make the speedy launches of executable systems possible, we have applied time-boxing as a key strategy for rapid application development. The time-boxing process [15] combined with the concept of pipelining allows parallel execution of iterations in a structured manner, resulting in a reduction in the cycle time for deliveries. Figure 1 depicts the adopted procedure.

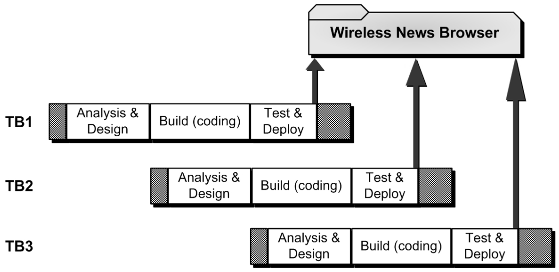


Figure 1. Developing in a time-boxed fashion

The system is developed in three equal-length iterations (time-boxes) each having three stages: analysis and design, coding, and test and deployment. The coding stage takes up most of the time in every time-box resulting in slack times for the first and third stages; however, as figure 1 suggests, by organizing the slack times of the first stage in the start and the last stage towards the end, the resource wastage has been minimized. The following describes the intended scope of the system for each iteration:

- **Time-box 1** aims at providing the core functionality of the final product in terms of communication with a remote web services endpoint. The deliverable for this initial iteration is a wireless client capable of searching for the desired feeds through the syndication database.
- **Time-box 2** provides support for headline exploration; this process makes use of the existing feeds list to browse the news stories associated with a particular feed.
- **Time-box 3** enhances the system capabilities by adding competitive features like e-mail transmission, remote service subscription, and internationalization support.

To maintain a high level of functional precision and performance of the product throughout a tight release schedule, an equal attention to the internal quality and conceptual integrity of the software is inevitable. To serve this purpose as well as to stick with the second criterion for type 2 systems, we have incorporated the modeling principles of UML into our development methodology. With UML 2.0 architectural diagrams like class and component diagrams, the interface definitions are compact and precise; also, they very clearly specify the messages that together form the interface, and the data content of those messages [16].

4. System analysis and design

This section lays out the artifacts of our study as well as the capabilities of the final product. A detailed empirical study based on the above stated research is presented here.

4.1. Designing with UML notations

The system modeling is performed using use-case, sequence, class, and component diagrams offered by the Unified Modeling Language.

4.1.1. Use case diagram. Use case diagrams describe what a system does from the standpoint of an external observer [17]. In each development iteration, a subset of the required use cases is delivered.

Figure 2 illustrates the use case diagram for the actions that the users can perform via the service manager MIDlet. There is only one actor in this diagram which is designated by the stick figure notation. This actor begins and controls the program.

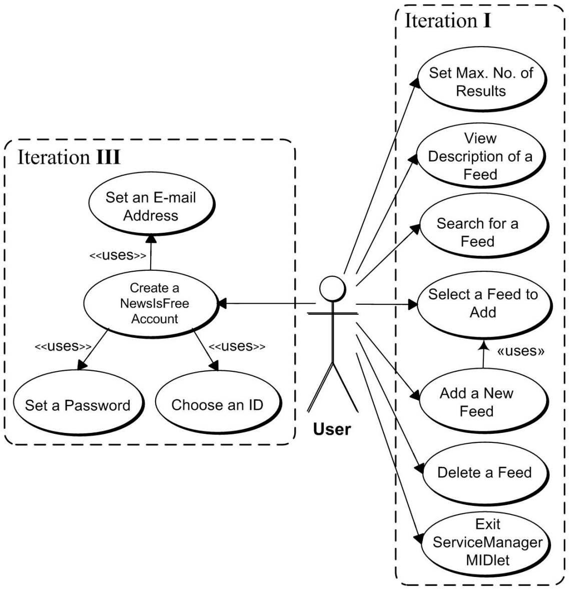


Figure 2. Service manager use-case diagram

The service manager MIDlet enables users to perform a text-based search to find their favorite feeds by specifying a query string and the maximum number of results to be shown on the display canvas. Moreover, in order for the users to make sure a feed best matches their preferences, a brief description of the sources can be viewed separately in another screen. Finally, the user can select best results to be saved for future browsing through the explorer. This MIDlet also allows the users to create a NewsIsFree account via their handheld device. As with the regular service, they will receive by e-mail a confirmation link that needs to be activated before the account can be used.

Figure 3 represents the use case diagram for the explorer part of the application. The following are within the capabilities of this MIDlet:

- Selecting a feed to browse its top stories
- Specifying the maximum number of items to be retrieved and shown on the screen
- Viewing a description of a particular news item
- Selecting headlines and have their associated URLs e-mailed to a certain address (user can view the full version of the article in his/her desktop later)
- Comparing stories belonged to different feeds
- Setting an appropriate font size to adjust data representation on the screen

A noticeable feature of these two diagrams is the use of system boundary boxes to identify which use cases will be delivered in each iteration of the system.

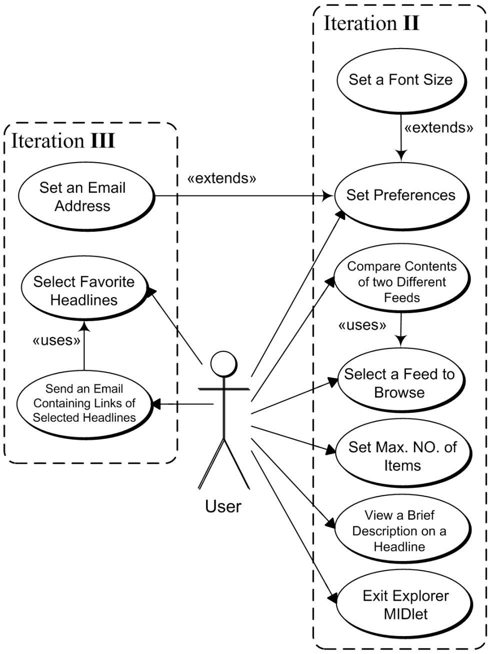


Figure 3. Use case Diagram of Explorer MIDlet

4.1.2. Sequence diagram. This diagram has been used to clarify the core of the interactions between system objects. Figure 4 shows the details of news retrieval mechanism from the moment the user chooses a feed until the system receives an XML-RPC response message and uses it to display the parsed items on the screen.

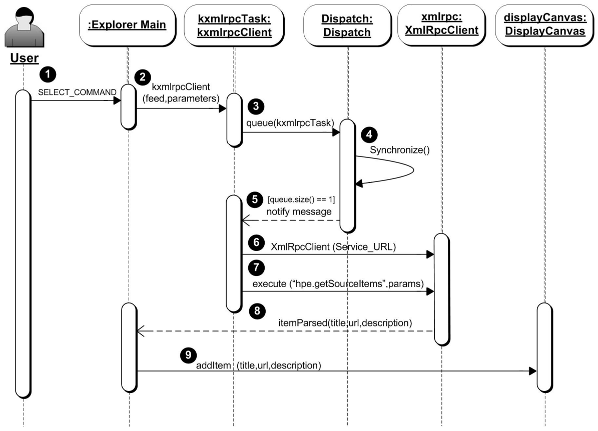

User
:Explorer Main
kxmlrpcTask: kxmlrpcClient
Dispatch: Dispatch
xmlrpc: XmlRpcClient
displayCanvas: DisplayCanvas
1
SELECT_COMMAND
2
kxmlrpcClient (feed,parameters)
3
queue(kxmlrpcTask)
4
Synchronize()
5
[queue.size() == 1] notify message
6
XmlRpcClient (Service_URL)
7
execute ("hpe.getSourceItems",params)
8
itemParsed(title,url,description)
9
addItem (title,url,description)

Figure 4. Sequence Diagram for Browsing a Feed

4.1.3. Class diagram. Being the most important entity in modeling object-oriented software systems, it is used to depict the classes and the static relationships among them [18]. It also represents the structure of interfaces and their functions in the program. Figure 5 shows the client-side class-diagram.

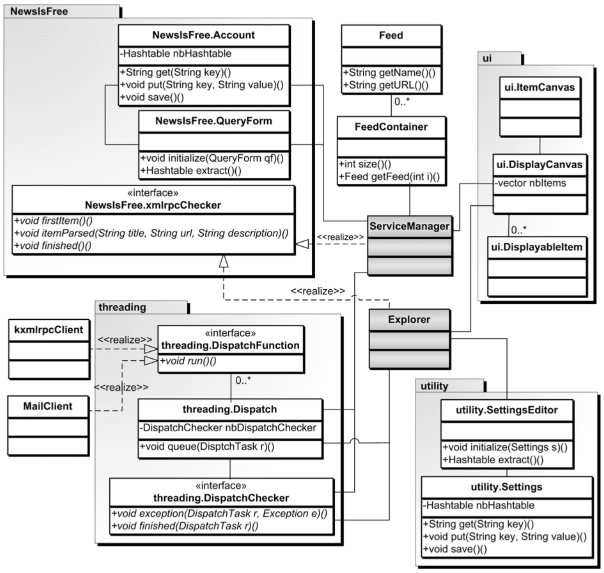


Figure 5. Client-side Class Diagram

There are two main classes in this application: ServiceManager and Explorer both as Java MIDlets. These two control all the other classes, respond to the user's input from several different screens and accordingly designate the next operation. The program groups the classes and interfaces into five packages as described below:

Default package consists of classes which are specific to the main MIDlets. Instances of the Feed class represent the RSS feeds used by the explorer to view the headlines. FeedContainer maintains a list of feeds in a J2ME recordstore structure.

NewsIsFree contains the necessary classes which help the user enter the required data for creating a new account on NewsIsFree syndication server. The package also contains a listener class called xmlrpcChecker which is supposed to be notified whenever a connection or a parsing event occurs.

threading allows multi-threaded execution of the different tasks by maintaining a queue of functions each running in a single separate thread. There is also a DispatchChecker class which acts as a registered listener and is notified by successful or abnormal termination of each DispatchTask.

ui package contains generalized GUI components and handles display of the list of headlines and of the details of any article selected by the user.

utility provides a settings class to store the user's e-mail address and a chosen font size which are used by the Explorer MIDlet for relaying messages to the associated e-mail server and adjusting data representation on the screen respectively. There is also an editor class to let users modify these settings arbitrarily.

4.1.4. Component diagram. Component diagram is one of UML's architectural diagrams used to effectively describe complex architectures as a hierarchy of components (subsystems) communicating through defined interfaces [16]. Figure 6 illustrates the interactive relationship among the components of the news browser system in both client and server sides.

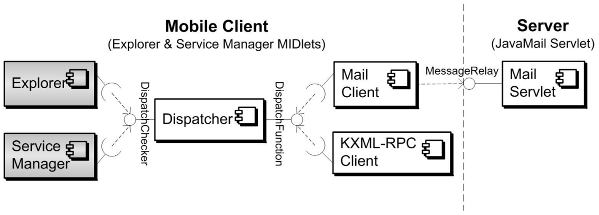


Figure 6. High-level Component Diagram

5. System implementation

This section lays out the details of system implementation with regard to the required software configurations and development environments.

5.1. Configuration/implementation environment

The system was implemented using Sun Java Studio Mobility 6 2004Q3 and the supplied emulators. The service utilizes the following programming technologies:

- CLDC 1.0
- MIDP 1.0
- Java Standard Edition 1.4.2
- JavaMail API (from J2EE)
- Jakarta Tomcat 5.0
- KXML-RPC (the J2ME implementation of XML-RPC protocol built on top of the KXML parser)

The final program for the client consists of two files: the NewsBrowser.jar and NewsBrowser.jad, together they have size of 170 KB without using obfuscator (see Wireless Toolkit documentations). The servlet consists of one java class.

5.2. Communication architecture

Figure 7 depicts the system architecture in terms of network connections and the internal client structure.

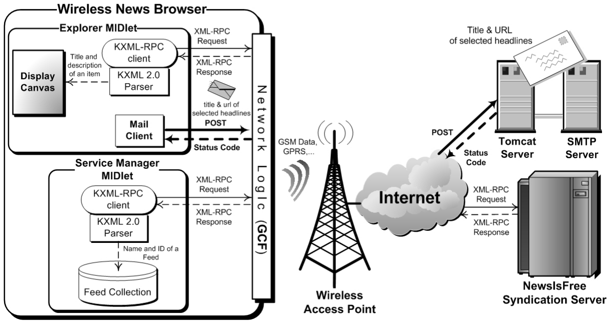


Figure 7. Overall Structure of the Service Environment

Two different HTTP connections may be established depending on the processing procedure of the program:

1- The client communicates with NewsIsFree service endpoint by sending and receiving XML-RPC requests and responses. The application MIDlet makes use of KXML-RPC [19] in its communication with the server. This implementation of XML-RPC empowers J2ME devices with an extremely lightweight mechanism for exchanging data and invoking web services, in a neutral, standardized format. In order to add support for feeds with Unicode encoding scheme, the code for the original implementation of KXML-RPC has been modified.

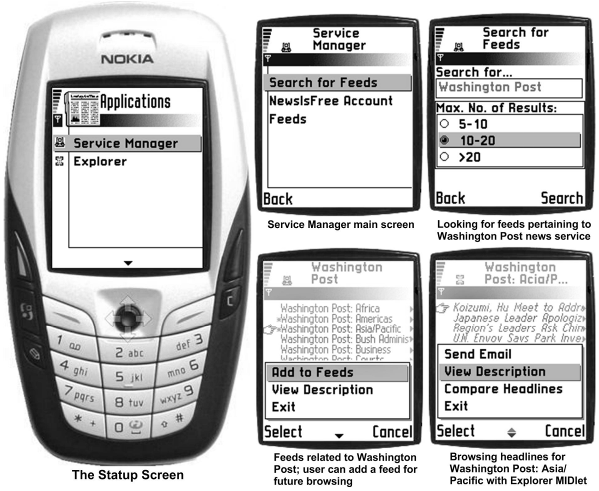
2- In order for the client to be capable of sending reminder email messages, a servlet has been written using packages in both J2EE and J2SE which runs over a Tomcat server. When the message is redirected through the operator's network to the server, the servlet receives incoming requests and in turn contacts the associated email server to which it relays the message.

6. Test and deployment

Various test cases are crucial for developing high-quality MIDlet suites. It is regrettable that the emulator can never equal the real device and debugging on a device is considerably difficult without any tracing possibility [20]; however, using an emulator in conducting network communications tests facilitates constructing and repeating specific test scenarios. It also renders the performance tests faster and cheaper to carry out. In this project, we made use of a variety of emulators ranging from the ones bundled in the Studio to the vendor-specific virtual devices like those offered by Nokia, Sony Erickson, and Motorola.

We performed all necessary unit tests using J2MEUnit framework which provides a lightweight approach toward generating and executing test modules for the limited Java environment of J2ME. Also, at the end of each implementation stage, a usability test was conducted using both a Nokia 3650 and a Palm Tungsten|W with the servlet running on a remote host. The feedback from the tests was quite acceptable and it disclosed that the service did have flaws that would have limited the service if it had been released as it was; however, most of the issues and test feedback were aimed at minor annoyances and flawed/missing code, which have been corrected.

The final product is deployable on any Java-enabled handheld device and since the project does not make use of any device specific GUI routines, most devices should be able to display the application screens without any problems. Figure 8 shows a typical running result on a Nokia 6600 emulator.



Service Manager main screen

Looking for feeds pertaining to Washington Post news service

The Statup Screen

Feeds related to Washington Post; user can add a feed for future browsing

Browsing headlines for Washington Post: Asia/ Pacific with Explorer MIDlet

Figure 8. Wireless News Browser in Action

The application suite has been deployed on a Nokia 3650 smart phone [22] and a Palm Tungsten|W pocket PC [23] whose connections to the Internet were made possible by means of a GSM data subscription. The operation of the deployed system on these two devices pointed out that the structure is acceptable and the user interface features respond correctly to the user's input.

7. Analysis of results

We successfully implemented the wireless news browsing system using UML and pipelined iterative development. The rigorous design and analysis phase combined with the quick iterative-based implementation enabled us to achieve a minimum value for the defect density metric while rendering the cycle times as short as possible. The defect rate of the final program code was 0.9 units per 1 KLOC (0.9/1KLOC).

The framework we used gave us good insights into selecting and improving a managerial process. The resulting approach was more effective in implementing our wireless system than if we had adopted a totally ad hoc development recipe.

The standardized language and graphical presentation of the UML diagrams helped improve the overall system's understanding. With the two use case diagrams, one for each Java MIDlet, a high-level view of the system's usage was easily described, and the range of the scenarios was designated in terms of the planned iterations. The sequence diagram elaborates on the news retrieval mechanism and provides a concise way to document the bulk of the interactions among the participating objects. Finally, the two structural diagrams (class and component) have made it easy to specify architecture and interfaces.

8. Conclusions and future work

This paper described a case study regarding the design and implementation of a wireless news browsing system in a product-focused manner. The development recipe was elicited from a framework promoting selective adoption of iterative process models for mobile commerce applications. We designed the system using UML and managed it in a time-boxed fashion which enabled us to achieve equilibrium in time and quality. Further, the system implemented a selection of the ideas presented by the earlier comparable products along with a tinge of competitive features. The implementation environment and the tools used, provided excellent support for the successful development of the system.

Our plans for future work include a more extensive study involving a more comprehensive process elicitation effort. We continue to evaluate product-focused approaches in other case studies toward a reference process spanning a specific family of mobile application systems. Working out a product-line architecture for all systems within the domain of the family is also fully intended. Further, since considering UML as part of mobile development strategy is still a new proposition, we also plan to evaluate empirically the unified modeling principles over the next research steps. A longer evaluation time will provide more data on the useful aspects of UML and hence will promote its usage in future projects as well.

References

[1] R. Holler, "Mobile Application Development: A Natural Fit with Agile Methodologies," Technical Report, VersionOne.

[2] B. Graham, "Developing embedded and mobile Java technology-based applications using UML," White Paper, IBM Developerworks, Nov. 2003.

[3] NewsIsFree Home Page, Available at: www.newsisfree.com. Last visited on Oct. 2005.

[4] H.G. Kemper and E.Wolf, "Iterative Process Models for Mobile Application Systems: A Framework," *23rd Int'l. Conference on Information Systems (ICIS '02)*, Barcelona, Spain, December 2002, pp. 401-413.

[5] M. Fowler, *UML Distilled: a brief guide to the standard object modeling language*, 3rd ed., Addison-Wesley, 2004.

[6] B. Unhelkar, "Applying the UML to Enhance Quality of Web Services," *OMG's 2nd Workshop on Web Services Modeling, Architectures, Infrastructures and Standards*, Philadelphia, PA USA, April 2003.

[7] M. Scotto, A. Sillitti, G. Succi, T. Vernazza, "Collecting and integrating news from heterogeneous services," *The 9th World Multi-Conference on Systematics, Cybernetics and Informatics (SCI 2004)*, Orlando, Fl., July 2004, Vol. 5, pp. 56-59.

[8] "GSM Drives the Business Opportunity for Consumer Wireless Applications," White Paper, 3GAmericas, Nov., 2004.

[9] W. Richardson, "RSS: A Quick Start Guide for Educators," ZDNet IT White Paper, ZDNet, July 2004.

[10] Syndic8 Home Page, Available at: www.syndic8.com. Last visited on Oct. 2005.

[11] MEERKAT Home Page, Available at: www.oreillynet.com/meerkat. Last visited on Oct. 2005.

[12] NewsMob Home Page, Available at: www.newsmob.com. Last visited on Oct. 2005.

[13] MobileRSSNewsReader:FreeNews, www.masternewmedia.org/news/2005/04/19/mobile_rss_news_reader_freenews.htm

[14] NewsGator Home Page, Available at: www.newsgator.com. Last visited on Oct. 2005.

[15] Egress Home Page, Available at: www.garishkernels.net/egress.html. Last visited on Oct. 2005.

[16] P.Jalote, A.Palit, P.Kurien, V.T. Peethamber, "Timeboxing: A Process Model for Iterative Software Development," *Journal of Systems and Software (JSS)*, Volume 70, Number 1-2, February 2004, pp.117-127.

[17] A. Gurd, "Using UML™ 2.0 to Solve Systems Engineering Problems," White Paper, Telelogic, July 2003.

[18] R. Miller, "Practical UML™: A Hands-On Introduction for Developers," White Paper, Object Mentor Publications, 1997.

[19] R. C. Martin, "UML Tutorial: Part 1 -- Class Diagrams," Object Mentor Publications, www.objectmentor.com, 1997.

[20] KXML-RPC Home Page, Available at: http://kxmlrpc.objectweb.org. Last visited on Oct. 2005.

[21] M. Kozakai, Jabber/J2ME based Instant Messaging: Client for Presence Service, M.Sc. Thesis, University of Applied Science Stuttgart, June 4, 2004.

[22] Nokia 3650 phone features and product specifications, www.nokiausa.com/phones/3650.

[23] Palm Tungsten Pocket PC features and specifications, www.palmone.com/us/products/handhelds.

IEEE COMPUTER SOCIETY