

Mobile Mapping Applications over J2ME Enabled Phones

Yesim Tunccelik[†] and Kivanc Dincer^{††},

[†]HAVELSAN Eskisehir Yolu 7.km
06520 Ankara, TURKEY

^{††}TUBITAK-UEKAE/G222 Unit, Atatürk Blv. No:211/20-21
06680 Kavaklıdere, Ankara, TURKEY

Summary

In this paper, a mobile mapping application called JMMA (J2ME-Based Mobile Mapping Application) is presented to show that geographical information and mapping applications can be effectively delivered over J2ME-enabled mobile phones that have got limited capabilities. JMMA is a client/server network application runs on a user's mobile phone and connects to an Internet enabled database server machine thru a wireless connection. The application software, which persists on the client, was developed using the Sun's J2ME Mobile Information Device Profile (MIDP). Borland J2ME Wireless Toolkit (WTK) was used as an emulation and testing environment. The result of the work proves that it is possible to implement Location Based Services and mapping applications over J2ME-enabled mobile phones and gives some important design and implementation hints for the followers.

Key words:

Mobile mapping applications, J2ME MIDP, J2ME-enabled phones, mobile mapping, LBS.

1. Introduction

The first generation mobile phones, which used analog technology, were heavy and their coverage was patchy. Technical innovations in terms of hardware, software and protocols have contributed to the success of mobile phones and added them new capabilities. The handsets have become smaller and lighter, the battery life has increased, the reception has improved due to the improvements in digital technology and better use of the finite spectrum. The current generation of mobile phones is built using the digital technology and they connect to the Internet via new generations of wireless communications. Therefore they provide higher information transfer capability than the earlier generations.

On the other hand, business and personal lifestyles are changing and evolving very fast. This calls for more flexible products and services that facilitate and support our new routines. In the mobile information society, delivering targeted and timely information and services has become essential.

The convergence of multiple technologies including Geographic Information Systems (GIS), Internet, wireless communications, location identification and mobile devices has given rise to exciting new types of information utilities that may be referred to as Location Based Services (LBS) [8]. LBS provide the capability to find the geographical location of the mobile device and then to offer services based on this location information.

To deliver such mobile services and content to wireless devices, companies have developed new platform technologies such as i-mode [5, 6], Wireless Application Protocol (WAP) [7] and Java 2 Micro Edition (J2ME) [15]. Based on these new enabling technologies, many mobile service applications have been reported in recent years. Minoi et al. [1] developed a shortest path navigation application, which was an extension of a location-based mapping system. It aimed to provide WAP phone users immediate access to the road information supplied as a bitmap map enabling them to navigate around the city and to find the shortest path between locations. Liu [2] developed a mobile client/server tourist map viewing application for Palm computers with integrated GPS devices where the provided maps are in GIF image format. Revel [3] developed an intelligent tourist application on a Palm device and tested the effects of using bitmap versus vector data formats for maps. Varrantaus et al. [4] developed an LBS application for Java-enabled phones and Personal Digital Assistants (PDA) that offered navigation services based on vector maps.

There are also several development efforts on mapping applications for laptops, PDAs and smart phones but our work concentrates specifically on J2ME-enabled mobile phones. Based on currently available enabling technologies, a new client/server mobile mapping application was developed for J2ME-enabled phones and presented here to identify a highly appropriate programming language and wireless application platform for these purposes. We here present a proof-of-concept application that shows mobile mapping applications, which are among the most popular location-based services (LBS), can be effectively delivered over mobile phones.

The rest of the paper is structured as follows. Section 2 gives an introduction to the architecture of the mobile mapping application, JMMA, developed for J2ME-enabled mobile phones. Section 3 introduces the services supplied by this application. Section 4 investigates the performance issues. Finally, Section 5 concludes the paper.

2. A J2ME-Based Mobile Mapping System (JMMA)

Location-based services (LBS) are especially used in special circumstances such as while driving a car; thus the user has not to be piled with large amounts of information that may distract her attention. A mobile environment imposes strong constraints on the implementation of LBS. Mobile devices have got limited memory and computational power, small screen size with low resolution, short battery life and uncomfortable input tools that usually do not allow a user to operate quickly. Besides, mobile networks have got high cost, limited bandwidth, high latency, low connection stability and low predictable availability. By taking all these constraints into consideration along with users' high performance and good service demands that conflict with them, we have developed the JMMA tourist mapping application for Java enabled mobile phones.

JMMA is a client/server network application running on the user's mobile phone and gets the required data from a database server on the wireless network (Fig.1). HTTP is assumed as the underlying protocol for communication between the client and the server. The graphical user interface of the client contains a canvas for displaying the map and has zoom in, zoom out and pan features. JMMA uses vector data format for city maps because vector graphics are better than bitmap images for querying, manipulating the data, e.g., scale and rotate operations. The server side hosts the main database on which queries are executed by using SQL (Structured Query Language), but some simple queries and computations can be delegated to the client. JMMA server runs on a workstation. ASP (Active Server Pages) is used to reach the database via web pages and form dynamic HTML pages that uses the data taken from the database. Client-side software was developed in Sun's J2ME CLDC MIDP [11] environment. J2ME applications persist on the client and can function on the client device even when the device cannot connect to the GSM network. Network drop outs are inconsequential when the application persists on the device; capable of caching newly entered data until a new connection is established.

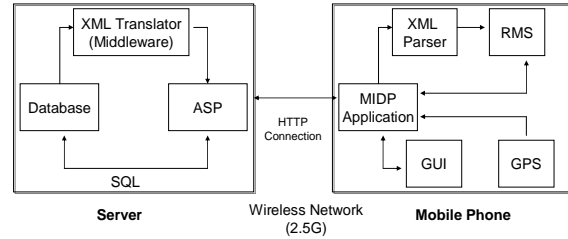


Fig. 1 JMMA client-server architecture and main software components.

2.1 Emulation Environment

Borland J2ME Wireless Toolkit [12] was used as the emulation environment. Borland JBuilder 7 and Mobile Set 3 IDEs were used as the development environment of the client side of JMMA. Mobile Set 3 includes many wireless toolkits, J2ME Wireless toolkit was adopted for this work among them.

2.2 Architecture

JMMA's client software resides on a mobile phone featuring small memory and display, low performance and network bandwidth, less functionality and less user-friendly input methods. The server resides on a workstation on the GSM network that has large memory and storage and fast processing ability. Fig.2 below shows the basic JMMA client/server functions. This application concentrates on a single server and a single client model.

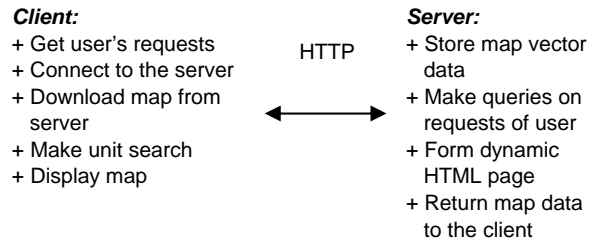


Fig. 2 JMMA client/server functions.

2.3 Data Formats

Borland's Interbase database system [16] was used to store the map data at the server side. Interbase is a quite small but powerful system that runs on several platforms (including Windows and Linux) and was released under GPL License.

In JMMA the geographical data is transmitted to the client in vector form, instead of bitmaps (raster form). Vector maps have many advantages such as allowing the linkage

of spatial data and attributes and inclusion of map layers, while requiring less storage space.

JMMA is capable of storing much more information than a paper map or what can be shown on the screen at once. Two general types of information have been included in the application: They are the geographic information that provide the position and shape of each map feature; and the attribute information that provide additional non-graphic information about each feature, such as a road's name associated with it.

2.4 Graphical User Interface (GUI)

Graphical user interface requirements of software for mobile phones are different from those for desktop computers. For example, the display size of mobile phones is small and input devices do not always include pointing tools such as a mouse or pen input. Therefore, the user-interface programming guidelines for applications running on mobile phones are not the same as for the desktop computers.

MIDP applications are expected to run on many different mobile phones without modification. This is particularly difficult in the area of user interface; Mobile Information Devices (MIDs) have got screens of all sizes, in grayscale and in color. Furthermore, for input, the MIDP is fairly open-ended. The input devices on MIDs vary widely in their abilities, from numeric keypads to alphabetic keyboards and soft keys to touch screens. The minimum screen size mandated by the MIDP is 96 x 54 pixels, with at least one bit of color depth.

Below there are some important guidelines that we kept in mind while designing the JMMA with MIDP graphical API functionality:

- User interface is kept simple and easy to use. Because novice users who probably have not used a J2ME-enabled phone before may use the JMMA.
- High-level APIs are used as much as possible, so that the application became portable across different mobile phones.
- No keys other than those defined in the Canvas class are assumed to exist.
- The application does not assume any specific screen size; instead, it queries the size of the display initially and adjusts itself accordingly.
- Choices are presented to users as lists because entering alphanumeric data through a mobile phone can be tedious.

There is a soft button located at the bottom of the screen. It has no predefined function and can be used to serve different purposes at different times.

2.5 Networking

In the JMMA, to allow the application to be portable across all mobile phones, HTTP connection is used to connect the client to the server. HTTP is a request-response application protocol where the GET or PUT commands are used to supply data. In the JMMA, GET method was used where the data of the request becomes part of (encoded in) the URL. Since the client's requests do not contain large amount of data, there is no risk of overrunning the capability of environment variables.

2.6 Server-Side Database Queries

In JMMA, once a client sends its request over the network to the server, it is then up to the server to interpret the request and generate an appropriate response. The JMMA's receiving end on the server is an ASP (Active Server Pages) page which is a regular HTML page with two elements added: First, the file's name ends with the ".asp" extension, so that the web server knows that it should be interpreted specifically. Second, the file includes some server-side code instructions, such as getting the current date, that the server should perform before sending the page to the client side.

In the JMMA's main menu shown to the user, a list of available layer names are displayed for the user to select. Once the user makes her selection, the client connects to the server and transmits the user's request parameters, which fires the ASP code execution on the server. The ASP code is used to reach the database and form a dynamic HTML page with the data pulled out from the database. In the JMMA application, ASP code mostly contains SQL statements to make the queries in the database. The resulting data is then sent back to the client.

2.7 Client-Side Database

In order to protect the application from the unexpected disconnections and crashes, the JMMA client application utilizes the MIDP's Record Management System (RMS), a simple record-oriented database, to preserve the recently used map data. This saves the running application from losing information and guarantees basic transactional properties to be supported. When the application is invoked again, it can retrieve data from this persistent database. The RMS uses non-volatile memory to store information. JMMA's read, write, sort and search data operations are done within the RMS.

3. Mobile Mapping Services in the JMMA

The JMMA includes five different types of mobile mapping services each of which exploits some different technologies. However, the common target of all services is to display maps on a mobile device based on the geographical data supplied by the server. We discuss each service in detail below.

Service 1. The target of the first service is to perform basic mapping operations. This service gets and displays the data for the selected layers from the server via an HTTP connection. It is possible to do several operations such as zooming in and out, panning, and searching for a geographical place.

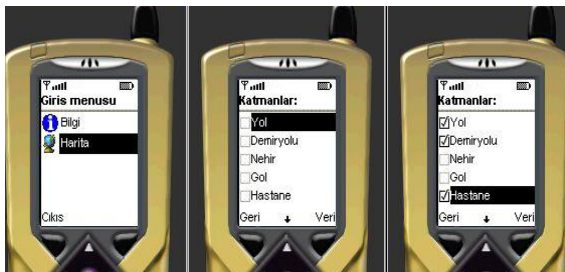


Fig. 3. a) Main Menu b-c) List of layers.

When the service is started, a menu with “Info (Bilgi)” and “Map (Harita)” options is displayed for the user as a list (Fig. 3). If the “Map (Harita)” option is selected, a multiple list containing the available layers that can be displayed on the map is shown. A layer corresponds to any geographical data set stored in the database. Selected layers can be created from the whole map, such as streets, buildings and customers layers. These are the layers that help orient the viewer geographically. The user can check any appropriate boxes to select as many layers as necessary. Of course, because of their complexity and detail, some layers do not appear on the map until a scale threshold is reached.

After selecting the layers to be displayed, the user should invoke the command – “Info (Bilgi)” which opens a connection to the server to get the appropriate data from the database to draw the map. The JMMA uses a user-centered approach; user preferences are taken into account. When the client connects to the server, it also sends the data of user that belong to the preferred layers. After retrieving data from the server to the client’s database, it is presented to the user in the form of a map.

Fig. 4 shows the information screens that are displayed while the client is trying to connect to the server and when the connection is established. The information downloaded from the database is stored in the local database of the device. The status of this operation is also

displayed for the user via a progress bar. The resulting data is translated into a map upon the completion of the download operation.



Fig. 4. a) The progress of data download operation. b) Connection established message. d) Display of stored data on the map.

The application supports basic operations like panning and zooming. When the user selects the “Menu (Menu)” command, a list that includes the “Zoom (Zum)” and “Search (Ara)” commands appears on the screen. With the selection of “Zoom” command the user sees a new list, which includes the zoom in and out features. In Fig.5, these lists and the zoomed maps can be seen. When a zoom in/out is performed, the map is displayed with a larger/smaller scale. If too many features are shown on a map it can create map clutter. If a map shows too many attributes, the colors, patterns and symbols are overwhelming that inhibits the usefulness of visual communication. Therefore, features such as the names of the locations are displayed only if the scale is large enough. JMMA has also the panning feature; the user can pan the map to all four directions.

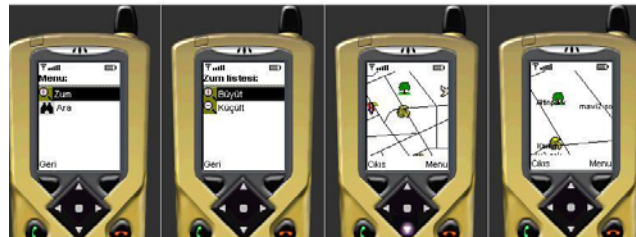


Fig. 5. a) The menu (b) The zoom options (c, d) One and two times zoomed in maps.

The user can start a unit search on geographical attributes by choosing the “Search (Ara)” option from the menu. This is performed on the device’s local RMS-based database. The user enters a unit name into the textbox using the keypad and selects “OK (Tamam)” command. The search is done on the local database and the resulting map is shown with the sought location in red color centered on the phone’s screen. In Fig. 6 below, menu and search type lists are shown. In that example, the user is trying to find the park nearest to her location.



Fig. 6. Doing a unit search in the JMMA.

Service 2. It is assumed that application relies on a GPS device attached to the user's mobile phone to determine the geographic location of the user. As in previous service, the user can retrieve information about a specific point of interest by selecting the "Search (Ara)" command in the Menu which provides the unit search (Fig.6) and nearest search options (Fig.7). If the user is searching for a position, she selects the appropriate command from the list and a new list appears on the screen. For the nearest search, the user enters the sought unit's name and type, like a park or mall, and the user's location are sent to the server which will do the nearest location calculation. The name and the coordinates of the unit are sent back to the client when found. The result is shown in the map with red color and is centered on the screen. The zooming and panning features can be used in the resulting map.



Fig. 7. Doing a nearest unit in the JMMA.

The unit search is performed by the client, while the other queries are delegated to the server application. The processor power of mobile phones usually allows processing only limited volumes of data. On the other hand, execution of queries on the server side causes hard use of the mobile network and even greater latency of queries because of the limited bandwidth of the network.

For these reasons, the mobile device can do basic searching, ordering and filtering operations efficiently. On the other hand, finding the location of the nearest unit is not easy since it can do only a few mathematical operations, namely add and subtract.

Service 3. Here, everything is the same with the above service, except the determination of the user's position. It is assumed that no location identification device or method is used and the user locates her position manually on the map shown on the screen. Zoom in and out, nearest and unit searches are done according to the manually determined position of the user.



Fig. 8. Finding the nearest unit to the manually determined position of the user.

A small square symbol is shown on the screen and the user moves it to the appropriate location that represents his current location (Fig.8). By pressing the "OK (Tamam)" button, this information is sent to the server along with the sought location to do the rest of the computation. Zooming, panning, attribute listing, and nearest point operations are supported in this service.

Service 4. The user can access the attribute information associated with geographical details as well as displaying maps (Fig.9). The list of choices of geographical details is supplied to the user in the menu. For example, if you select the hotels in the list, the list of all nearby hotels is displayed on the screen. Then the user can select one of them to see the info such as the phone number or a picture of the hotel. The search of attribute information is performed on the server as it is not wanted to keep to this information on a mobile device with very limited memory.

Service 5. In the last application, all data is transformed into an XML document and transferred to the mobile device in this format. As in other services, all data is kept in the database on the server and certain search queries are done on this database. The results are translated to XML documents using MS SQL Server's standard utilities. The kXML parser that runs on the client site within JMMA does the parsing of the XML documents received by the device. The parsed data is stored on the local database and used for drawing maps. This service is a preparation for the upcoming XML-based GIS format called GML [17].

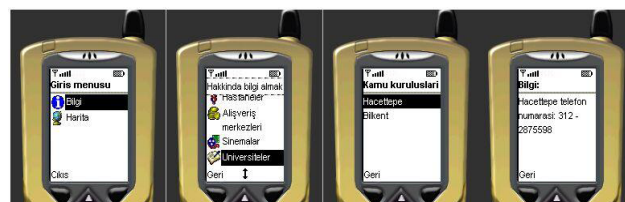


Fig. 9. Querying the attribute information associated with geographical details.

4. Performance Issues

The JMMA is a mapping application which generally uses low-level APIs for drawing and contains many pictures and works with large amount of data. It gets the required data from the server via an HTTP connection over a wireless network. All these features of the JMMA affects the runtime performance negatively. Therefore, we had to handle the following runtime performance issues during our implementation.

Optimization of Canvas Graphics Operations. The runtime demands of the graphical operations used in displaying maps where low-level APIs are frequently used become crucial. In order to increase the speed of canvas operations, a special “Paint” method is used. This method ensures that only the changing delta portion of the map is redrawn on the screen. In this fashion, the number of painted details is reduced and the map drawing operation is sped up.

Reduction of HTTP round trips. It may take a long time to establish a connection between a client and the server on a wireless network. The latency time of HTTP on such networks is quite high as shown in Table 1. Therefore the number of HTTP round trips in the application is kept at minimum. As Internet web browsers make several round trips to get the data associated with a web page, all the data is carried in one round-trip in the JMMA.

Table 1: Typical latency times in GSM and GPRS wireless networks.

	HTTP Round Trip Time	
	First Message	Following Messages
GSM CSD (Circuit Switched Data)	5-10 secs	2-3 secs
GSM HSCSD (High Speed Circuit Switched Data)	5-10 secs	2-3 secs
GPRS	5-8 (plus) secs (depends on the network traffic)	2-4 (plus) secs (depends on the network traffic)

Avoidance of Low Performance Protocols. The XML-based protocol used in the JMMA causes performance penalties due do the total size of XML tags in the communication, the time required for parsing XML files, and the overhead of parser codes. Besides, the size of the JAR file is increased since the XML parser has also to be included in the JMMA application. Therefore, no extra information except the necessary reply is put into the XML document. And a very small XML parser was selected in order not to increase the JAR file size. As seen in different applications of the JMMA, using a binary format to convey messages is preferable in terms of performance as compared to the XML format.

Exploiting the Use of Multithreading. In the JMMA, the client can move on the menus or in the off-line operations while the client is trying to connect to the server, or passing data on the network or parsing an XML document. This decreases the waiting time and keeps the user occupied at most times.

The Size of the JAR file. The JAR file includes all the class files, pictures, symbols, and other source files in the MIDP application. The size of the JAR files is important for several reasons. For example, the memory size allocated for MIDP applications on mobile devices is usually very limited. Therefore, the users prefer small-size applications to better use of the available space. In addition, some wireless network operators put a limit on the size of the MIDP applications that can be downloaded from the network. We took two precautions about this issue:

a. Use of Obfuscator. The obfuscator is a special program that reduces the size of a JAR file significantly by removing the classes, methods and variables that exist but never used/referenced in a compiled Java program and shortening the long class, package, method and variable names. Especially in applications that use an XML parser, size of a parser can be reduced up to 10% by removing the unused parts. On the other hand, the Obfuscator has very little effect on the size for frequently used pictures in PNG format that are part of the JAR file.

b. Use of Libraries. Java applications spend more time in library codes than time used in program codes during execution. Therefore, excessive care was taken not to call libraries, especially the graphics libraries, except absolute necessity within the JMMA application.

Performance Observed by the User. In the JMMA, the user has to wait a little bit while a network connection is established or an XML document is parsed. However, during this delay, a status bar is shown to the user to indicate that the application is still running. Furthermore, the user has the option to cancel the current operation by means of a free thread.

5. Conclusion

Mobile mapping applications are one of the most popular location based service categories. We developed a mobile mapping application called JMMA for Java enabled phones. This work is among the few studies in this area where most of the previous effort concentrate on larger and more capable mobile devices such as PDAs and smart phones. JMMA proves that mapping applications can be supplied successfully on mobile phones with a rich user interface and with high performance with the right selection of technologies. We give many design and

implementation hints in this respect. JMMA builds on a pyramid, where the lowest layer is wireless network technologies and services, mobile devices and software systems. The middle layer is the J2ME technology that made the development of JMMA possible. On top of the pyramid, JMMA is a client/server type of mobile mapping application. It uses wireless network and communication protocols by getting the mapping data from a server and displays maps on the small screen of the phone along with many functions.

References

- [1] Minoi, J. L., Green P., Arnab, S., 2002, Navigation Application with Mobile Telephony: Shortest-path, University Malaysia Sarawak, University of Manchester, Map Asia 2002, <http://www.gisdevelopment.net/technology/lbs/techlbs008pf.htm>
- [2] Liu, J., 2002, Mobile Map: A Case Study in the Design & Implementation of a Mobile Application, Computer Engineering Dept of Carleton University, <http://www.sce.carleton.ca/wmc/chameleon/mc/JaneThesis.pdf>
- [3] Revel, S., 2001, How Does Location Govern Task? The Delivery of an Intelligent Tourist Aide Over a Palm Device, Univ. of Edinburgh, MSc Dissertation Plan, <http://www.geo.ed.ac.uk/~gisadmin/submission/msc0138/plan.html>
- [4] Virrantaus, K., Markkula, J., Garmash, A., Terziyan, V., Veijalainen, J., Katanosov, A., Tirri, H., 2001, Developing GIS-supported Location Based Services, Helsinki University of Technology, University of Jyväskylä, Finland, Second International Conference on Web Information Systems Engineering (WISE'01), Volume 2, Japan, <http://www.cs.jyu.fi/ai/papers/WGIS-01.pdf>
- [5] Megler, V., 2002, i-mode: From bandwidth problem into Internet phenomenon, <http://www-106.ibm.com/developerworks/library/wi-imode/index.html>
- [6] Keryer, P., Nara, T., 2001, i-mode: A Successful Launch of The Mobile Internet Market, Alcatel Telecommunications Review – First Quarter 2001, http://atr.alcatel.de/hefte/01i_1/gb/pdf_gb/12keryergb.pdf
- [7] Developnet, Why is J2ME MIDP superior to WAP, <http://www.developnet.co.uk/wap.htm>
- [8] Dru, M-A, Saada, S., 2001, Location-Based Mobile Services: The Essentials, Alcatel Telecommunications Review – First Quarter 2001, http://atr.alcatel.de/hefte/01i_1/gb/pdf_gb/14drugb.pdf
- [9] Goodman, D. J., 2000, The wireless Internet: Promises and Challenges, July 2000 IEEE Computer magazine.
- [10] Knudsen, J., 2002, What's New in MIDP 2.0, <http://wireless.java.sun.com/midp/articles/midp20/>
- [11] Laitinen, H., 2001, Sun's 2001 Worldwide Java Developer Conference, Development Tools for the J2ME, http://www.forum.nokia.com/main/1,35452,1_0_75,00.html
- [12] JBuilder MobileSet 3.01 Developer's Guide, <http://info.borland.com/techpubs/jbuilder/jbuilder8/me/contents.html>
- [13] Capone, J. M., 2001, Java 2ME Bridging wireless Gap?, <http://www.onjava.com/lpt/a/1068>
- [14] Angelides, J., 2002, The right technology is vital for location based applications, Wireless Europe,

wireless.iop.org, September 2002, http://www.trueposition.com/Download/Whitepapers/09.02_righttechnologyforlbs_wirelesseurope.pdf

- [15] Java 2 Platform, Micro Edition – J2ME, Official Site, <http://java.sun.com/javame/index.jsp>.
- [16] Interbase Database Management System, <http://www.codegear.com/>, <http://www.borland.com>.
- [17] Geography Markup Language (GML) 2.0, OpenGIS Implementation Specification, <http://www.opengis.net/>.



Yesim Tunccekcik received her B.S. degree in Electrical and Electronics Engineering from Ankara University, Turkey in 1997 and M.S. degree in Electrical and Electronics Engineering from Hacettepe University, Turkey in 2003, respectively. During 1997-98, she worked as a system engineer in a consultant engineering company which works for the NATO Infrastructure Division of Turkish

Ministry of Defense. Since 1998 Ms.Tunccekcik has been with HAVELSAN Co., one of the largest defense companies in Turkey, as a test and software engineer since 1998. This work was conducted as part of her M.S. thesis research work under the supervision of Dr. Dincer.



Kivanc Dincer received his B.S. in Computer Engineering from Middle East Technical University, Turkey in 1989, M.S. in Computer Science from Iowa State University, U.S.A. in 1991, and Ph.D. in Computer Science from Syracuse University, USA in 1997. During 1997-99, he worked as an assistant professor in Department of Computer Engineering, Baskent University,

Turkey. Since 1999, he has been working as a Chief Researcher at TUBITAK-UEKAE which is a governmental research institute. He periodically teaches undergraduate and graduate level courses at various universities in Ankara as a part-time instructor. He was the co-advisor of Ms. Tunccekcik's during her M.S. studies at Hacettepe University where this work was conducted.