RC22510 (W0207-023) July 8, 2002
Computer Science

IBM Research Report

Purchase Order Status for Mobile Commerce

Thao N. Nguyen, Kakan Roy, Linh Lam, Thomas Kwok
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Rosa Bolger, Linda Brower
IBM Corporate Webmasters: Technology Innovation and Strategy
17 Skyline Drive,
Hawthorne, NY 10532

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Purchase Order Status for Mobile Commerce

Thao Nguyen, Kakan Roy, Linh Lam, Thomas Kwok
IBM T.J. Watson Research, 19 Skyline Drive, Hawthorne, NY 10532
(914) 784-7101
tnnguyen@us.ibm.com

Rosa Bolger and Linda Brower
IBM Corporate Webmasters: Technology Innovation and Strategy
http://www.ibm.com/webmaster/
17 Skyline Drive, Hawthorne, NY 10532

ABSTRACT

In recent years, there has been a significant increase in purchase order status inquiries due to the exponential growth of electronic commerce. In this paper, we present an implementation of a wireless solution that enables users to look up the status of their purchase orders via a wireless device equiped with a web browser such as cellular phones and PDA devices. In addition, this solution is extended in a prototype to support cellular phones without web browsers by incorporating telephony technology. As a result, order status may be accessed via any conventional telephone in an automated fashion. This solution provides better service to customers through broader accessibility and lower operating costs of customer support centers by minimizing human assisted telephone calls.

1. INTRODUCTION

By way of introduction, let us imagine a business scenario. John Smith is a marketing manager for a firm that plans to announce a major software product. His assignment is to present and demonstrate the new product to potential customers at an Internet trade show. A week before the show, his manager suggested to him that using a large Liquid Crystal Display (LCD) would make a huge difference to his presentation and demo. With the manager's approval, John went shopping on the Internet and purchased a 30-inch LCD and had it shipped directly to the trade show. On his way to the show, John realizes that he has forgotten to check the status of his order and is concerned that the display might not arrive in time. His Internet store provides order status online but he could not access the Web due to the lack of an Internet connection while on the road. John wishes that he could check the order status on his wireless phone the same way that he could for traffic/weather conditions and stock prices. In despair, he calls the store's customer support center, hoping to talk to someone even though it is already past the normal working hours. Fortunately, he finds a service representative who stays later than usual and is able to confirm the delivery. John is very happy with the news, however, he certainly appreciates an automated, self-help phone line that allows him to check order status at any time including weekends. While the previous scenario is hypothetical, it is not far fetched and illustrates a case for purchase order status in a mobile commerce environment.

In this paper, we first discuss the business motivation, then outline the overall architecture and the implementation of a wireless purchase order status solution that provides the users with a mobile browsing experience paralleling what is available over the wired Internet. As a result, users can look up

the status of their purchase orders via a wireless device. However, there are some limitations of mobile devices and networks. The small screens available on most cell phones and PDAs provide only a few lines of text, and the stateless connection and limited bandwidth of most wireless networks greatly constrains the amount of information that can be delivered to wireless devices. In this paper, we describe some of the techniques that are used to circumvent these constraints. We also describe our approach to extend this purchase order status solution to include a prototype for voice access by transcoding eXtensible Markup Language (XML) [1] application content into Voice Extensible Markup Language (VoiceXML), using a VoiceXML [2] browser, and utilizing the text-to-speech engine in a voice server. Finally, the potential use of this solution as a push service application is discussed.

2. MOTIVATION

Two applications of mobile commerce [3] that we have considered are ordering (shopping) and order status. Given the current display size, bandwidth and input limitations, shopping does not appear to be practical at this juncture, although continuing advances in wireless technology and improvements in products and services will make it more viable and available in the future. Order status, however, seems both practical and offers great value and utility to users now.

If there is an existing order status application, customers will already be familiar with it, and have the added utility of being able to access order status and shipping information, if there are links to shipping companies, from any location. All that is needed is a customer identifier such as a customer number and an order number. Of course, in both the wired and wireless environments these numbers must be entered via keyboard or keypad, and as described in a later section, we have prototyped voice enablement of these applications to make input easier. A natural phone interface would improve the usability of this and other applications. Wireless enabled order status and voice enabled order status fit a strategy of delivering value applications and of increasing their function and usability.

Market studies reveal that the number of wireless users as well as Internet users continues to grow at a rapid pace.

Global Internet and Wireless Users, 2001, 2004 and 2007			
Subscribers	**2001**	**2004**	**2007**
Internet users (millions)	533	945	1,460
Wireless Internet users as a % of all Internet Users	16	41.5	56.8

Source: eMarketer, March 2002

However, research and experience indicate that for many users the novelty of wireless Internet usage has worn off, and that some have either stopped using their devices for Internet access or discontinued service. In order to achieve the projected growth, it appears that business users are going to drive usage, and that applications like order status will provide the value they are looking for. There may be more glamorous applications on the way, but this type of application is an important step in addressing customers' needs and desires.

At these early stages, what would a business get out of this? The benefits are many. Obvious ones will come from providing better service to important customer segments. Many readers will be able to relate to the fact that significant cost savings may be realized when users get order status online vs. talking to someone in call centers. For those in the business it can be a demonstration of their commitment to m-commerce and the strength and breadth of their wireless commerce enabling products and services.

3. ARCHITECTURE AND DESIGN CHOICES

This section briefly describes a high-level architecture of the wireless order status application and discusses the advantages and disadvantages of possible design choices.

The key components of the wireless order status application are depicted on the upper half of Figure1:

- ➔ HyperText Transfer Protocol (HTTP) Web server handles HTTP requests and replies from wireless devices
- ➔ Transcoder converts the reply data into a format suitable for a given wireless device such as Wireless Markup Language (WML), Handheld Device Markup Language (HDML), HTML and compact HyperText Markup Language (cHTML).
- ➔ Application Server contains the business logic to process the order status requests.

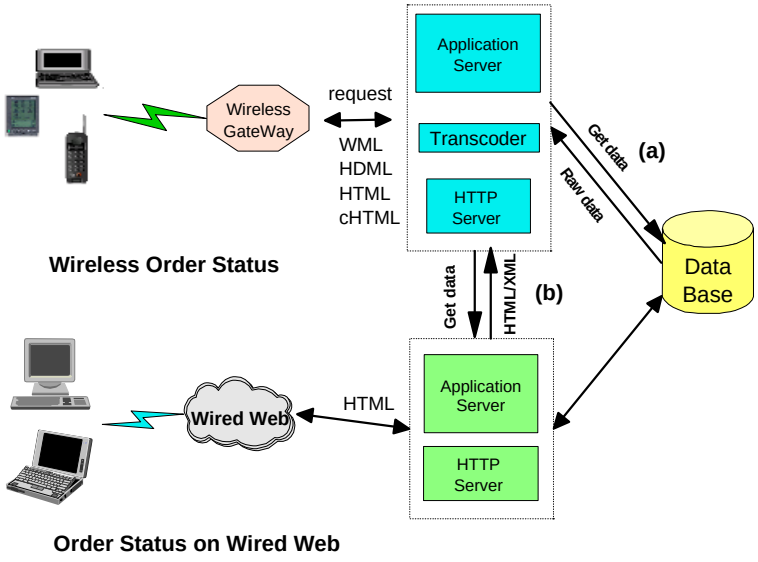

Figure 1: The architecture of order status for wireless and wired network

In a typical usage scenario, individuals utilize the micro-browser on their wireless devices to send a request to a designated URL to look up their purchase order status. The request passes through their wireless service provider's gateway and is forwarded to an HTTP server which then calls an

application server. The application server needs to retrieve the order status data to fulfill the request. It can pull the data directly from an order fulfillment database as shown by (a) in Figure 1, or it can get the data indirectly from another application server that provides order status on the wired web as depicted by (b) in the same figure. The order status result must be transformed into an appropriate data format so that it can be displayed on the wireless device. This transformation is performed by the transcoder.

There are two approaches to building a wireless order status solution which differ in the way the order status data is obtained:

a. Building a customized wireless order status application

The wireless order status application is developed from scratch and accesses the order fulfillment database directly for any required data. This approach offers more freedom and flexibility in the implementation which allows tighter integration with the data source and better optimization for performance. However, it can require more resources and time to build and deploy.

b. Extending an existing order status application

The alternative is to exploit the availability of an existing order status application on the wired web. It is likely that many Internet stores already provide a way for customers to check order status online via a web browser on a desktop or notebook computer, as shown in the lower half of Figure 1. In this case, it is feasible to extend the online order status capability to the wireless devices [4]. This approach requires the extraction of the order status data from the HTML or XML content served by the existing application. However, it can result in significant reduction of development time and expenses.

In this work, we have chosen to implement the wired order status solution by extending the existing order status application on the wired web.

4. WIRELESS ORDER STATUS

The wireless order status solution is implemented on the IBM Websphere application platform. In particular, the Websphere Application Server and Websphere Transcoding Publisher are used as the application server and Transcoder, respectively. Sections 4.1 through 4.6 describe the design and key components of the solution, selected results in the forms of screen shots and discuss some implementation issues and their resolutions.

4.1 Application Design

The key part of our wireless order status implementation is the Wireless Order Status Server shown in Figure 2. It handles incoming requests from wireless devices and interacts with the Order Status Server and the Shipping Status Server, as needed, on the wired web. The Wireless Order Status Server is comprised of the following components:

- Wireless Order Status Java Server Pages (JSPs)
- Wireless Order Status Java Servlet
- IBM Websphere Application Server (WAS)
- IBM WebSphere Transcoding Publisher (WTP) [5]

- IBM HTTP Server

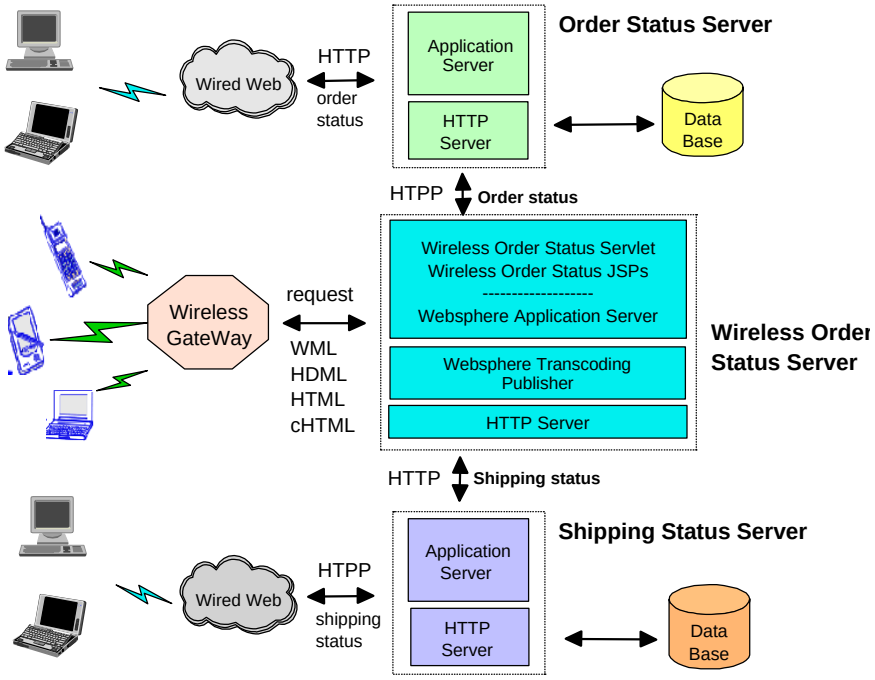

Figure 2: Wireless Order Status Implementation

The Websphere Application Server and the HTTP server provide the web infrastructure and application platform for the wireless order status program. The function of the other components are described below:

4.2 Wireless Order Status Java Server Pages

The Wireless Order Status (JSPs) contain the presentation logic for I/O data processing and user interactions. A user invokes an input JSP to initiate a request for an order status. This JSP prompts the user for a customer number and a purchase order number. When the user submits the request, an output JSP is invoked which, in turn, calls the Wireless Order Status Servlet and passes the two numbers as input parameters. The servlet returns an XML document to the output JSP for data transformation. The output JSP uses the Websphere Transcoding Publisher to transcode the data from an XML document into one of the following formats: Wireless Markup Language (WML) [7], Handheld Device Markup Language (HDML) [8], HTML, compact HyperText Markup Language (cHTML) [9] by invoking an appropriate stylesheet based on the WTP condition [5] set in the XML document and returns the result to the user.

4.3 Wireless Order Status Servlet

The Wireless Order Status Servlet (WOSS) contains the business logic necessary to process order status requests. It was designed to leverage the existing online order status application that provides order status in an HTML format through an HTTP request on the wired web. When the WOSS

is invoked by the input JSP, it makes a secured HTTP request to the online Order Status Server as if it is a user on the wired Internet. It then parses the HTML reply, extracts essential data and creates an XML document. If it finds URL links to Shipping Status Servers, it will make additional HTTP requests to those servers on the wired web, extracts the necessary shipping status data and adds it to the XML document.

Because of the small screen size and limited bandwidth of the wireless device, only selected data deemed as relevant and critical about the order and shipping status are extracted from the wired web servers and to be displayed to the user. A Document Type Definition (DTD) for wireless order status is created to structure the extracted data in an XML document for the output JSP as mentioned above.

4.4 Websphere Transcoding Publisher

Every time a request is made from a wireless device, the input JSP retrieves the user_agent information from the request's header to find out the characteristics of the originating microbrowser. This information is passed along with the XML order status results to the WebSphere Transcoding Publisher (WTP) [5] to select an appropriate stylesheet to transcode the results to a format and layout suitable for the wireless device.

In WTP, wireless devices can be classified into different groups based on their common user_agent information, such as the data format supported by their microbrowsers and the device rule conditions. Each group is characterized by a device property file, for example, the device property file named "wml_standard" specifies all the wireless devices with micro-browsers that support WML format, and the "chtml_standard" device property file specifies devices with browsers that support cHTML. In our wireless order status solution, we classified the wireless devices into six groups with six different device property files: Research In Motion (RIM)devices, Openwave Browser, WML, cHTML, HDML and HTML browsers. For each request, WTP maps its user_agent information to the device rule conditions specified in one of these six device property files, and the matched property file name will be used in the *WTP-condition* to decide which stylesheets to use for displaying the result of this request.

A WTP-condition is a WTP processing instruction that can be used to specify different stylesheets in the prologue of an XML document. Below is a example of the WTP-conditions appear in the prologue of a XML document showing how the four different stylesheets (XSLs) for WML, HDML, HTML, cHTML can be matched with a selected device property file name. For example, WTP maps a wireless device with user_agent specifying WML support to the device property file with the name "wml_standard". This name is then used to match the WTP-conditions to select a stylesheet. In this case the following WTP-condition is a match and the corresponding stylesheet will be used.:
<?wtp-condition stylesheet=<%=xsltRoot%>xslt/wml/stylesheetname.xsl" condition="device=wml_standard"?>

```
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/wml/stylesheetname.xsl" condition="device=wml_rim_neomar"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/hdml/stylesheetname.xsl"  condition="device=hdml_standard"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/wml/stylesheetname.xsl" condition="device=wml_listsupportyes"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/wml/stylesheetname.xsl"   condition="device=wml_standard"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/chtml/stylesheetname.xsl" condition="device=chtml_standard"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/html/stylesheetname.xsl"  condition="device=html_handheld"?>
<?wtp-condition stylesheet="<%=xsltRoot%>xslt/html/stylesheetname.xsl"  condition="device=*"?>
```

4.5 Wireless Order Status Screen Shots

Figures A through G illustrate a set of pages a user may view while checking the order status through a wireless device. These are screen shots taken from Pixo Internet Microbrowser 2.1 emulator [8]. Figure A shows the whole picture of the phone, but subsequent figures show only the display screen.

Fig. A

Fig. B

Fig. C

In Figures A & B, the user is asked to enter the order information: 7 digit customer and 6 digit quote/order number. After providing this information, the user submits the form.

In Figure C, the user receives a menu of options from the server and is asked to make a selection on the order status result to be displayed. The menu contains 4 choices: Order Status, Order Details, Billing Details and Shipment Details. The user can also select "Another Order" to check the status of another order.

Fig. D

Fig. F

Fig. E

Fig. G

Fig. D shows the general order status information when the “Order Status” option on the menu page is selected.

Fig. E displays the order details such as line items, taxes and shipping charges when the “Order Details” option on the menu page is chosen.

Fig. F depicts the shipping and billing addresses when the "Billing Details" option on the menu page is selected.

Fig G shows the shipping status when the "Shipment Details" option on the menu page is chosen..

These pictures are just a representation of what a user can view on a wireless device. Other pages such as error conditions are not included. For instance, if the order number and/or customer number are entered incorrectly, the user will see an error page with an option to retry.

4.6 Design Issues:

This section describes some of the design issues that we came across during our implementation of the wireless order status solution and the approaches we took to handle these issues.

- **Limited Buffer size of the wireless devices**:

Most of the wireless devices have a limited buffer size typically 1.5K to 2.5K bytes for wireless phones. When the requested page exceeds the buffer size, it creates a compilation problem. Since the order status page can easily exceed the limited buffer size, we broke it into shorter pages which allows the user to view one page at a time by clicking a link or a button.

After building the XML data, we added an index to the XML. This indicates the part of the data to be shown to the user at a certain time and stores the XML data in the server memory. When the stylesheet is invoked by WTP, the index is extracted and the final page is built depending on the value of the index along with links to the other available pages.

When a user clicks on a link, JSP is invoked to retrieve the data from memory and updates the index value before passing the data to the stylesheet. As different user's requests increase over time, the server memory will be filled and, hence, needs to be cleared periodically.

- **Cookie**:

The WOSS makes an HTTP or HTTPS connection to an application server on the wired web to retrieve order status and/or shipping status. Sometimes the application server needs to set a session cookie on the requester's machine before serving the actual data. This need can create a problem because many wireless devices do not support cookies. The WOSS circumvents this problem by making a connection to the server, retrieving the cookie from the header, setting the cookie in the next request's header and then making another request to the server for the actual result page.

5. VOICE ENABLEMENT

The wireless order status application requires an Internet microbrowser which may not be available in certain wireless devices. To broaden the application accessibility, we have investigated adding a voice interface to the front-end of the application. This voice interface allows users to call and listen to their orders status over the phone, thus enabling not only any wireless but also wired conventional phones. This section describes the voice-enabled order status prototype that we have extended from the wireless order status solution.

5.1 Design

Figure 3 illustrates the addition of a voice interface to the wireless order status solution by incorporating the IBM WebSphere Voice Response Server [10] as a front-end voice processor. This Voice Response server can accept voice (using voice recognition) as well as phone pad inputs from users. It also converts output data in text or VoiceXML format back to voice for delivery to the users' phone. The incremental development required to support voice is to provide one more stylesheet for WTP to convert XML order status data into VoiceXML. A device condition also needs to be added to the order status XML document for voice support.

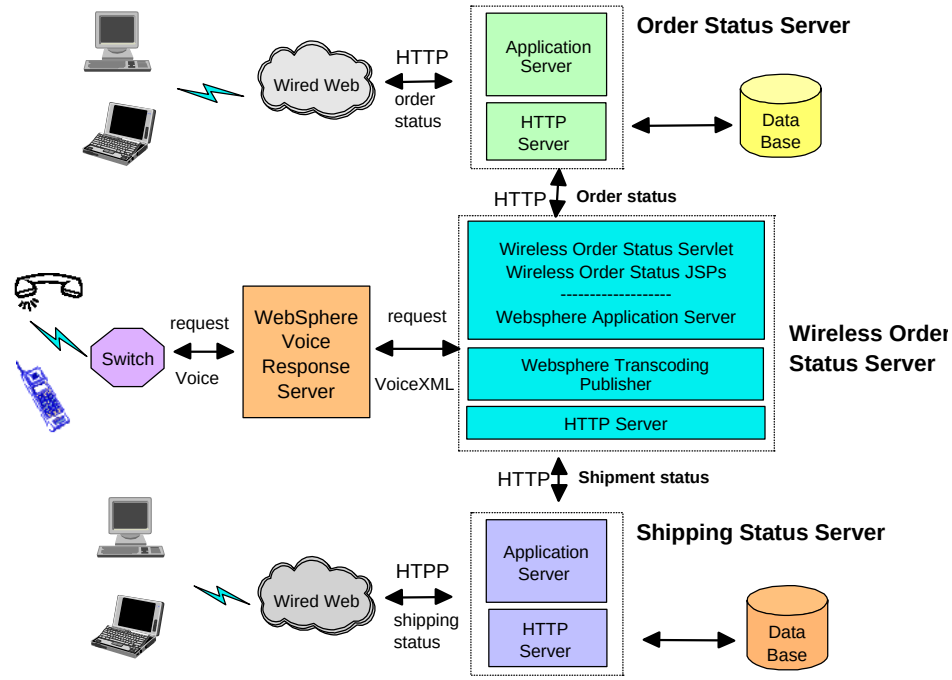

Figure 3: Voice Enabled (Wireless)Order Status Prototype

5.2 Prototype Implementation

To demonstrate the feasibility of extending voice support to the wireless order status application, we have built a prototype by integrating the wireless order status solution with the IBM WebSphere Voice Server Software Development Kit (SDK) [11] that simulates a telephony supported environment. A user requests order status by keying on a simulated phone pad or speaking to a microphone the customer number and order number. The SDK then invokes our wireless order status solution to retrieve the order status XML document. The WTP transcodes the XML data into VoiceXML by using the appropriate stylesheet. Finally, the SDK converts VoiceXML to voice using a Text-to-Speech engine and plays it through the computer speakers. The user is presented with a menu of options and uses the phone pad to select a section of the order status results to listen to. The user can switch to another section of the results at any time during the voice playback.

6. DISCUSSIONS

A major benefit of wireless is its ability to immediately deliver and/or push information to users whenever it becomes available rather than when it is requested. One of applications that can certainly take advantage of this benefit is the push service [12] to users for their purchase order status whenever there is a change or update if they sign up for it.

Since our wireless purchase order status leverages the existing purchase order status application, we know exactly where to access the purchase order status data on the wired web. If the data format and/or the page layout changes on the wired web then the current HTML parser that extracts the data for the wireless application may experience problems. A solution for this problem is to develop a more flexible parser or a web crawler to retrieve the required data independent of the above mentioned changes.

As wireless traffic can be intercepted by anyone with a scanning device, most mobile e-commerce transaction applications require the wireless web [13] to be secure in terms of privacy, authentication, data integrity, and nonrepudiation. However, WAP versions 1.2.1 [14] or earlier versions uses the Wireless Transport Layer Security (WLTS) protocol and a wireless version of Public Key Infrastructure (PKI) to secure wireless transactions. Secure communication from a cellular phone to a WAP gateway travels with WTLS until it gets to the gateway where it is converted to a Secure Sockets Layer (SSL) as it continues its path to a web server. Before a WAP gateway can convert a WTLS stream to an encrypted SSL stream, it must first decrypt the WTLS packets. There is a potential security problems at this junction. As a result, our purchase order status solution does not involve any transactions or reveal any credit card information.

7. CONCLUSIONS

We implemented a wireless purchase order status solution to enable users to check order status on a wireless device equipped with a web browser. This solution utilizes the services and tools provided by the Wesphere Transcoding Publisher and the Websphere Application Server. It also leverages an existing order status application on the wired web, resulting in reduced development time and expenses. We have addressed some of the implementation issues related to limitations and differences in features and capabilities of wireless devices. We have also extended this purchase order status solution in a prototype to support voice access. As a result, the order status can be accessed via cellular phones without web browsers as well as conventional phones. Order status appears to be a promising mobile commerce application because it is practical for today's wireless technology and offers value and utility to users. It also has a potential to reduce costs for businesses in customer support and calls centers.

8. ACKNOWLEDGMENTS

The authors would like to acknowledge the review of the manuscripts, suggestions for improvement and support from Nagui Halim, Daniel Dias, David Leip and Colby Mims.

9. REFERENCES

1. Goldfarb, Charles F., and Prescod, Paul., ***XML Handbook***, Prentice Hall, New York, 2001.
2. Lamb M., and Horowitz, B., "Guidelines for a VoiceXML Solution Using WebSphere Transcoding Publisher". Available at ftp://ftp.software.ibm.com/software/wtp/info/VxmlTranscodingGuide.pdf.
3. May, Paul., ***Mobile Commerce***, Cambridge Univ. Press, London, 2001.
4. IBM Redbook, IBM WebSphere Transcoding Publisher Version 1.1: Extending Web Applications to the Pervasive World, SG24-5965-00
5. IBM, "IBM WebSphere Transcoding Publisher Version 4.0 Developer's Guide". Available at ftp://ftp.software.ibm.com/software/webserver/transcoding/brochures/tpdgmst.pdf.
6. XSL Transformations (XSLT) Version 1.0. Available at http://www.w3.org/TR/xslt
7. The WAP standards. Available at http://www.wapforum.org
8. Pixo Internet Microbrowser 2.1 Web Site. Available at http://www.pixo.com/
9. Frengle, Nik., ***I-Mode: A Primer***, John Wiley & Sons, New York, 2001.
10. IBM, WebSphere Voice Response for Windows NT and Windows 2000, Version 3.1. Available at http://www-3.ibm.com/software/enterprise/ep_lit.html#win31
11. IBM WebSphere Voice Server SDK 2.0. Available at http://www-3.ibm.com/software/enterprise/ep_11.html
12. Lubar, Dan., ***Wireless Messaging Demystified: SMS, EMS, MMS, and Others***, McGraw-Hill Professional, New York, 2002.
13. Alesso, H. Peter., and Smith, Craig F., ***The Intelligent Wireless Web,*** Addison-Wesley, New York, 2002.
14. Dornan, Andy., ***The Essential Guide to Wireless Communications Applications, From Cellular Systems to WAP and M-Commerce***, Prentice Hall, New York, 2000.