

Department of Computer Science and Engineering

The Chinese University of Hong Kong

LYU0001

Wireless based Mobile E-Commerce on the web

Final Year Project Report, Fall 2000

Supervisor: Professor Michael R. Lyu

Prepared by: Wat Hong Fai (Student ID 98603881)

Yan Wai Keung (Student ID 98730434)

Date: 1st December 2000

Contents

1. Abstract	5
2. Background	6
2.1 E-commerce	6
2.2 Trend on Wireless Internet	7
3. WAP	8
3.1 Definition	8
3.2 History	8
3.3 Background Statistics	8
3.4 Usage and potential of WAP in different area	9
3.5 WAP Architecture	10
3.5.1 HTTP Interface	12
3.5.2 WSP	13
3.5.3 WTP	13
3.5.4 WTLS	13
3.5.5 WDP	13
3.5.6 Bearer	13
3.6 Key to build an application	16
3.7 WML Basic	18
3.8 Advantages	21
3.9 Limitations	22
3.10 The future	24
4. Project Considerations and Approach	25
4. Project Considerations and Approach	26
4.1 Motivation	26

4.2 System Features	26
4.2.1 Introduction.....	26
4.2.2 User Registration	26
4.2.3 Login.....	27
4.2.4 User Profile Management	28
4.2.5 Posting Product Information.....	29
4.2.6 Browsing Products in Categories.....	30
4.2.7 Giving Response	31
4.2.8 View Status	32
5. Programming Environment.....	33
5.1 Hardware Platform.....	33
5.2 Server Configuration.....	33
5.3 Programming Language.....	34
5.4 Database System	34
5.5 Program Testing	35
6. System Implementation	36
6.1 Overview.....	36
6.2 Site Map.....	37
6.3 Database.....	38
7. Problems Encountered and Lesson Learned.....	41
7.1 Problems Encountered	41
7.1.1 Department Firewall	41
7.1.2 Perl with Oracle	41
7.2. Things Learned	41
7.2.1 E-commerce Models	41

7.2.2 Wireless Internet Access	42
8. Conclusion and Future Plans	43
8.1 Conclusion	43
8.2 Planning for XML and XSL.....	43
9. References.....	44
Appendix. Overview of XML and XSL.....	49
A.1 Definition of XML	49
A.2 Component of XML.....	49
A.3 Advantage of using XML.....	51
A.4 XML Development Goals	51
A.5 Apply to our project	53
A.6 XSLT	54
A.7 Transformation from XML to non-XML using XSL	56
A.8 Example of using XSLT:.....	57

1. Abstract

In this report, we will mention briefly the e-commerce trend and the behavior of usage of wireless Internet nowadays. Then we will present a Wireless Application Protocol (WAP) section for its background, architecture, advantage and limitations. We will also briefly talk about its future. A Wireless Markup Language (WML) section will also be included.

Then, we will introduce our project and describe the system features. In a Programming Environment section, we describe the system and programming language used in this project. A System Implementation section includes more detailed implementation of our system such as the database schema we used.

In the first semester, we have encountered some problems. Consequently, we will include some of them in the Problem Encountered section. Apart from problems, we have learned a lot of things, which will be listed in a section about our learning.

Lastly, we will have a section on our future plans, describing plans we are going to do in the coming semester, including XML and XSL.

2. Background

2.1 E-commerce

In the past, Internet is only used by people to search for information, and later evolved to be a tool for communication. As the Internet is gaining its popularity, merchants begin to use the Internet as a tool for advertising and doing transactions. This is known as electronic commerce, E-commerce.

According to a study “2000 America Online/Roper Starch Cyberstudy”, online commerce has become an everyday activity for millions of users. This year, more than half of the survey's respondents (56 percent) say that they currently shop online, nearly double the percent of those who did in 1998 (31 percent). In addition to shopping more frequently, consumers are also spending more when they shop online.

Internet Use, 1998 vs. 2000		
Use	1998	2000
Shop online	31%	56%
Intend to shop online in future	14%	28%
Plan to increase online purchases	41%	49%
Banking	16%	25%
Trading stocks	11%	16%
Source: AOL/Roper Starch Cyberstudy		

As users grow accustomed to integrating the online medium in their everyday lives, they are also increasingly demanding access to it from any place, at any time. "That increased consumer reliance on popular online services can be seen in strong and growing demand for Internet-accessible cell phones, PDAs, two-way pagers, small kitchen devices, and TV set top boxes." said Marshall Cohen, America Online Senior Vice President of Brand Development.

2.2 Trend on Wireless Internet

According to a research of the E-Business Unit (EBU) of Cap Gemini America (<http://www.usa.capgemini.com/>) and Corechange Inc (<http://www.corechange.com/>), seventy percent of the present users use wireless data applications on their cell phones at least once a day and 60 percent use it for at least 1 to 3 minutes per occasion. In the future, 88 percent will use it at least once a day and 19 percent will use it more than three times a day. Sixty-six percent will use it more than three minutes every time. This indicates an enormous growth in wireless traffic data.

As predicted by International Data Corp (<http://www.idc.com/>), by mid-2001, all digital cellular/PCS handsets shipped in the world will be WAP-capable, and by the end of 2002, there will be more wireless subscribers capable of Internet access than wired Internet users. When this happens, there will be a fundamental shift in the thinking of the Web community and the IT industry as a whole. "Once there are more wireless Internet users than wired users, Web masters may first consider the needs of the wireless users and secondly, the wired PC users," said Iain Gillot, vice president, Worldwide Consumer and Small Business Telecommunications research at IDC. "Today, Internet sites have to be retro-fitted for wireless users, but three years from now, it is conceivable they might have to be redesigned for wired users."

Recently, mobile phones can get connected to the Internet, but this not the same way as we do from our wired computer. Since there is a different computational environment with much more limitations on wireless devices, a new protocol, WAP, is being developed that best suit the mobile situation.

3. WAP

3.1 Definition

The Wireless Application Protocol (WAP) is an open, global specification, which is optimized for mobile users with wireless devices the opportunity to easily access, interact with information and services, and more importantly, surfing the internet. The protocol is developed by WAP Forum (<http://www.wapforum.org/>), an organization of some of the most powerful Internet and telecom companies.

3.2 History

Phone.com, Ericsson, Nokia and many others began developing standards independently of each other, but soon they realized that it would make more sense to focus development around a common standard. They all wanted to establish a common format for Internet transfers to mobile telephones, without having to customize the Internet pages for the particular display on every different mobile telephone or personal organizer. With WAP Forum they were able to find such a protocol.

3.3 Background Statistics

Forrester is one of the leading independent research firms which analyses the future of technological change and its impact on businesses, consumers, and society. This research institute has conducted a research on WAP, and presumes that by 2004, one-third of all Europeans -- more than 219 million consumers -- will regularly use their mobile phones to access Internet services. According to a recent report from Forrester, ninety percent of the e-commerce executives interviewed plan to launch Internet sites designed for mobile phones with tiny displays and thin connections.

Forrester-analysts stress the fact that the only thing which can slow down this development is restricted user access.

In other words, smart cell phones will become the dominant means of accessing the Internet in some years, when Web-capable phones will make up 79 percent of the unit volume of Internet appliances, according to a report by Semico Research.

3.4 Usage and potential of WAP in different area

Using mobile phone to surf the net is better than what we can have in desktop computer in some way.

If we have a mobile phone we will probably have noticed that our phone is a more “private” device than our computer. Our phone is always within reach, we can get hold of all our friends using it and we have a lot of personal information in store. People that have had to leave their homes in a hurry, often tell the press that the items they grabbed on their way out were the wallet and the mobile phone. The latter has almost become a part of their personalities, and people like to express their identities through the design and the price of their phones. Leading phone producers will tell us that they use great amounts of time and money to find out what segment of mobile users prefer what kind of phones when it comes to size, colour and functionality.

As opposed to our PC, the WAP phone will always be within range. It invites us to order something wherever we are, and we can search for information in the middle of a discussion around the café table. We can check our e-mail on the run, and we can be in touch with the Intranet in our office anytime.

In a few months, everybody will buy a WAP phone when their old phone collapse. Nowadays, it is the coolest guys and the business people who buy them.

Young people will buy a WAP phone to keep in touch with their friends. We can pop our mail by our phone, and we can chat with strangers if we like that. Teenagers will love keeping themselves updated on the fashion and culture arena by WAP, and they can buy movie tickets and other items by phone.

Business people will buy a WAP in order to buy stocks on their way to the airport. They will check the destination weather before they buy some new clothes for the trip, and they will be logged on to the company WAP pages whenever and wherever they want to.

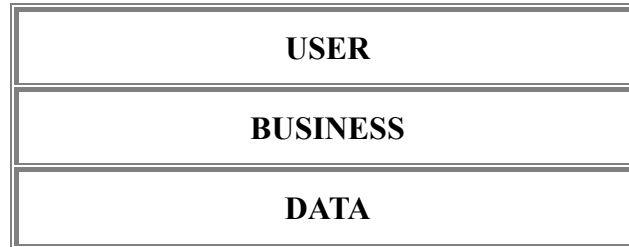
Our neighbour will buy a WAP phone because we already have one, and because he thinks it is practical to have access to the latest Wall Street news.

We will buy a WAP phone because we never remember to buy a TV guide, and we need to know if our favourite TV program is on TVB today. Besides, it is nice to be able to read about we favourite band when we are stuck on the train and everybody is starrng at the chewing gums on the floor.

3.5 WAP Architecture

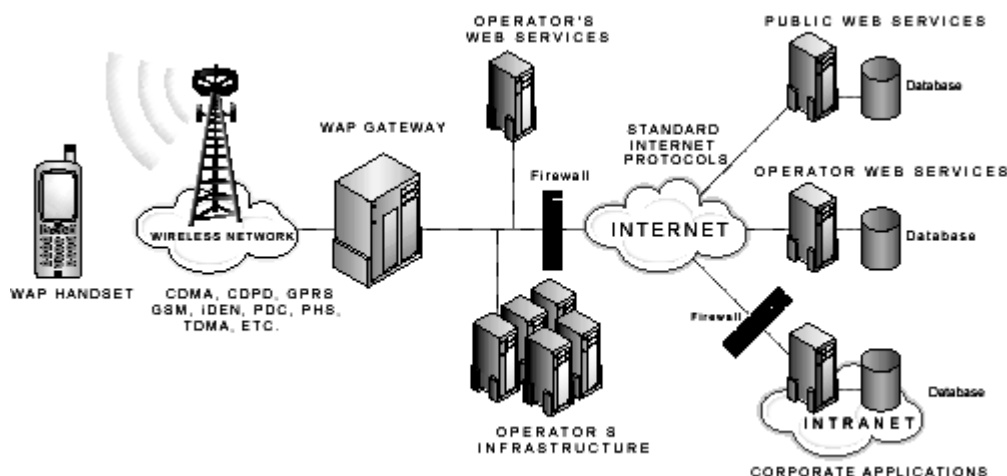
The WAP protocols are analogous to many of those used in existing Internet technology, yet are optimized for the challenges raised by small, narrowband client devices (such as mobile phones) and limited over-the-air bandwidth.

As WAP is an extension of the Internet onto wireless devices, it makes sense that it should share the same architecture. In its most simplified form, this is referred to as the three-layer model, as shown in the diagram below.



Internet applications work as follows:

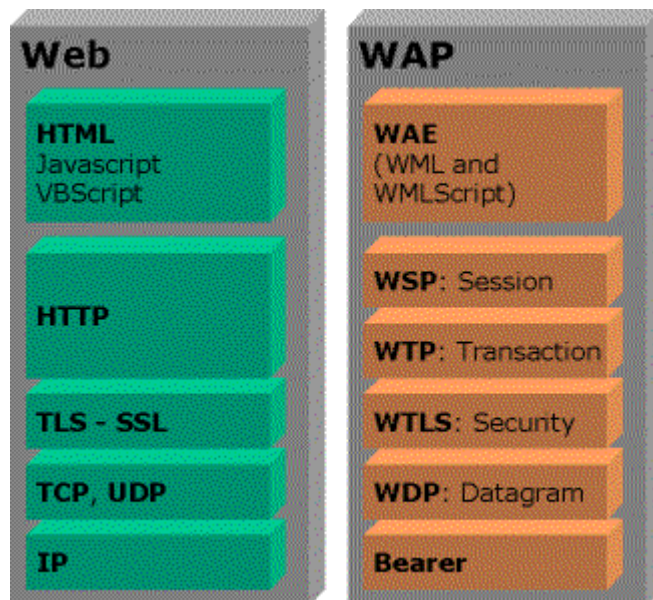
The User layer exists to present information to end-users; the business layer is where the bulk of application execution takes place. Finally the Data layer consists of databases, legacy systems, external information feeds and other resources that the application might need to access. In general, Internet applications extend this model to support the thin-client model of the Web browser - an additional layer, known as the presentation layer, is added between the user layer and the business layer. In physical terms, the presentation layer maps onto the Web server just as the user layer maps onto the Web browser. The common language of the Internet, TCP/IP, separates the user layer and the presentation layer.



It is this boundary, between user and presentation layers, that is the most important in the WAP-based wireless application architecture.

An essential principle of WAP is that it adds wireless proxy technology to the model above - rather than having a web server communicating directly with a browser on a wireless device, additional facilities are added to overcome the difficulties caused by today's wireless communications. These facilities include a protocol stack designed to minimize bandwidth requirements and to preserve the security and integrity of transactions, together with client and server extensions to execute the WAP-specific elements of the application.

The WAP Protocols cover both the application level (WML and WMLScript; collectively known as WAE, the Wireless Application Environment), and the underlying transport layers (WSP and WTP, WTLS, and WDP). The fact that the WAP Forum defines all these levels



of the communications stack means that everything should hang together quite nicely. (In the web world, the application layers and the transport layers are specified by different organizations, but actually it still hangs together quite nicely).

3.5.1 HTTP Interface

The HTTP interface serves to retrieve WAP content from the Internet requested by the mobile device.

WAP content (WML and WMLScript) is converted into a compact binary form for

transmission over the air

3.5.2 WSP

The WAP session protocol (WSP) layer provides a lightweight session layer to allow efficient exchange of data between applications.

3.5.3 WTP

The WAP transaction protocol (WTP) layer provides transaction support, adding reliability to the datagram service provided by WDP.

3.5.4 WTLS

Wireless transport layer security (WTLS), an optional security layer, has encryption facilities that provide the secure transport service required by many applications, such as e-commerce.

3.5.5 WDP

The WAP datagram protocol (WDP) is the transport layer that sends and receives messages via any available bearer network, including SMS, USSD, CSD, CDPD, IS-136 packet data, and GPRS.

3.5.6 Bearer

The 'bearer' level of WAP depends on the type of mobile network. It could be CSD, SMS, CDMA, or any of a large number of possible data carriers. Whichever our target client is using, the development above remains the same.

Note that the WAP application architecture is not about the languages or data structures that are used. In theory, WAP could support the transmission of any data format. In practice, formats are currently limited to the Wireless Markup Language,

WML and the scripting language, WMLscript. In physical terms, each of the layers in our model is affected by WAP, as follows:

User Layer. In the WAP architecture, the client device is currently a mobile phone or a PDA running a WAP-compliant micro-browser. This part of the WAP specification is known as the Wireless Application Environment, or WAE.

Presentation Layer. A WAP proxy is added to the Web server in order to manage and serve WML pages. In addition, a filter can be included between the Web server and the WAP proxy, which converts HTML pages to WML on the fly.

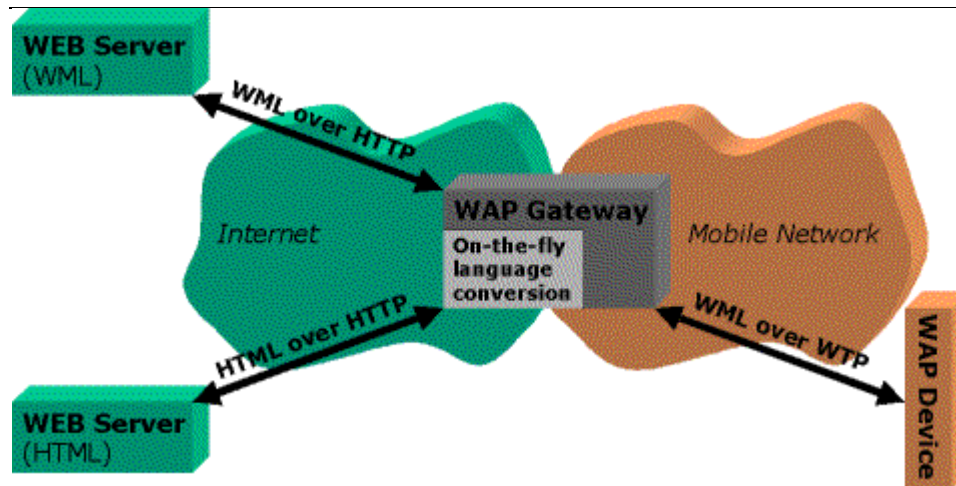
Business layer. Application servers are available that have been designed or extended specifically to take into account the needs of wireless applications. These are covered in more detail in a coming article.

The Internet comprises clients (browsers) and servers which actually host the pages that we view - or the applications that produce the pages on-the-fly.

WAP is no different. Our clients are now the mobile devices, and the servers are still hosting static pages and applications.

The only consideration we have to now make is that we will have to connect the mobile network to the Internet - so that the phone can make a request for the page and it will be able to reach the server. (And of course the response has to make it back).

To connect these two mega-networks, the WAP Specification assumes there will be a WAP Gateway. At its simplest level, this is a 'stack converter'. It converts the 'WAP' request into a 'Web' request and the 'Web' response into a 'WAP' response.



In theory, the gateway could also be made to convert the HTML page content itself on-the-fly as well. For example, if a phone makes a request to a HTML page, the gateway might feel it is capable of converting the HTML response into WML - so that the phone can understand it.

However, we need to bear in mind that WML is designed for small screens, and is therefore not as visually rich as HTML. Any such conversion will compromise a great deal of the HTML content and formatting - as well as any in-line scripting.

Some of the WAP Gateway products that are now coming on to the market (such as Nokia's WAP Server) also provide hosting capabilities themselves. There are no advantages to hosting static content on a WAP Gateway (other than perhaps performance), but if we are running dynamic web applications on a WAP Gateway, we may be able to determine more about our target client's request. In theory, and in the future, we might be able to integrate our WAP Server into the mobile network to gain information about the subscriber's location, for example.

Hosting on our own WAP Gateway-type server will require us to maintain

connections into the mobile network somehow. For example, we might dial into the network's SMS engine, or provide dial-in modems for CSD access (that is, the mobile will dial our server as though it was a regular point of presence, like an ISP).

Although the impact of WAP upon the Internet application architecture is relatively low, there are nonetheless a number of factors that affect the way that applications are constructed. Perhaps most clearly, the user interface will change - it must be designed to be usable on small-screen, limited-input devices. Also, it needs to be written in WML and WMLscript rather than HTML and JavaScript or DHTML.

An additional element of the WAP architecture differs from the Internet model, as its origins lie in telecommunications rather than IT. This is the WTA server (Wireless Telephony API). WTA enables WMLscript applets to access telephony functionality such as call control, phone book access and messaging. Hence This could be used in a number of ways. Perhaps the simplest application is that of being able to dial a number directly from a WMLscript-based address book application. More advanced would be the capability to forward a call by selecting options from a menu. Finally, fully-fledged WTA is envisaged to be used in "click-to-talk" applications. For example, if a user is booking a flight ticket and needs to consult with an agent, the user could click a button to dial directly into the customer service department responsible. This could spell the end of voice mail prompts, to nobody's chagrin.

3.6 Key to build an application

Additional impacts will be felt in the business layer. The transactions to be invoked or the information to be captured may stay the same, however the way in which these things happen may well change. Consider, for example, purchasing a flight ticket

using a Web browser compared to using a mobile phone with limited input capabilities. In the former, a more complex sequence of user interactions would be possible such as search for flights, construction of an itinerary, selection of dietary requirements and so on. In the latter (mobile phone) case, such a sequence would be too complex to be usable; hence the process might well be restricted to booking return fares across a limited selection of routes.

The key to building applications that work both on Web browsers and on mobile devices is to build the applications using component-based development techniques. This boils down to defining application building blocks (components) that clearly bound the different elements of the application. Such applications are likely to be written using the Object Management Group's Common Object Request Broker Architecture (CORBA), or Microsoft's Common Object Model (COM) architecture. These approaches are nothing new for applications that have been written explicitly to support Web functionality such as E-Commerce.

Use of components will only help if the right sorts of components are used. They need to be of the right granularity, too big and they will be unwieldy, too small and the resulting application will be too complex. What is more, a success is dependent on the existence of a certain sort of component, known as a Controller. Correctly structured applications will use Controller components to separate device-specific functionality from generic functionality, hence migrating an E-Commerce application to a WAP environment becomes a task of writing new Controller components supporting the mobile devices.

3.7 WML Basic

Let's go right back to basics for one moment. On the web, we have the very ubiquitous HTML (HyperText Markup Language) for describing the look and text of pages that are seen by web browsers. These pages also tell the browser about the locations of the graphics to put on the page, and the links to be made to other pages.

It's quite simple in WAP: WML is the equivalent of



HTML - and WBMP graphics are the monochromatic { GIF format WBMP format
be able to view.

HTML these days is used to encapsulate both the exact form *and* the content of a page. Modern web pages are pixel-perfect and often works of graphic art. Combined with knowledge of the quirks of Internet Explorer or Netscape Navigator, an experienced web author can dictate exactly what the page will look like.

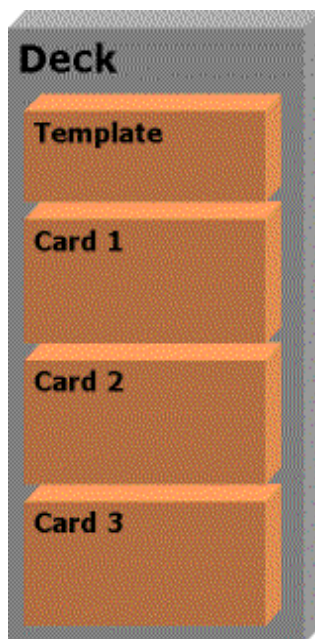
Unfortunately, WML doesn't quite offer that power of design; and if we're expecting pixel-perfect WAP pages, then we're in for a big surprise.

WAP devices and their characteristics are very varied - from tiny phone screens to a respectable PDA display. With this in mind, WML (which is the language used to write WAP pages) makes very few assumptions about how the page will be rendered on the final device.

In other words, we can be fairly sure that although the WML page will contain the content we want to display, we should leave any dreams of dictating form at the door!

Once the market is full of WAP phones, testing pages across all potential devices will be impossible - it's therefore worth getting used to this new philosophy now.

Many web designers will find that writing in WML is actually fairly straightforward. It's been designed with knowledge of how HTML has evolved, and as a result is very pragmatic.



There are well-defined rules for WML structure, which we will find are well documented by the WML Standards (at <http://www.wapforum.com>) - and we can be fairly sure that if our content meets the rules, it should be rendered sensibly by all compatible phone devices.

However, we can realize that these rules are *very* strict compared to those of HTML. (Or, to be more precise, the HTML standards are treated quite loosely these days). For example, we can't get away with nesting tags incorrectly, and the encoding of special characters like the ampersand (&) is very important.

A WML document is composed of a deck, and a deck contains multiple cards. In practice, we can think of a deck as an HTML document comprising many individually addressed parts. The #label syntax is used to refer to a card in a deck.

The mobile device downloads a complete deck, but only displays one card at the time. User can navigate through the links inside the document by tasks associated with the

soft keys.

Here is a simple WML document:

File name:- index.wml

Location:- <http://www.cse.cuhk.edu.hk/~lyu0001/index.wml>

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML
1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml><!-- This is a comment. -->
<template><!-- Provide characteristics which are
common to cards across the whole of a deck. -->
<do type='prev' label='Back'><prev/>
</do>
</template>
<card title="FYP 00-01" id="main">
<p>CUHK CSE Department 2000-2001 Final Year
Project<br/>
<a href='#subject'>Continue..</a>
<!-- When the link is selected, next card in the
same deck with id="subject" will be displayed. -->
</p>
</card>
<card title="FYP LYU0001" id="subject">
<p>LYU0001 Wireless-based Mobile E-Commerce on the Web<br/>
<a href='description.wml'>Description</a><br/>
<!-- Go to another deck with file named "description.wml" in relative path.
-->
<a href='http://www.cse.cuhk.edu.hk/~lyu0001/people.wml'>People</a>
<!-- The link can also be specified by absolute path. -->
</p>
</card>
</wml>
```



**Note: WML is much stricter than HTML, every tags must have its end tag. All the tags and attributes should be in lower case. Otherwise, it will result in errors!!*

Since WML is simply a specific type of eXtensible Markup Language (XML) language. The second line describes the type of WML that we are using, and where the client device could go to look up the instructions for how to deal with this file.

A deck has to be a valid XML document, and this implies that a WML document should start with the standard XML header and the reference to the Document Type Definition (DTD).

Once our mobile device receives a deck, it will display the text of the first card. If we press the link, we will be able to navigate to the next card or another deck.

Here are the functional areas that recent WML support:

1. Text presentation and layout
2. Deck/card organizational metaphor,
3. Inter-card navigation and linking
4. String parameterization and state management (WML variables)
5. Interface to phone features

3.8 Advantages

We cannot get access to the WAP services without a WAP phone. They are not too expensive, and the telecom experts predict that WAP phones will be better, faster and cheaper in the next few months. This is a technology that develops fast, and the newest WAP phones are much better compared to yesterday's phones which we can buy in our local telecom shop for a nice price.

Internet is still the best when it comes to exchange of information. But if our only interest is to be kept updated on the stock market or the latest sport reviews, WAP is

the fastest and easiest way to be online. The e-commerce business' WAP is even better than shopping on the Internet. We don't need a big screen to order the last Smashing Pumpkins CD. All we need to make the order is a WAP-phone, click a few times, wherever we are - whenever we choose.

3.9 Limitations

Since the client device used to surf the internet through WAP is a little bit different to the original web browser, WAP is designed with following constraints:

1. Small display
2. Limited input facilities
3. Narrow bandwidth network connection
4. Limited memory
5. Insufficient computational resources

1. Small Display

As long as the mobile phones are tiny, practical and easy to carry in our pocket, the WAP screen has to be small. We can't read through the same amount of information as on our computer. And, the screen quality is not as good. We can send pictures with WAP, but if they are too complicated they will merely appear as blurs.

2. Limited Input Facilities

It is much harder to write an e-mail on our phone than on our PC. When people start to use their mobile phone to read and write e-mails, the need for a better keyboard will force the producers to develop a better system

3. Narrow bandwidth of network connection

Telecom-developers are constantly working on a better and faster way of sending information to our mobile phone. The GSM network implemented in most of Europe today, is a slow way of sending information, proximately 9.6 Kbps. New network technologies are called HSCD (high-speed circuit switch data) and GPRS (General Packet Radio Service). They will give we a faster connection. In a few years most of the European countries will build a new broadband network, called UMTS.

4. Limited Memory

The WAP phones we get today have little memory capacity. The total size of a standard WAP page should not exceed 1.4 Kbytes. The development of new WAP services will probably press the hardware producers to come up with better memory systems for mobile terminals in the near future.

5. Insufficient Computational Resources

Since the recent cellular phones are made for mobile telephony service in the first place, the size of such devices is to be made as tiny as possible. Moreover, limited battery supply is also a great concern in design a phone. Thus, the WAP phones now available do not have much computational power to process complex data sent through the network.

3.10 The future

The tremendous surge of interest and development in the area of wireless data in recent times has caused worldwide operators, infrastructure and terminal manufacturers, and content developers to collaborate on an unprecedented scale, in an area notorious for the diversity of standards and protocols. The collaborative



efforts of the WAP Forum have devised and continue to develop a set of protocols that provide a common environment for the development of advanced telephony services and Internet access for the wireless market. If the WAP protocols were to be as

successful as transmission control protocol (TCP)/Internet protocol (IP), the boom in mobile communications would be phenomenal. Indeed, the WAP browser should do for mobile Internet what Netscape did for the Internet.



As mentioned earlier, industry players from content developers to operators can explore the vast opportunity that WAP presents. As a fixed-line technology, the Internet

has proved highly successful in reaching the homes of millions worldwide. However, mobile users until now have been forced to accept relatively basic levels of functionality, over and above voice communications and are beginning to demand the industry to move from a fixed to a mobile environment, carrying the functionality of a fixed environment with it.

Initially, services are expected to run over the well-established SMS bearer, which will dictate the nature and speed of early applications. Indeed, GSM currently does

not offer the data rates that would allow mobile multimedia and Web browsing. With the advent of GPRS, which aimed at increasing the data rate to 115 kbps, as well as other emerging high-bandwidth



bearers, the reality of access speeds equivalent or higher to that of a fixed-line scenario become evermore believable. GPRS is seen by many as the perfect partner for WAP, with its distinct time slots serving to manage data packets in a way that prevents users from being penalized for holding standard circuit-switched connections.

4. Project Considerations and Approach

4.1 Motivation

As prediction on the Internet shows that there will be more wireless subscribers capable of Internet access than wired Internet users in a few years, it is inevitable that the way people doing e-commerce will change from the current wired network to wireless means. Our project is to explore the world of e-commerce in it's beginning age of moving to the wireless world.

4.2 System Features

4.2.1 Introduction

One of the advantages of using wireless Internet access over the traditional wired access is the ability to let user access the get online anytime, anywhere, which allow user to get more update and more interaction from Internet sites. In order to take advantage of this feature through e-commerce, we have chosen to implement a second hand market system that allows users to access through traditional web, and through WAP enabled devices.

4.2.2 User Registration

Users are required to register in our system to get an account before posting products for sell and request, although pure browsing through categories do not require a user account. There is a page for users to register to open an account in our system, users are required to enter their names, the preferred username and password and telephone number. Bank account number is also needed if the user wants to sell products through our system so that money credited through the selling of the products are transferred to the user's bank account. Users are able to fill in credit card information,

such as card number and expiry date, so that they can use the card information to make purchasing in the future without having to type in every time, especially when user is using a WAP phone of which is time consuming to input a long string.

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled 'Internet Resale Store TO SECOR'. The page has a dark blue background with a yellow navigation menu on the left containing links: Index, Category for Sale, Category for Buy, Post Products, Registration, and Contact Us. The main content area is a yellow box titled 'Please Enter Your Information:' containing a registration form with the following fields:

- Login ID:
- Password:
- Password (again):
- Name in Full:
- Contact number:
- Address:
- Email:
- Bank Account:
- Payment method: (By Credit card, Optional)
 - Visa:
 - Name:
 - Expiry Date (MM/YY):
 - Master:
 - Name:
 - Expiry Date (MM/YY):

At the bottom of the form are 'Submit' and 'Reset' buttons. The browser's status bar at the bottom shows '完成' (Done) and '近端 intranet' (Local intranet).

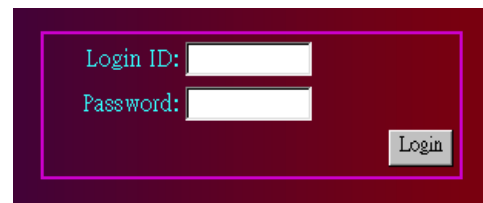
4.2.3 Login



After the users have got account in our system, they can login to use our system.

When they successfully login, a pseudo-random session id is generated to represent

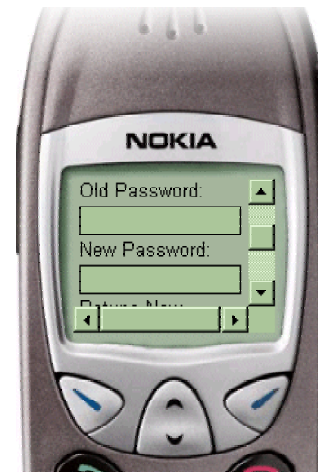
the users' login session. The session id will become invalid once the user logout the session or the session is idle for long time that the automatic logout is activated. With



this mechanism, even if people can peep another user's session id or see the session id from browser history, they still cannot get access to the user's account without knowing the password since the session id is different for every login of the user.

4.2.4 User Profile Management

From time to time, users may want to change their phone number and their credit card may expire. Therefore, there is a need for user to modify their own profile in our system. And they may need to change their password from time to time. In order to avoid the situation that a user using a public terminal and forget to logout, and other people try to change the password and change the bank account and other personal information, both



changing of user profile and changing of password require entering the old password so that unauthorized user cannot change the account information without knowing the password.

Internet Resale Store

View Item Status
View History
View/Edit Info
Change Password
Logout

Index
Category for Sale
Category for Buy
Post Products
Registration
Contact Us

Please Edit Your Information:

Password: [*****]

Name in Full: [Chan Tai Man]

Contact number: [98765432]

Address: [The Chinese University of Hong Kong]

Email: [tmchan@cuhk.edu.hk]

Bank Account: [123-4-567890]

Payment method: (By Credit card, Optional)

Visa: [123-4-567890]
Name: [Chan Tai Man]
Expiry Date (MM/YY): [01/00]

Master: [123-4-567890]
Name: [Chan Tai Man]
Expiry Date (MM/YY): [01/00]

[Submit] [Reset]

完成 近端 intranet

4.2.5 Posting Product Information

Registered users can post product information to sell product or request for a product. They can input product name, description and price etc for the product, and can choose the pick-up method for the product of either in person or from counter. When chosen to pick-up in person, the buyer and seller are required to meet at some time to settle the transaction, and when chosen to pick-up from counter, the seller is supposed to put the product in our physical counter, and the buyer choose a time they want to get the product from the counter.

4.2.6 Browsing Products in Categories

Users with and without account in our system are able to browse the products requested and for sale from other users.

The products are divided into different categories.





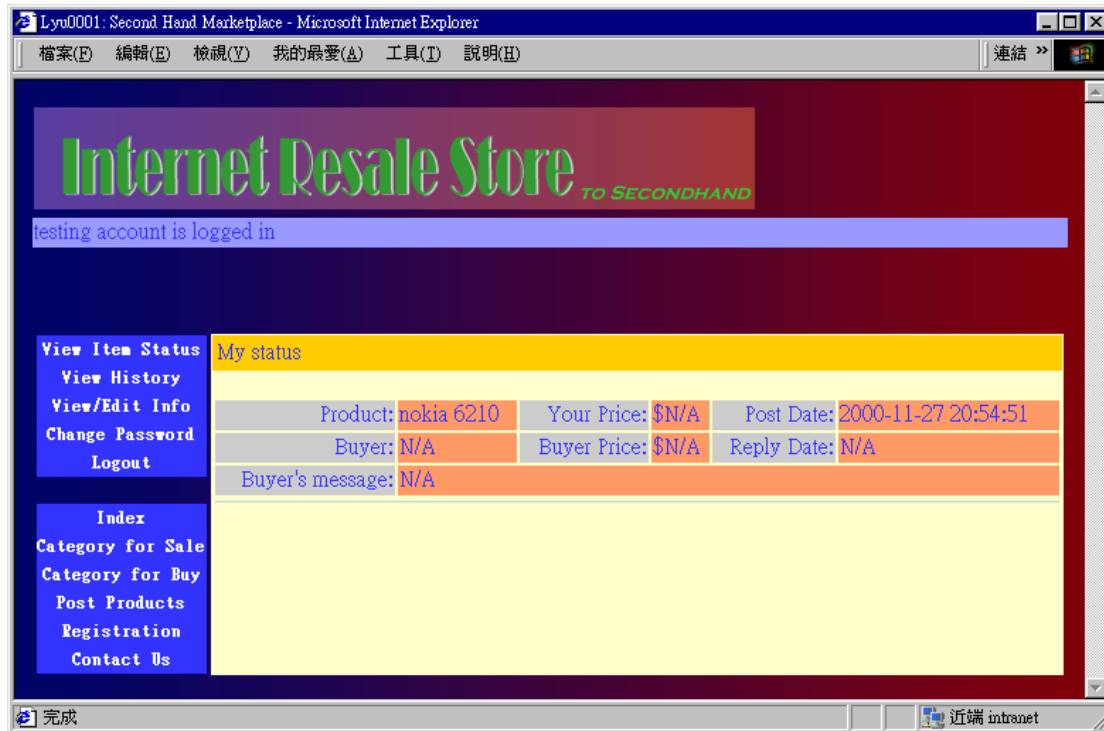
4.2.7 Giving Response

When users see a product for sale or a product request which attract them, they can give response to the posting by sending the price the user suggest, and any reasons and information involved.



4.2.8 View Status

After the users logged in to the system, the users can check the product they post for sale or request to see if there is any person replying. If there is reply, they can choose to adjust the price if they found the price suggested by the respondent is not good enough, or they can accept the response and make a deal.



5. Programming Environment

5.1 Hardware Platform

A PC machine installed with Microsoft Windows NT Server 4.0 and Internet Information Server (IIS) 4.0 is used to host server scripts of our system. And throughout our implementation, platform independence is also considered so that our implementation is made as platform independent as possible.

5.2 Server Configuration

The next step for us is to configure our server to comply with wml files, thus making us wap enabled.

We can serve wml documents from any normal (UNIX or NT) server (all NCSA types and most of the others). This means it is easy to make our own wap hosting system using any regular server (eg. Apache, the most commonly used server in the world today). Basically, all it involves is changing our MIME types file so that it involves wap related file extensions with the appropriate type. For example, the server should hand out all *.wml files with the MIME type text/vnd.wap.wml.

Here are the full MIME types:

Doc. type	Official MIME type	Extension
WML document	text/vnd.wap.wml	.wml
Wireless Bitmap Image	image/vnd.wap.wbmp	.wbmp
WMLScript	text/vnd.wap.wmlscript	.wmls
Compiled WML document	application/vnd.wap.wmlc	.wmlc
Compiled WML Script	application/vnd.wap.wmlscriptc	.wmlsc

The way we install these MIME types is different for differing servers, but here's how we do it in Apache:

First, locate the srm.conf file (in /etc/httpd/conf), or .htaccess files in the directory/ies that we require. Then type the following in:

WAP MIME Types

AddType text/vnd.wap.wml .wml

AddType image/vnd.wap.wbmp .wbmp

AddType text/vnd.wap.wmlscript .wmls

AddType application/vnd.wap.wmlc .wmlc

AddType application/vnd.wap.wmlscriptc .wmlsc

Then, save the file and restart Apache.

To add the MIME types to a Microsoft IIS server, we need to do the following:

On the server console, open the management console; from here we can define the MIME types to either be valid for the entire server or to be valid for separate directories. To add a MIME type to a directory all we need to do is right click on the directory in which we want to add the MIME type, select the HTTP headers tag and click the file types button at the lower right. Click 'new type' and supply the extension and the content type as listed above. Just as simple as for Apache.

To add a MIME type to an entire server; right click on the server, click the 'file types' button and follow the instructions above.

5.3 Programming Language

We have chosen PHP as the server scripting language. With the current distribution of PHP, there are both windows version and unix/linux version available, so that our programs are able to port to other platforms with ease.

5.4 Database System

A MySQL database system is installed to serve as a database backend. MySQL system also have windows version and unix/linux version so that our implementation

will not be limited to a single platform.

5.5 Program Testing

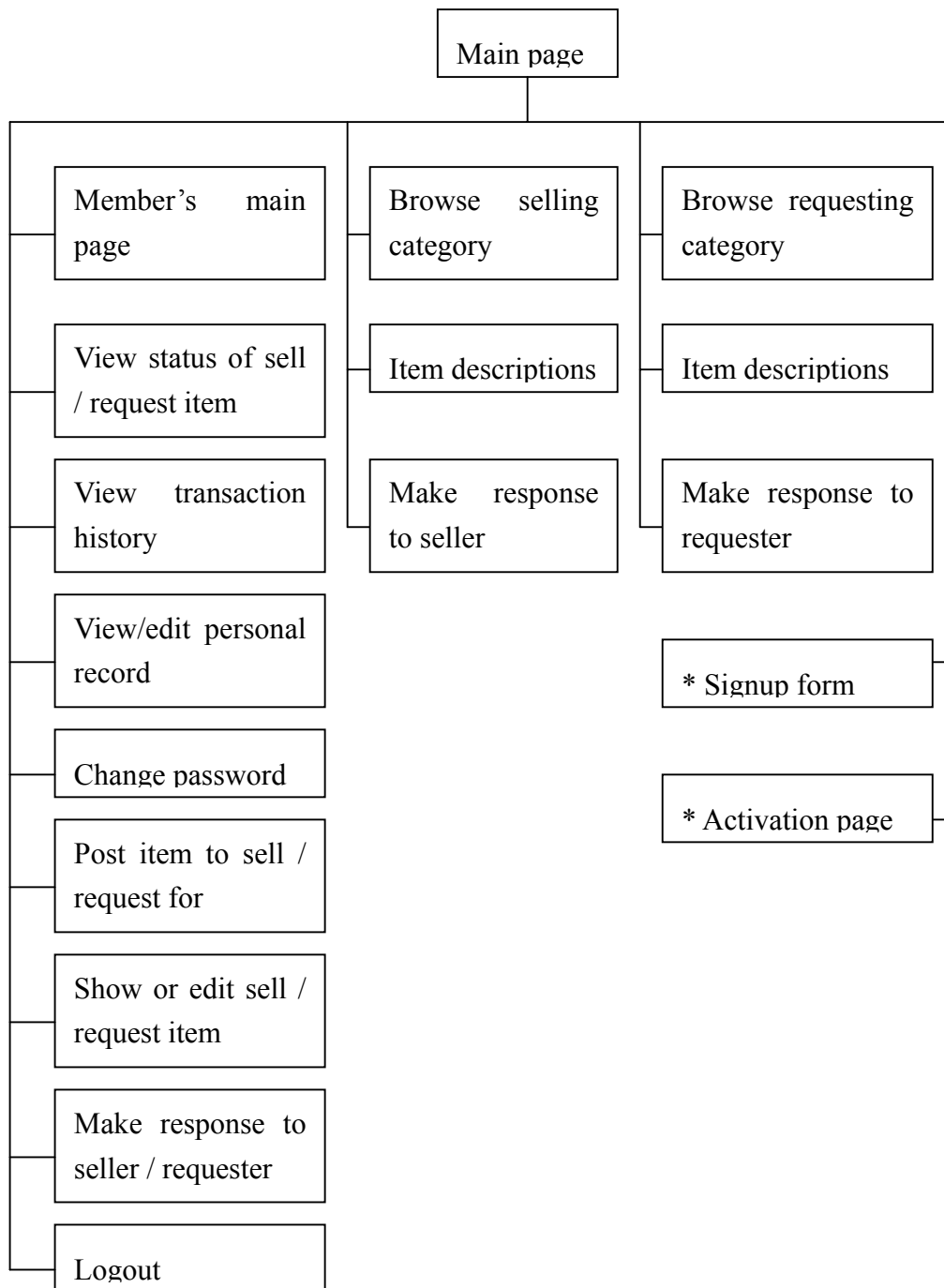
Since current web browser is not able to render WML pages, so in addition to using web browsers to test our system, WAP simulators are used to test our WAP version of our system. Since different simulators have different behaviors, and simulators are more 'forgiving' than actual WAP phones, so WAP phones and WAP service of mobile network providers are occasionally required to test our system for errors.

6. System Implementation

6.1 Overview

In our system, there is a database storing users information and product information post by users. A set of server scripts are generating web pages for users using web browser to access our system, and there is another server scripts generating WML pages for users using WAP phones (or WAP enabled device) to access our system. The two set of scripts both interact with the database and generate appropriate pages to users.

6.2 Site Map



6.3 Database

Database: sechand

Tables
sechand_action
sechand_cat
sechand_item
sechand_payment
sechand_pickup
sechand_reply
sechand_session
sechand_user

Table: Sechand_action

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI		auto_increment
description	varchar(60)				

Table: Sechand_cat

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI		auto_increment
description	varchar(60)				

Table: Sechand_item

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI		auto_increment
owner	int(11)			0	
category	int(11)			0	
action	int(11)	YES			

pickup	int(11)	YES			
payment	int(11)	YES			
title	varchar(60)	YES			
description	text	YES			
district	varchar(20)	YES			
price	int(11)	YES			
image	varchar(255)	YES			
postdate	datetime			0000-00-00 00:00:00	

Table: Sechand_payment

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI		auto_increment
method	varchar(20)				

Table: Sechand_pickup

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI		auto_increment
description	varchar(60)				

Table: Sechand_reply

Field	Type	Null	Key	Default	Extra
itemid	int(11)			0	
replyerid	int(11)			0	
replydate	datetime			0000-00-00 00:00:00	
price	int(11)	YES			
reason	text	YES			

Table: Sechand_session

Field	Type	Null	Key	Default	Extra
id	char(32)		PRI		
userid	int(11)			0	
expire	datetime			0000-00-00 00:00:00	

Table: Sechand_user

Field	Type	Null	Key	Default	Extra
Id	int(11)		PRI		auto_increment
username	varchar(60)				
password	varchar(32)	YES			
fullname	varchar(255)	YES			
contactnumber	varchar(60)	YES			
address	varchar(255)	YES			
email	varchar(255)	YES			
bankacct	varchar(60)	YES			
visaname	varchar(60)	YES			
visanum	varchar(127)	YES			
visadate	varchar(10)	YES			
mastername	varchar(60)	YES			
masternum	varchar(127)	YES			
masterdate	varchar(10)	YES			

7. Problems Encountered and Lesson Learned

7.1 Problems Encountered

7.1.1 Department Firewall

Since the lab machines of our department are behind a firewall, WAP simulator programs are unable to access WAP pages since it need to connect to a WAP gateway through a specific port. Luckily, there are a few online WAP simulator of which most processing are done on server side so that there is no need for connecting to WAP gateway directly.

Moreover, the IP of the lab machines and IP of wireless LAN of our department are not accessible from outside network, we failed to try installing a web server in removable hard-disk and laptop connecting to department wireless LAN and have that server pages access by WAP phone. As a result, we stick to using the NT machine with fixed IP as hosting server.

7.1.2 Perl with Oracle

Many efforts have been put into installing Perl module in the NT machine to access oracle server of the department, but without success. In view of time remain to do the implementation, we decided not to continue to put more effort in solving the problem, and instead switch to use PHP and MySQL, which are considered by web site developers as “perfect match for small to medium scale database-driven web sites”.

7.2. Things Learned

7.2.1 E-commerce Models

During our study of current e-commerce situation, we have a better idea of the current

e-commerce market, how they work, and how the system are designed. This allow us to have a hand-on experience on designing and implementing an e-commerce system.

7.2.2 Wireless Internet Access

As we are implementing a WAP enabled system, we need to get familiar with the Wireless Application Protocol (WAP), which is a relatively new protocol. We need to get into details on what the current protocol allows us to do, and what it can't do, and have to get expectations and prediction on the future of this protocol so that we can have better planning of our project. Since the computer and Internet world is changing every minute, experience and skills on adapting and expecting changes are valuable.

8. Conclusion and Future Plans

8.1 Conclusion

In this semester, we have read though many materials on e-commerce and the WAP, and have hand-on experience to set-up a web site, and have written a simple site of a e-commerce application, second hand market. Throughout the problem we have faced, we have learned and have experience on the workflow of how a e-commerce on the wireless network works.

8.2 Planning for XML and XSL

The current implementation of the system requires writing two different sets of server scripts to generate HTML pages for web users and generate WML pages for WAP users. In view of the fact that the data obtained from database are the same for both web and WAP version of the system, it will be beneficial if we can have the database queries applied once and have the data ready, and two different display processor interpret the data and generate appropriate output.

Using XML and XSL can address this issue. Database queries can be applied only once and have the data obtained represented in XML format, then apply different XSL documents to the XML document to generate different forms of output such as HTML and WML documents. This also allows possible future extensions to generate different output formats with appropriate of XSL documents in case other output standards appear, and maybe have different XSL documents for different WAP phone browsers.

9. References

- [1] ARC Group (1999), “Wireless Industry Showing us the Future for the Internet”, **Paper**, <http://www.the-arc-group.com/press/wi.htm> (1999-06-23)
- [2] AU-System Radio (1999) “WAP White Paper”, **Paper**, <http://www.wapguide.com/WhitePaper/WAPWP3.pdf> , (1999-06-16)
- [3] Bittlingmayer, G. (1999), “Economics of Information Age Industries”, **Paper**, <http://www.gsm.ucdavis.edu/~gnbittli/economics.html> (1999-10-21)
- [4] Buckingham, Simon (1999, 1), “An introduction to SMS – Short Message Service”, **Paper** http://www.gsmworld.com/gsmdata/sms_01.htm, (1999-09-21)
- [5] Buckingham, Simon (1999, 2), “An introduction to GPRS – General Packet Radio Service”, **Paper**, http://www.gsmworld.com/gsmdata/gprs_01.htm, (1999-09-21)
- [6] Ericsson (1999), **Web page**, <http://www.ericsson.com/wireless/products/mobsys/gsm/subpages/wise/subpages/hscsd.shtml>, (1999-09-21)
- [7] Intel (1999), **Web page**, <http://www.gsmdata.com/whatsgsm.htm> (1999-09-21)
- [8] Mobile Lifestream Ltd. (1999, 1): FutureFoneZone, **Web page** <http://www.dataonsms.com/FutureFoneZone/default.htm> (1999-06-23)
- [9] Mobile Lifestream Ltd. (1999, 2), “Data on WAP”, **Paper** <http://www.links2mobile.com/wp/wp4.htm> (1999-08-20)
- [10] Mobile Lifestream Ltd. (1999, 3), “SMS Center selection criteria”, **Web page**, <http://www.links2mobile.com/wp/wp2.htm> (1999-08-14)

-
- [11] Nokia (1998), "Getting Started", **Paper**,
http://www.forum.nokia.com/wap_developer/download/sgs.pdf
(1999-06-28)
- [12] Nordin, B. (1999) "Computer Sweden: Dum, dummare, Siemens", **Web Page**, <http://nyheter.idg.se/display.pl?ID=991014-CSD2> (1999-10-14)
- [13] Post & Telestyrelsen (PTS) (1999), "Sund konkurrens på framtidens mobiltelefonmarknad", **Paper**, <http://www.pts.se/pdf/regutotalnyx.pdf>,
(1999-10-06)
- [14] PricewaterhouseCoopers (1999), "Sverige: Kunskap & Lösningar" **Web page**,
http://www.pricewaterhousecoopers.com/se/swe/ins-sol/survey-rep/infokomsektorn_seswe.html, (1999-10-30)
- [15] ScienCentral, Inc, and The American Institute of Physics (1999), "Moore's Law", <http://www.pbs.org/transistor/background1/events/mooreslaw.html>
(1999-08-20)
- [16] Govekar S.(1998), "Shailesh Govekar's Network Glossary" **Web page**,
<http://www.geocities.com/ResearchTriangle/Lab/6601/b.html>
(1999-11-08)
- [17] Techmall (1997) Nokia Developed Smart Messaging Defines New Wireless Service Category, **Web page**,
<http://www.techmall.com/techdocs/TS970829-12.html> (1999-06-23)
- [18] Teleaffärer, "Ericsson och Nokia i samarbete kring WAP", 1999-06-30
- [19] UMTS Forum (1999), **Web site**, <http://www.umts-forum.org>
(1999-09-21)
- [20] Unwired Planet, Inc. (1999), "Wireless Internet today: The Wireless

- [21] Application Protocol”, **Paper**,
<http://www.phone.com/pub/feb99WAPWP.pdf> (1999-06-16)
- [22] Walsh, N. (1998) “What is XML?” **Paper**,
<http://www.xml.com/pub/98/10/guide1.html> (1999-08-24)
- [23] WAP Forum (1999, 1), **Web site** www.wapforum.org
- [24] WAP Forum (1999, 2), “The WAP Roadmap”, **Paper**,
http://www.wapforum.org/what/docs/WAPRoadmap_v0.8.pdf
(1999-06-16)
- [25] WAP Forum (1999, 3), “Wireless Application Protocol Architecture Specification”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WAPArch-19980430.pdf> (1999-08-12)
- [26] WAP Forum (1999, 4), “Wireless Markup Language Specification”,
Paper,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WML-19990616.pdf> (1999-08-12)
- [27] WAP Forum (1999, 5), “Wireless Application Environment Specification”,
Paper,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WAESpec-19990524.pdf> (1999-08-16)
- [28] WAP Forum (1999, 6), “WMLScript language specification”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WMLScript-19990617.pdf> (1999-08-18)
- [29] WAP Forum (1999, 7), “Wireless Telephony Application Specification”,
Paper,

-
- <http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WTA-19990716.pdf>
(1999-08-23)
- [30] WAP Forum (1999, 8), “Wireless Session Protocol Specification”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WSP-19990528.pdf>
(1999-08-23)
- [31] WAP Forum (1999, 9), “Wireless Transaction Protocol Specification”,
Paper,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WTP-19990611.pdf>
(1999-08-24)
- [32] WAP Forum (1999, 10), “Wireless Transport Layer Security”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WTLS-19990211.pdf>,
(1999-08-24)
- [33] WAP Forum (1999, 11), “Wireless Datagram Protocol Specification”,
Paper,
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WDP-19990514.pdf>
(1999-08-24)
- [34] WAP Forum (1999, 12) “WAP Forum Policy Statements and Notes”,
Paper, <http://www.wapforum.org/what/policy.htm>, (1999-08-24)
- [35] WAP Forum (1999, 13) “Wireless Application Protocol Identity Module
Specification”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=PROP-WIM-19991105.pdf>
(1999-11-20)
- [36] WAP Forum (1999, 14) “Wireless Application Protocol WMLScript
Crypto Library Specification”, **Paper**,
<http://www1.wapforum.org/tech/terms.asp?doc=PROP-WMLScriptCrypto>

-19991105.pdf, (1999-11-20)

[37] whatis.com Inc: (1999), “What Is...a smart card (a definition)”, **Web page**

<http://www.whatis.com/>, (1999-11-20)

[38] Internet.com: “US Internet users going wireless”, **Web Page**

http://cyberatlas.internet.com/markets/wireless/article/0,1323,10094_342491,00.html, (2000-8-15)

[39] Internet.com: “E-Commerce, Mobile Access Drawing Interest from Net Users”,

Web Page

http://cyberatlas.internet.com/big_picture/geographics/article/0,,5911_494701,00.html, (2000-10-25)

[40] Roper Starch Worldwide: "Demand for E-Commerce Activities and New Devices

Surges Ahead”, **Web Page** <http://www.roper.com/news/content/news218.htm>,

(2000-10-24)

[41] Forrester.com: “Europe's Mobile Internet Beyond The WAP Backlash”, **Report**

<http://www.forrester.com/>, (2000-9)

Appendix. Overview of XML and XSL

A.1 Definition of XML

XML (extensible Markup Language) is a markup language for documents containing structured information.

Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.). Almost all documents have some structure.

A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.

A.2 Component of XML

The number of applications currently being developed that is based on, or make use of, XML documents is truly amazing. For our purposes, the word "document" refers not only to traditional documents, like this one, but also to the myriad of other XML "data formats". These include vector graphics, e-commerce transactions, mathematical equations, object meta-data, server APIs, and a thousand other kinds of structured information.

XML is based on the concept of *documents* composed of a series of *entities*.

('Entity' is the English spelling of the French word 'entité', the Teutonic equivalent of which is 'thing'. Those familiar with modern programming techniques will probably

be more comfortable using the word 'object'. All these terms are synonymous.) Each entity can contain one or more logical *elements*. Each of these elements can have certain *attributes* (properties) that describe the way in which it is to be processed. XML provides a formal syntax for describing the relationships between the entities, elements and attributes that make up an XML document, which can be used to tell the computer how it can recognize the component parts of each document.

XML differs from other markup languages in that it does not simply indicate where a change of appearance occurs, or where a new element starts. XML sets out to clearly identify the boundaries of every part of a document, whether it be a new chapter, a piece of boilerplate text, or a reference to another publication.

To allow the computer to check the structure of a document users must provide it with a document type definition that declares each of the permitted entities, elements and attributes, and the relationships between them.

Document Type Definition (DTD), which is a formal model that users of XML can check that each component of document occurs in a valid place within the interchanged data stream. An XML DTD allows computers to check, for example, that users do not accidentally enter a third-level heading without first having entered a second-level heading, something that cannot be checked using the HyperText Markup Language (HTML) previously used to code documents that form part of the World Wide Web (WWW) of documents accessible through the Internet.

A.3 Advantage of using XML

XML was created so that richly structured documents could be used over the web in a flexible way. The only viable alternatives, HTML is not practical for this purpose. HTML comes bound with a set of semantics and does not provide arbitrary structure.

While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, a more satisfactory solution for the creation and long-time storage of complex documents. XML will be the standard procedure for web delivery. However, the database system still exists. We dump out the resultant queries as XML formatted. That is, the database gets back to its original job of just storing data and keeping indexing information. Then on top of that we layer some APIs to extract the raw information and format it into XML. The main advantage is that we can define one format and then have different "views" of it, such as HTML and WML pages.

Because XML is ASCII-based, it promises to make deploying Internet-based distributed applications easier than is possible with binary-based messaging technologies like DCOM (Distributed Component Object Model) or CORBA (Common Object Request Broker Architecture). But, this doesn't mean that binary-based architectures should be tossed in favor of XML - they often work very well together. XML is a means of providing more efficient data exchange. The ultimate goal is to eliminate data redundancy and foster data reuse.

A.4 XML Development Goals

It shall be straightforward to use XML over the Internet. Users must be able to view

XML documents as quickly and easily as HTML documents. In practice, this will only be possible when XML browsers are as robust and widely available as HTML browsers, but the principle remains.

1. XML shall support a wide variety of applications. XML should be beneficial to a wide variety of diverse applications: authoring, browsing, content analysis, etc. Although the initial focus is on serving structured documents over the web, it is not meant to narrowly define XML.
2. It shall be easy to write programs that process XML documents. The colloquial way of expressing this goal while the spec was being developed was that it ought to take about two weeks for a competent computer science graduate student to build a program that can process XML documents.
3. The number of optional features in XML is to be kept to an absolute minimum, ideally zero. Optional features inevitably raise compatibility problems when users want to share documents and sometimes lead to confusion and frustration.
4. XML documents should be human-legible and reasonably clear. If we don't have an XML browser and we've received a hunk of XML from somewhere, we ought to be able to look at it in our favorite text editor and actually figure out what the content means.
5. The XML design should be prepared quickly. Standards efforts are notoriously slow. XML was needed immediately and was developed as quickly as possible.
6. The design of XML shall be formal and concise.
7. XML documents shall be easy to create. Although there will eventually be

sophisticated editors to create and edit XML content, they won't appear immediately. In the interim, it must be possible to create XML documents in other ways: directly in a text editor, with simple shell and Perl scripts, etc.

A.5 Apply to our project

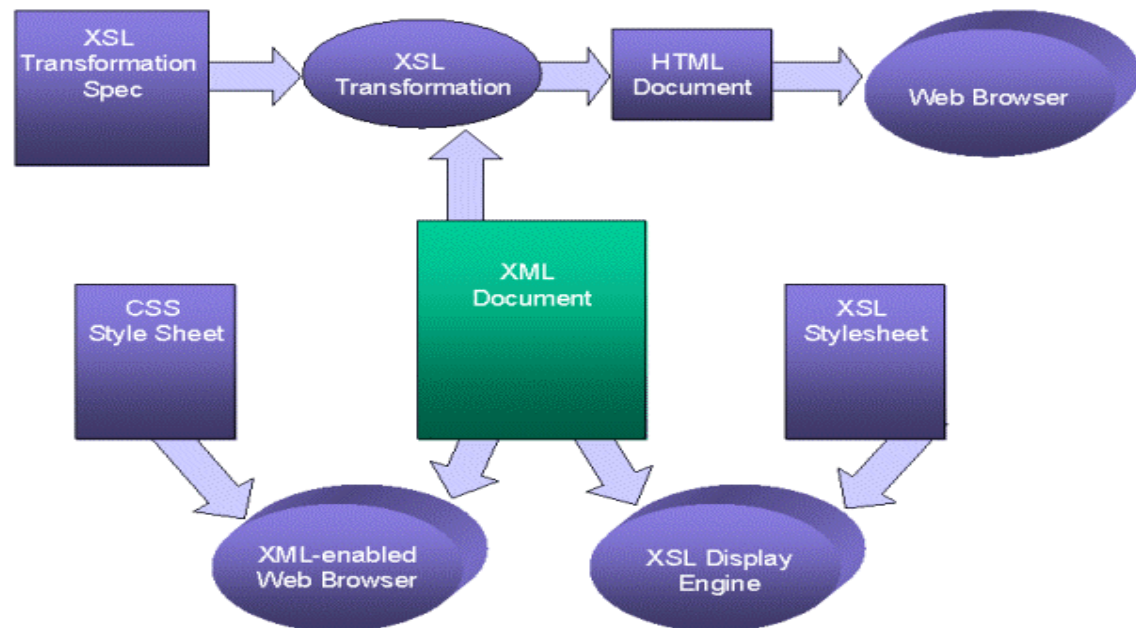
XML allows users to:

- bring multiple files together to form compound documents
- identify where illustrations are to be incorporated into text files, and the format used to encode each illustration
- provide processing control information to supporting programs, such as document validators and browsers
- add editorial comments to a file

However, that XML is not:

- a predefined set of tags, of the type defined for HTML, that can be used to markup documents
- a standardized template for producing particular types of documents.

XML was not designed to be a standardized way of coding text: in fact it is impossible to devise a single coding scheme that would be suit all languages and all applications.



Instead XML is formal language that can be used to pass information about the component parts of a document to another computer system. XML is flexible enough to be able to describe any logical text structure, whether it is a form, memo, letter, report, book, encyclopedia, dictionary or database.

A.6 XSLT

The Extensible Stylesheet Language Transformations (XSLT) W3C recommendation describes a transformation vocabulary used to specify how to create new structured information from existing XML documents.

Unlike with a programming language, we don't need to be a programmer to successfully describe how to transform our information. XSLT implements transformation "by example", not just "by program logic", and builds in support for the kinds of transformation typically needed to present information.

Our objective, as an XSLT stylesheet writer, is to give an XSLT engine examples of how each of the constructs in our information is supposed to be structured once it has been transformed. We create these examples as "templates" of the result and we tell the engine when these templates get added to the resulting tree our transformation is creating. Our stylesheet templates can include our instructions to the XSLT engine to hunt down information anywhere in our input XML file, or many XML input files, to fill in holes in our template where our own information belongs.

When using the Extensible Stylesheet Language (XSL) formatting objects to present our information, our stylesheet objective is to transform our information into a hierarchy exclusively comprised of the XSL formatting object vocabulary. A rendering engine then takes this result hierarchy and interprets the semantics of the XSL vocabulary to produce our desired rendition, all without using a single construct of our own vocabulary, because we've transformed our own information into rendering information.

With XSLT we are not restricted to present our information in the same order we created it, nor are we required to present all of our information. Also, we can traverse our source XML multiple times if we need to reuse our information more than once in a single result, perhaps to simultaneously create tables of content and internal cross references described abstractly in our raw structured content.

The XSLT recommendation describes how XSLT engines can choose to support different ways to serialize the templates we have added, in combination with the information we glean from our source tree, to the result tree. Our result can be

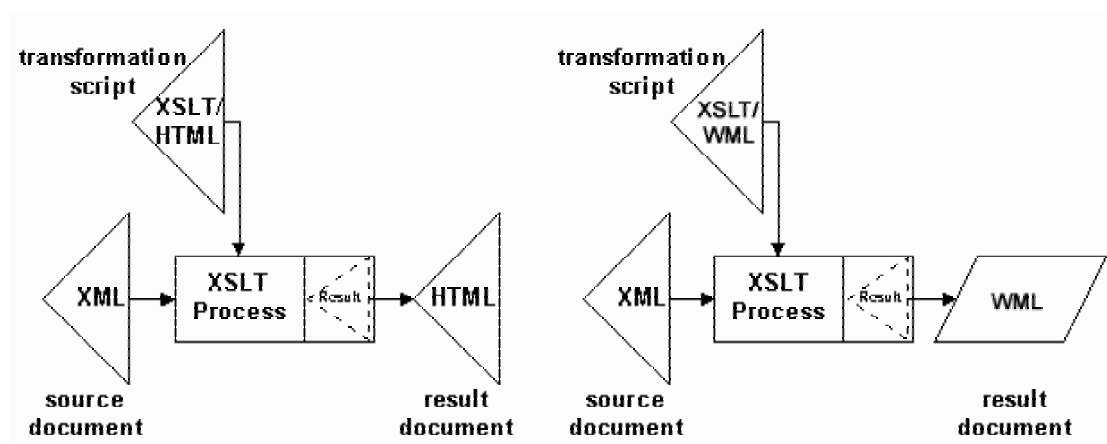
realized using XML syntax conventions, HTML syntax conventions (with or without CSS) or WML syntax conventions.

Think of the possibilities: with XSLT we can take our structured XML information and synthesize new instances for our colleagues and customers to use, or build HTML/CSS web pages from data stores, or feed other systems with flat text representations of our data, or create operating system scripts, etc.

This gives us the freedom to organize the information the way we want from the beginning, to best meet our own business requirements, and still fulfill our obligations and desires to utilize our information in a myriad of ways downstream.

A.7 Transformation from XML to non-XML using XSL

An XSLT processor may choose to recognize the stylesheet writer's desire to serialize a non-XML representation of the result tree:



The XSLT Recommendation documents two non-XML tree serialization methods that can be requested by the stylesheet writer. When the processor offers serialization, it is only obliged to reify the result using XML lexical and syntax rules, and may support producing output following either HTML or WML lexical and syntax rules.

A.8 Example of using XSLT:

Here is an example showing that we use one XML, two XSL together with a PHP script to make use of XSLT that can generate the HTML and WML files separately according to the user's request. Clients only need to request the URL of test.php file for both web and WAP browsers.

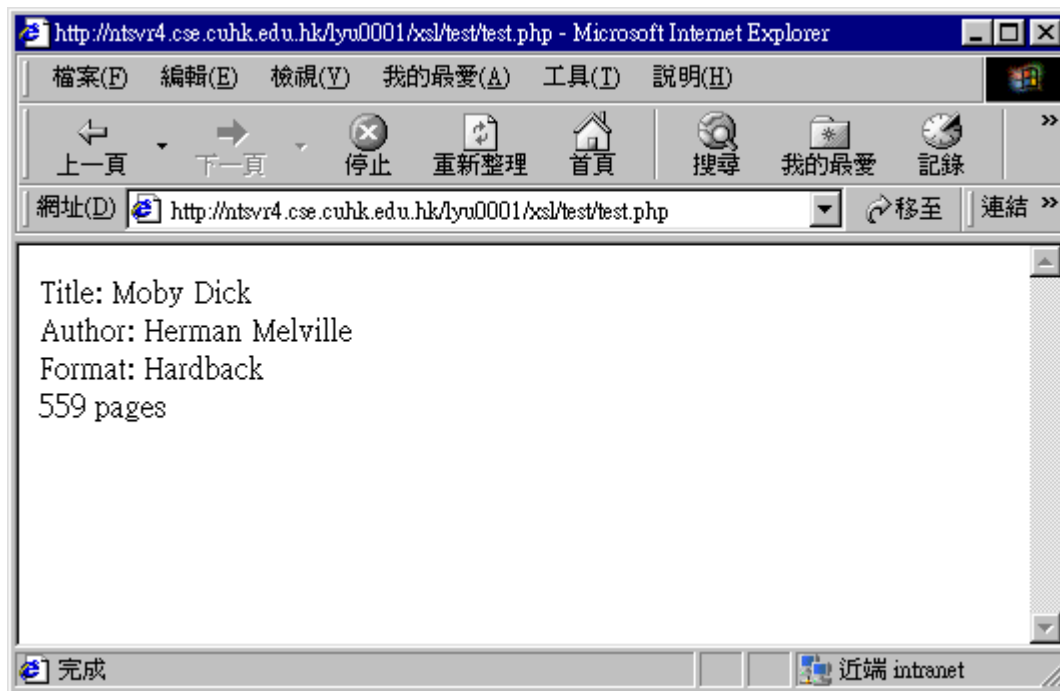
A simple XML document (test.xml)

```
<?xml version="1.0"?>
<book>
  <title>Moby Dick</title>
  <author>Herman Melville</author>
  <format>
    <spine>Hardback</spine>
    <pages number="559" />
  </format>
</book>
```

A simple PHP script (test.php)

```
<?php
  if (ereg("wml",$HTTP_ACCEPT))
  {
    $xslfile = "wml.xsl";
  }
  else
  {
    $xslfile = "html.xsl";
  }
  $xslt = xslt_create();
  $result="";
  $xslldoc=join("",file($xslfile));
  $xmldoc=join("",file("test.xml"));
  xslt_process($xslldoc,$xmldoc,$result);
```

```
echo $result;
@xslt_free($xslt);
?>
```



A simple XSL document for HTML output (html.xsl)

```
<?xml version="1.0"?>
<xsl:stylesheet                                version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        Title: <xsl:value-of select="book/title"/>
        <br/>
        Author: <xsl:value-of select="book/author"/>
        <br/>
        Format: <xsl:value-of select="book/format/spine"/>
        <br/>
        <xsl:value-of select="book/format/pages/@number"/> pages
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

A simple XSL document for WML output (wml.xsl)

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output
    method="xml"
    doctype-public="-//WAPFORUM//DTD WML 1.1//EN"

doctype-system="http://www.wapforum.org/DTD/wml_1.1.xml"/>

  <xsl:template match="/">
    <wml>
      <card>
        <p>
          Title: <xsl:value-of select="book/title"/>
        </p>
        <p>
          Author: <xsl:value-of select="book/author"/>
        </p>
        <p>
          Format: <xsl:value-of select="book/format/spine"/>
        </p>
        <p>
          <xsl:value-of select="book/format/pages/@number"/> pages
        </p>
      </card>
    </wml>
  </xsl:template>
</xsl:stylesheet>

```

