

LỜI CẢM ƠN

Trước khi trình bày nội dung nghiên cứu của luận văn, chúng em xin dành những dòng đầu tiên để gửi lời cảm ơn chân thành đến:

Cô Huỳnh Thụy Bảo Trân, người đã tận tình hướng dẫn và tạo mọi điều kiện tốt nhất cho chúng em trong suốt thời gian thực hiện luận văn.

Chúng em xin gửi lời cảm ơn chân thành đến quý Thầy Cô trong Khoa Công nghệ thông tin, trường Đại học Khoa học Tự nhiên Thành phố Hồ Chí Minh đã giảng dạy, hướng dẫn và giúp đỡ chúng em thực hiện tốt luận văn này.

Chúng con cũng xin gửi lời cảm ơn sâu sắc đến ba mẹ và gia đình đã luôn chăm sóc và động viên trong suốt quá trình học tập, nghiên cứu để chúng con có thể đạt được kết quả như ngày hôm nay.

Và cũng xin chân thành cảm ơn sự giúp đỡ hết sức nhiệt tình của các bạn Hồ Phạm Thái Vinh, Chu Hoàng Nam, Đặng Xuân Hữu và Nguyễn Quốc Bảo đã hỗ trợ thiết bị giúp chúng tôi hoàn thành tốt đề tài này.

Tp.Hồ Chí Minh, ngày 12 tháng 6 năm 2005

Nhóm sinh viên thực hiện:

Nguyễn Thanh Phong - Nguyễn Ngọc Phương Vi

MỤC LỤC

LỜI NÓI ĐẦU	9
PHẦN I.	
TÌM HIỂU CÔNG NGHỆ WAP	12
Chương 1. TỔNG QUAN VỀ WAP	13
1.1. Giới thiệu.....	13
1.2. Kiến trúc ứng dụng WAP.....	14
1.2.1. WAP Client.....	16
1.2.2. WAP Proxy, WAP Gateway và WAP Server.....	17
1.3. Ngăn xếp giao thức WAP – WAP Protocol stack.....	23
1.3.1. Wireless Application Environment – WAE.....	25
1.3.2. Wireless Session Layer – WSP.....	26
1.3.3. Wireless Transaction Layer – WTP.....	27
1.3.3.1. Yêu cầu không tin cậy – Unreliable request.....	28
1.3.3.2. Yêu cầu có thể tin cậy – Reliable request.....	28
1.3.3.3. Yêu cầu tin cậy với một thông điệp kết quả.....	29
1.3.4. Wireless Transport Layer Security – WTLS.....	30
1.3.5. Wireless Datagram Protocol – WDP.....	32
1.4. Vấn đề bảo mật trên WAP.....	32
1.4.1. So sánh các mô hình bảo mật.....	32
1.4.1.1. Bảo mật trên Internet.....	32
1.4.1.2. Bảo mật trên WAP.....	34
1.4.2. Vấn đề bảo mật trên WAP.....	37
1.4.2.1. Chứng thực người dùng.....	37
1.4.2.2. WAP Gateway.....	38
1.4.2.3. TLS và WTLS.....	39
Chương 2. SỰ PHÁT TRIỂN CỦA CÁC NGÔN NGỮ ĐÁNH DẤU PHỤC VỤ CHO WAP.....	43
2.1. Ngôn ngữ đánh dấu (Markup-Language).....	43
2.2. WAP và WML.....	45
2.3. XHTML cơ sở.....	46
2.4. XHTML Mobile Profile.....	47
2.5. WAP CSS.....	47
2.6. So sánh XHTML với HTML, WML.....	49
2.6.1. Sự khác nhau giữa XHTML và HTML.....	49
2.6.2. Sự khác nhau giữa XHTML và WML 1.x.....	51
2.7. Các giao thức chuyển tải WML và XHTML.....	55
2.8. Cuộc cách mạng của trình duyệt WAP.....	57
PHẦN II.	
CÔNG CỤ TÌM KIẾM.....	60
SEARCH ENGINE	60
Chương 3. TỔNG QUAN VỀ MÁY TÌM KIẾM.....	61

3.1. Sơ lược về máy tìm kiếm	61
3.2. Phân loại máy tìm kiếm	61
3.2.1. Máy tìm kiếm meta	62
3.2.2. Máy tìm kiếm thông thường	65
3.2.2.1. Nguyên lý hoạt động của một máy tìm kiếm	65
3.2.2.2. Hệ thống thu thập dữ liệu (robot, spider, crawler...)	66
3.2.2.3. Hệ thống phân tích và lập chỉ mục dữ liệu	66
3.2.2.4. Hệ thống tìm kiếm (truy vấn dữ liệu)	67
Chương 4. MÁY TÌM KIẾM HỖ TRỢ THIẾT BỊ DI ĐỘNG	68
4.1. Tìm hiểu các dịch vụ tìm kiếm hỗ trợ thiết bị di động hiện có	68
4.1.1. Google Mobile Search	68
4.1.2. Các máy tìm kiếm trên WAP hiện nay	71
4.2. Chuyển đổi các tài liệu sẵn có từ chuẩn web sang WAP	71
4.2.1. Nhu cầu chuyển đổi	71
4.2.2. Hoạt động của các bộ chuyển đổi	73
4.2.3. Điều kiện quyết định khả năng chuyển đổi nội dung một tài liệu	74
4.2.4. Các ưu điểm	75
4.2.5. Các nhược điểm	76
PHẦN III.	
ỨNG DỤNG MINH HỌA	77
Chương 5. PHÂN TÍCH THIẾT KẾ HỆ THỐNG “CÔNG CỤ TÌM KIẾM HỖ TRỢ THIẾT BỊ DI ĐỘNG”	78
5.1. Khảo sát hiện trạng	78
5.2. Phân tích và xác định yêu cầu	79
5.3. Mô hình hoạt động	81
5.3.1. Mô hình chung	81
5.3.2. Mô hình chi tiết máy tìm kiếm	82
5.4. Mô hình Use-case	82
5.4.1. Xác định Actor và Use-case	82
5.4.2. Mô hình Use-case	83
5.5. Đặc tả Use-case	84
5.5.1. Tìm kiếm	84
5.5.2. Thay đổi thông số hiển thị kết quả tìm kiếm	84
5.5.3. Chuyển đổi nội dung	85
5.6. Module Máy tìm kiếm – Search Engine	86
5.6.1. Phần thu thập dữ liệu	86
➤ Thuật toán duy trì thông tin cho máy tìm kiếm	87
5.6.2. Phần thu thập dữ liệu	89
5.6.2.1. Thiết kế dữ liệu	91
5.6.2.1.1. Bảng định danh tài liệu	91
5.6.2.1.2. Cấu trúc từ điển chỉ mục	92
5.6.2.1.3. Cấu trúc tập tin chỉ mục nghịch đảo	93
5.7. Module nhận và phân tích query từ người dùng	96
5.7.1. Mô hình hoạt động	96

5.7.2.	Mô hình xử lý.....	98
5.7.3.	Mô tả	98
5.7.4.	Mô hình sequence	99
5.8.	Module chuyển đổi trang web.....	101
5.8.1.	Mô hình hoạt động	101
5.8.2.	Mô tả	101
5.8.3.	Mô hình sequence	102
Chương 6.	CÀI ĐẶT.....	104
6.1.	Hệ thống cơ sở dữ liệu chỉ mục	104
6.2.	Module chuyển đổi trang HTML sang trang WAP	105
6.2.1.	Các lớp cài đặt chính.....	105
6.2.2.	Phần chuyển đổi WAP 1.x – Servlet Html2Wml	105
6.2.3.	Phần chuyển đổi WAP 2.0 – Servlet Html2Xhtml	106
6.3.	Module nhận và phân tích query từ người dùng	106
6.3.1.	Các lớp cài đặt chính.....	106
6.3.2.	Phần xử lý detect trình duyệt	107
6.3.3.	Phần xử lý query	108
6.3.4.	Phần truy vấn cơ sở dữ liệu tìm kiếm kết quả	108
6.3.5.	Giao diện tìm kiếm trên thiết bị di động	109
6.3.5.1.	Giao diện cho trình duyệt hỗ trợ WAP 2.0.....	109
6.3.5.2.	Giao diện cho trình duyệt hỗ trợ WAP 1.x.....	111
Chương 7.	THỬ NGHIỆM	112
7.1.	Thử nghiệm trên các bộ giả lập.....	112
7.2.	Thử nghiệm trên môi trường thực tế	112
Chương 8.	TỔNG KẾT	114
8.1.	Kết quả đạt được	114
8.2.	Hạn chế.....	115
PHẦN IV.		
ĐÁNH GIÁ – HƯỚNG PHÁT TRIỂN		116
TÀI LIỆU THAM KHẢO		118
PHỤ LỤC A		
HƯỚNG TRIỂN KHAI HỆ THỐNG		119
1.	Cơ sở dữ liệu	120
a)	Cài đặt	120
b)	Đăng ký cơ sở dữ liệu với hệ thống ODBC	120
2.	Web server.....	123
a)	Cài đặt	123
b)	Thiết lập server	124
3.	Kết hợp nối hệ thống thông qua bộ giả lập trình duyệt wap của Nokia... ..	126
4.	Đưa trang Web lên internet qua đường truyền ADSL (self-hosting).....	127
a)	Đặc điểm	127
b)	Nguyên tắc	127
c)	Mô hình hoạt động	128

d) Trình tự kết nối từ bên ngoài	129
e) Thiết lập	129
PHỤ LỤC B	
QUẢN TRỊ HỆ THỐNG MOBILE SEARCH ENGINE	137
1. Trang chủ.....	138
a) Cài đặt thiết lập cơ sở dữ liệu Oracle cho hệ thống.....	138
b) Thêm URL và download các trang web	140
c) Kiểm tra thông tin tự diễn của hệ thống	142
PHỤ LỤC C	
BỘ TOOLKIT CỦA NOKIA.....	145
1. Nokia Mobile Internet Toolkit v4.1	146
a) Giới thiệu	146
b) Các chức năng.....	146
2. Nokia WAP Gateway Simulator	150
3. Nokia Browser Simulator.....	152
PHỤ LỤC D	
BỘ WAP CSS	154
1. Các vấn đề được kiểm soát bởi các dạng mẫu	157
2. Áp dụng các kiểu định dạng.....	157
a. Các bảng định dạng bên ngoài.....	158
b. Phần tử style trong đầu đề tài liệu.....	158
c. Phần tử style trong thân tài liệu	158
d. Luật thác nước cho các phần tử mẫu	158
e. Sử dụng các thuộc tính của XHTML	159
3. Những điều cần tránh	162
PHỤ LỤC E	
CÁC NGUYÊN TẮC THIẾT KẾ.....	164
1. Trước khi thiết kế một wapsite.....	165
2. Các nguyên tắc chung cho một thiết kế tốt	165
3. Cần chú ý đến mô hình liên kết.....	166
4. Thiết kế hệ thống phân cấp trong liên kết	167
5. Nguyên tắc thiết kế cho màn hình nhỏ.....	167
6. Đảm bảo các tài liệu phải có kích thước nhỏ	169
7. Tạo các ứng dụng trên điện thoại di động.....	170
8. Đảm bảo các tác vụ tiến hành trôi chảy và sử dụng hợp lý các hình ảnh	170
9. Đảm bảo cấu trúc wapsite dễ dùng đối với người mới sử dụng.....	171
10. Cung cấp vừa đủ thông tin trên một trang.....	171
11. Phản ánh được hành động của người dùng.....	172
12. Hạn chế số lượng và kích thước của màn hình.....	173
13. Thiết lập các thuộc tính chiều cao và chiều rộng màn hình	174
14. Sử dụng bảng một cách cẩn thận	174
15. Cần cân nhắc các tùy chọn	175
16. Loại bỏ các khoảng trắng và các ghi chú trong phần code.....	175

17.	Sử dụng các chỉ dẫn trong phần tiêu đề HTTP trong việc lưu trang	175
18.	Sử dụng mã Unicode cho các nội dung XHTML	176
19.	Sử dụng chính xác các kiểu MIME và mã XHTML	176
20.	Các tiêu đề chỉ dẫn và các nhãn phân tử.....	177
21.	Thực hiện kiểm tra khả năng sử dụng của hệ thống.....	178
PHỤ LỤC F		
DANH SÁCH CÁC THUẬT NGỮ'.....		179

KHOA CNTT

DANH SÁCH HÌNH

Hình 1.1-1: Sự phát triển của điện thoại di động	13
Hình 1.2-1: Kiến trúc trên Internet.....	15
Hình 1.2-2: WAP được dùng truy cập internet	15
Hình 1.2-3: WAP được dùng truy cập intranet	15
Hình 1.2-4: WAP Client.....	16
Hình 1.2-5: Server gốc kết nối trực tiếp với Internet.....	18
Hình 1.2-6: Truy cập Internet thông qua proxy server	18
Hình 1.2-7: Gateway server nằm giữa hai loại mạng khác nhau	19
Hình 1.2-8: Sử dụng WAP proxy/gateway	19
Hình 1.2-9: WAP gateway trong mạng không dây	20
Hình 1.2-10: Các bước thực hiện khi tiến hành một phiên giao dịch WAP	21
Hình 1.2-11: Quá trình bên dịch các yêu cầu tại gateway chuyển đổi giao thức.....	22
Hình 1.2-12: Mô tả chức năng mã hoá/giải mã của WAP gateway	22
Hình 1.3-1: Ngăn xếp WAP 1.x	23
Hình 1.3-2: Ngăn xếp WAP 2.0	24
Hình 1.3-3: Sự kế thừa của ngăn xếp WAP từ mô hình OSI	25
Hình 1.3-4: Unreliable request.....	28
Hình 1.3-5: Reliable request	29
Hình 1.3-6: Reliable request với thông điệp kết quả	29
Hình 1.3-7: WAP gateway điều khiển phiên an toàn.....	31
Hình 1.4-1: Mô hình giao tiếp cổ điển trên Internet	33
Hình 1.4-2: Mô hình giao tiếp trên WAP.....	35
Hình 1.4-3: Giao thức sử dụng trên Tầng Vận Chuyển của WAP 2.0.....	37
Hình 2.1-1: Mối quan hệ giữa các ngôn ngữ đánh dấu.....	45
Hình 2.5-1: Minh họa tham chiếu bảng định dạng bên ngoài tài liệu.....	48
Hình 2.7-1: Sự truyền tải nội dung WML.....	56
Hình 2.7-2: Sự truyền tải nội dung XHTML	56
Hình 2.7-3: Sự truyền tải phối hợp WML và XHTML.....	57
Hình 3.2-1: Mô hình hoạt động của máy tìm kiếm.....	66
Hình 4.1-1: Google Mobile Search	68
Hình 4.1-2: Chức năng tìm kiếm trang web của Google Mobile Search.....	69
Hình 4.1-3: Chức năng tìm kiếm hình ảnh của Google Mobile Search.....	70
Hình 4.2-1: Lược đồ mô tả quá trình chuyển đổi.....	73
Hình 5.3-1: Mô hình hoạt động chung.....	81
Hình 5.3-2: Mô hình hoạt động chi tiết.....	82
Hình 5.4-1: Mô hình Use-case	83
Hình 5.6-1: Lưu đồ xử lý qui trình thu thập thông tin	87
Hình 5.6-2: Lưu đồ xử lý qui trình duy trì thông tin cho máy tìm kiếm.....	88
Hình 5.6-3: Lưu đồ xử lý qui trình lập chỉ mục	90
Hình 5.7-1: Mô hình hoạt động module nhận và phân tích query từ người dùng	97

Hình 5.7-2: Mô hình xử lý quá trình nhận và phân tích query người dùng.....	98
Hình 5.7-3: Quá trình nhận và phân tích query từ người dùng.....	99
Hình 5.7-4: Mô hình sequence.....	100
Hình 5.8-1: Mô hình hoạt động của bộ chuyển đổi.....	101
Hình 5.8-2: Quá trình chuyển đổi trang web.....	102
Hình 5.8-3: Mô hình sequence của module chuyển đổi trang web.....	103
Hình 6.2-1: Các bước chuyển đổi WAP 1.x.....	105
Hình 6.2-2: Các bước chuyển đổi WAP 2.0.....	106
Hình 8.2-1: Mô hình hoạt động quá trình cung cấp IP thông qua đường ADSL (self-hosting).....	128
Hình 8.2-2: Trang web nhà cung cấp dịch vụ DynDNS.....	130
Hình 8.2-3: Giao diện kết nối thành công vào ADSL modem/router.....	131
Hình 8.2-4: Chọn chức năng cấu hình bảng NAT.....	132
Hình 8.2-5: Bảng NAT.....	133
Hình 8.2-6: Đăng ký dịch vụ DNS động với nhà cung cấp dịch vụ.....	134
Hình 8.2-7: Thực hiện unlock các dịch vụ cần thiết.....	135
Hình 8.2-8: Trang showip.com.....	136
Hình 8.2-1: Các kiểu định dạng khác nhau trên các trình duyệt khác nhau.....	156
Hình 8.2-2: Thay đổi kiểu bullet sử dụng mẫu.....	157

DANH SÁCH BẢNG

Bảng 1.4-1: Một vài điểm khác nhau giữa TLS và WTLS	40
Bảng 2.6-1: Các qui tắc XML có trong XHTML nhưng không có trong HTML.....	49
Bảng 2.6-2: Những khác nhau phổ biến giữa XHTML MP với CSS và WML 1.x ..55	
Bảng 2.8-1: Cuộc cách mạng trình duyệt WAP trên các thế hệ điện thoại di động của Nokia.....	58
Bảng 3.2-1: Các công cụ tìm kiếm meta dạng 1	63
Bảng 3.2-2: Các công cụ tìm kiếm meta dạng 2	64
Bảng 4.1-1: Các máy tìm kiếm hỗ trợ WML và WAP	71
Bảng 5.2-1: Xác định yêu cầu	81
Bảng 5.6-1: Bảng định danh tài liệu	91
Bảng 5.6-2: Bảng mục từ	93
Bảng 5.6-3: Cấu trúc trang cho từng mục từ trong tập tin chỉ mục nghịch đảo.....	94

LỜI NÓI ĐẦU

Sự phát triển của hệ thống mạng toàn cầu – Internet đã đưa việc tiếp cận thông tin trở nên dễ dàng hơn bao giờ hết. Với những tiến bộ vượt bậc của các thiết bị phần cứng cũng như các chuẩn mạng di động hiện nay, việc truy cập thông tin từ Internet thông qua các thiết bị di động ngày càng phổ biến hơn. Đó là nhờ vào hệ thống các wapsite đã và đang được phát triển trên nền tảng của công nghệ WAP. Thậm chí các trang web truyền thống cũng đã có thể được truy cập từ các điện thoại di động thông minh có hỗ trợ chuẩn HTTP hoặc các chuẩn WAP mới nhất hiện nay.

Nhưng với lượng thông tin khổng lồ từ Internet, hiện nay vẫn chưa có nhiều dịch vụ tìm kiếm hỗ trợ cho người dùng truy cập từ thiết bị cầm tay – vốn có những hạn chế về khả năng xử lý, hiển thị đồ họa và băng thông mạng thấp. Đến tháng 6/2005, Google chính thức tham gia thị trường dịch vụ tìm kiếm hỗ trợ cho các thiết bị di động có tích hợp thêm chức năng tìm kiếm hình ảnh và các trang web, đẩy lên sự cạnh tranh với một số ít các dịch vụ tìm kiếm khác hiện có.

Một thực trạng khác cũng đặt ra nhiều khó khăn đó là phần lớn thông tin, tài liệu lưu hành trên Internet chỉ có thể hiển thị bằng máy tính desktop. Và vấn đề là làm sao tận dụng được những thông tin có sẵn dưới dạng các trang web truyền thống để có thể hiển thị trên điện thoại di động, những thiết bị vốn rất hạn chế về bộ nhớ, khả năng xử lý và đặc biệt là khả năng hiển thị đồ họa.

Xuất phát từ những vấn đề trên chúng em đã thực hiện đề tài **“TÌM HIỂU WAP VÀ CÔNG CỤ TÌM KIẾM HỖ TRỢ THIẾT BỊ DI ĐỘNG”**. Mục tiêu của đề tài là tìm hiểu về công nghệ WAP và các công nghệ hỗ trợ, tìm hiểu cấu trúc của máy tìm kiếm và xây dựng thử nghiệm một hệ thống tìm kiếm hỗ trợ cho thiết bị di động.

Nội dung của đề tài được chia làm 4 phần gồm 8 chương:

PHẦN I. TÌM HIỂU CÔNG NGHỆ WAP

Chương 1. Tổng quan về WAP : giới thiệu tổng quan công nghệ WAP về kiến trúc cũng như ứng dụng.

Chương 2. Sự phát triển của các ngôn ngữ đánh dấu phục vụ cho WAP: giới thiệu các bước phát triển của các ngôn ngữ đánh dấu và so sánh đặc điểm các ngôn ngữ phục vụ cho công nghệ không dây. Đồng thời tìm hiểu xu hướng mới của các trình duyệt WAP hỗ trợ các ngôn ngữ này.

PHẦN II. CÔNG CỤ TÌM KIẾM – SEARCH ENGINE

Chương 3. Tổng quan về máy tìm kiếm: giới thiệu tổng quan về kiến trúc máy tìm kiếm, tìm hiểu các máy tìm kiếm phổ biến hiện nay.

Chương 4. Máy tìm kiếm hỗ trợ thiết bị di động: tìm hiểu các máy tìm kiếm hiện có hỗ trợ các thiết bị di động, các vấn đề liên quan về việc tận dụng và chuyển đổi những nội dung sẵn có trên web.

PHẦN III. ỨNG DỤNG MINH HỌA

Chương 5. Phân tích thiết kế hệ thống “Máy tìm kiếm hỗ trợ thiết bị di động”: phân tích, thiết kế các chức năng của chương trình, các mô hình, lưu đồ và các vấn đề liên quan đến việc xây dựng ứng dụng.

Chương 6. Cài đặt: Giới thiệu môi trường phát triển và cài đặt ứng dụng, đồng thời thực hiện phân tích một số vấn đề về cài đặt chương trình về chức năng xử lý cũng như thể hiện.

Chương 7. Thử nghiệm: Kết quả thực hiện trên bộ giả lập và trên môi trường thực tế.

Chương 8. Trình bày kết quả đạt được cũng như các vấn đề còn hạn chế.

PHẦN IV. ĐÁNH GIÁ – HƯỚNG PHÁT TRIỂN

Thực hiện đánh giá những kết quả mà đề tài đã đạt được, đồng thời đưa ra hướng phát triển trong tương lai cho đề tài và ứng dụng.

KHOA CNTT

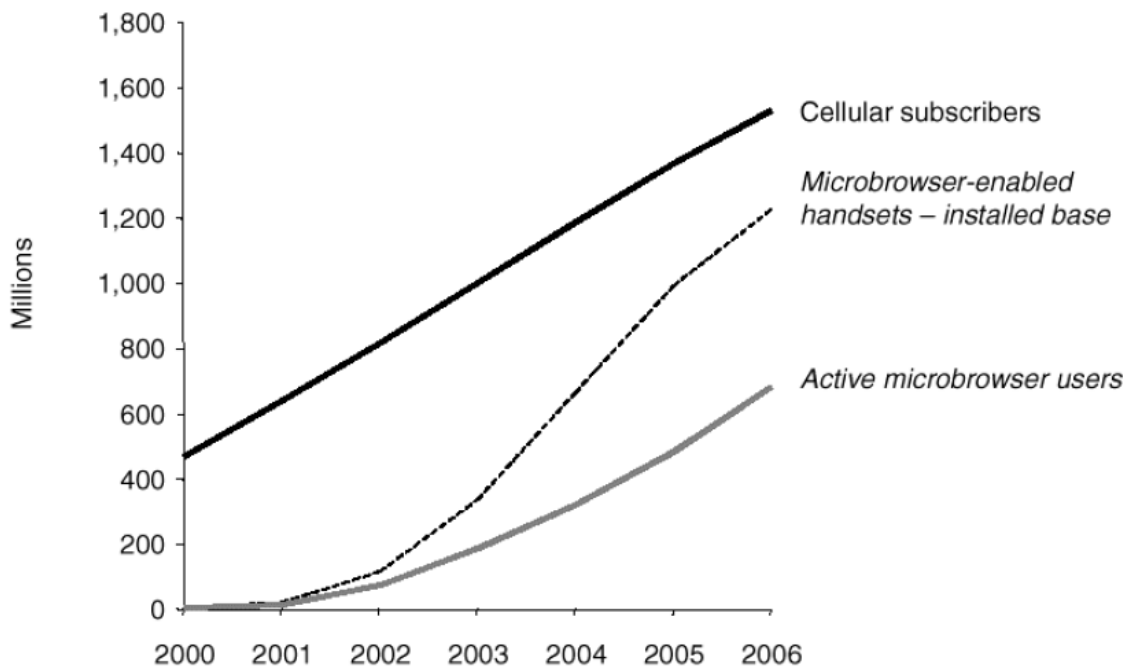
PHẦN I.
TÌM HIỂU CÔNG NGHỆ WAP

Chương 1. TỔNG QUAN VỀ WAP

1.1. Giới thiệu

Trong những năm gần đây, khái niệm về thông tin di động rất được mọi người ưu chuộng, người ta quan tâm ngày càng nhiều hơn đến các thiết bị nhỏ gọn với tốc độ xử lý nhanh.

Đồ thị bên dưới đây được trích từ bài báo có tên *WAP Market Strategies* của Ovum (<http://www.ovum.com>) [7]. Bài báo này đã so sánh và dự đoán sự phát triển của các thế hệ điện thoại di động nói chung, các thế hệ điện thoại di động hỗ trợ Internet, và các loại điện thoại di động hỗ trợ Internet hiện đang được dùng để truy cập thông tin:



Source: Ovum (WAP/A)

Hình 1.1-1: Sự phát triển của điện thoại di động

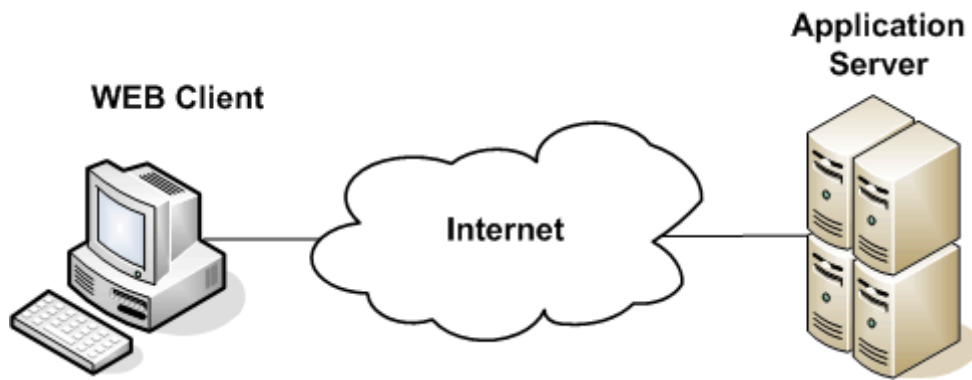
Chính nhu cầu truy cập thông tin từ các thiết bị di động đã mở đường cho các công nghệ không dây phát triển mạnh mẽ. Wireless Application Protocol (WAP) là một dạng đặc tả theo chuẩn công nghiệp mở cho các ứng dụng thực thi trên môi trường mạng không dây, chú trọng vào các ứng dụng trên thiết bị di động, đặc biệt là điện thoại di động. Các tiêu chuẩn này được đưa ra bởi WAP Forum, nhóm này hình thành vào tháng 6 năm 1997 bởi Ericsson, Nokia, Motorola, và Unwired Planet, và hiện tại đã được hàng trăm công ty khác tham gia, bao gồm IBM, Hewlett Packard, Visa, và Microsoft. Theo thống kê chính thức của WAP Forum, những thành viên thuộc WAP Forum là đại diện cho trên 90% nhà sản xuất điện thoại di động trên toàn thế giới.

WAP đã và sẽ được hỗ trợ trên nhiều loại thiết bị, từ đơn giản như điện thoại di động thông thường cho đến những thiết bị thế hệ mới - các điện thoại “thông minh” với màn hình rộng có thể chạy được nhiều ứng dụng; thậm chí là những máy trợ lý cá nhân kỹ thuật số (PDA), các palmtop hay các máy tính với kích thước nhỏ gọn. Tất cả các thiết bị di động rồi sẽ được áp dụng công nghệ WAP, trực tiếp từ nhà sản xuất hay là từ phiên bản nâng cấp nào đó thuộc nhóm các công ty thứ ba (third-party). Mỗi một thiết bị có một cách hiển thị khác nhau và các phương pháp nhập liệu khác nhau. Công việc của công nghệ WAP là sắp xếp lại “mớ hỗn độn” đó và cung cấp một khung làm việc (framework) chung cho phép các ứng dụng chạy được trên cả tất cả hệ nền khác nhau này.

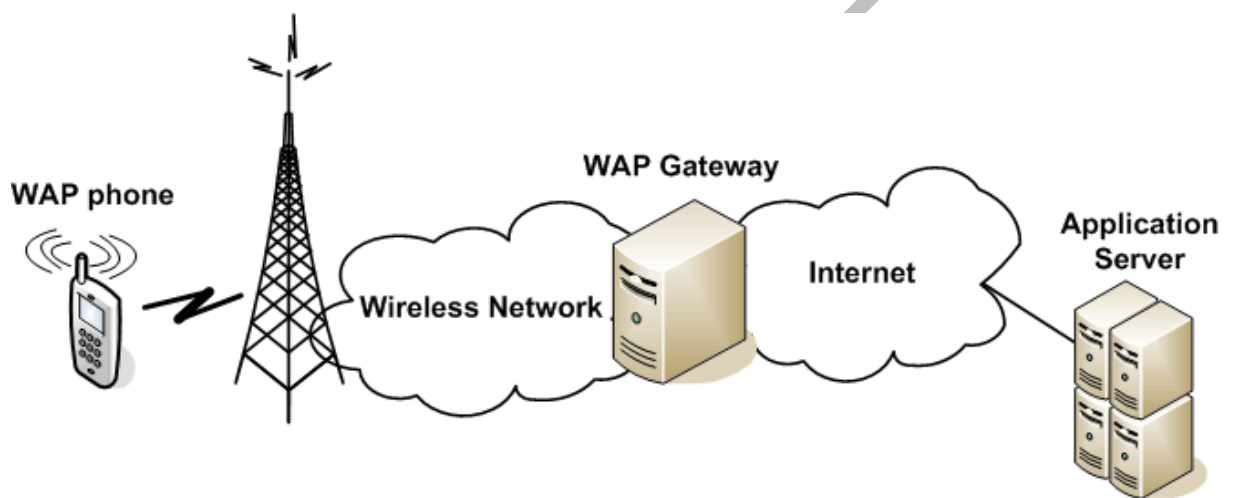
1.2. Kiến trúc ứng dụng WAP

Các giao thức WAP được thiết kế trên nền của các giao thức web. Mục đích của WAP là sử dụng lại cấu trúc cơ sở của web, để từ đó nâng cao quá trình giao tiếp giữa nhà cung cấp và các thiết bị di động, giúp quá trình này trở nên hiệu quả và tốn ít thời gian hơn là sử dụng chính các giao thức web.

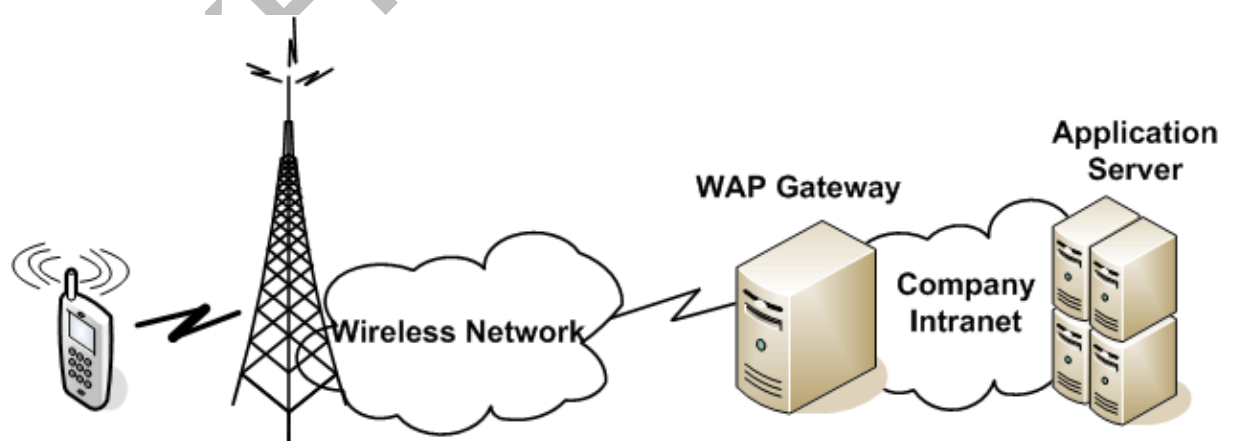
Do kiến trúc của WAP được thiết kế gần giống với Web, nên nó cũng kế thừa mô hình client-server được dùng trên Internet của Web. Điểm khác nhau chính đó là sự có mặt của WAP gateway dùng cho việc chuyển đổi giữa HTTP và WAP.



Hình 1.2-1: Kiến trúc trên Internet



Hình 1.2-2: WAP được dùng truy cập internet



Hình 1.2-3: WAP được dùng truy cập intranet

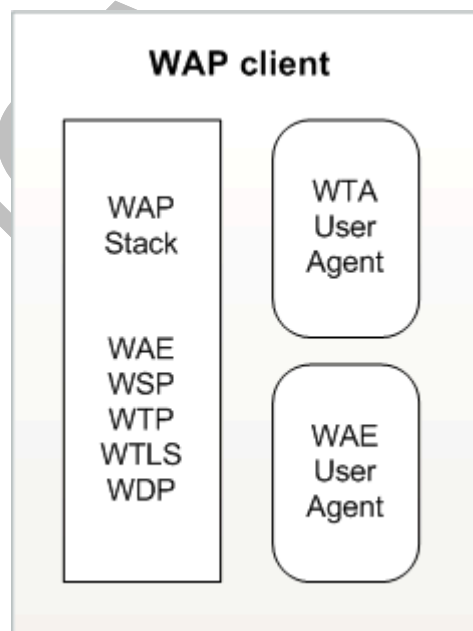


Để truy cập vào một ứng dụng trên server, client khởi tạo một nối kết với WAP gateway và gửi đi yêu cầu của mình. Gateway sẽ chuyển đổi những yêu cầu này sang định dạng được dùng trên Internet (HTTP), và sau đó chuyển chúng đến server cung cấp dịch vụ. Nội dung trả về được gửi từ server đến gateway, tại đây nó sẽ được chuyển sang định dạng WAP, để sau đó gửi về cho thiết bị di động. Như vậy, gateway đã giúp Internet có thể giao tiếp với môi trường mạng không dây [9].

1.2.1. WAP Client

Các đặc tả WAP cho phép những nhà sản xuất di động có nhiều lựa chọn cho riêng mình. Nó không bắt buộc thiết bị WAP phải trông như thế nào hay sẽ hiển thị nội dung nhận được từ Internet ra sao, mà nó gắn liền với giao diện người dùng với tổ chức bên trong của chức năng điện thoại [9].

Yêu cầu duy nhất cho một thiết bị hỗ trợ WAP đó là nó phải cung cấp một tác nhân người dùng WAE (WAE User Agent), một tác nhân người dùng WTA (WTA User Agent) và ngăn xếp WAP (WAP Stack).



Hình 1.2-4: WAP Client

- **WAE User Agent** (Wireless Application Environment User Agent) là một loại trình duyệt nhỏ (microbrowser) thực hiện hoàn trả nội dung phục vụ việc hiển thị. Nó nhận vào WML, WML Script đã được biên dịch và các hình ảnh từ WAP gateway, sau đó xử lý hoặc hiển thị chúng lên màn hình. WAE User Agent cũng quản lý việc giao tiếp với người dùng, chẳng hạn như nhập liệu văn bản, thông báo lỗi hay các thông điệp cảnh báo khác.
- **WTA User Agent** (Wireless Telephony Application User Agent) nhận các tập tin WTA được biên dịch từ WTA server và thực thi chúng. WTA User Agent bao gồm việc truy cập vào giao diện điện thoại và các chức năng mạng như quay số, trả lời cuộc gọi, tổ chức phonebook, quản lý thông điệp và các dịch vụ định vị.
- **WAP Stack** cho phép điện thoại nối kết với WAP gateway sử dụng các giao thức WAP.

Các khái niệm này sẽ được mô tả chi tiết hơn trong phần sau.

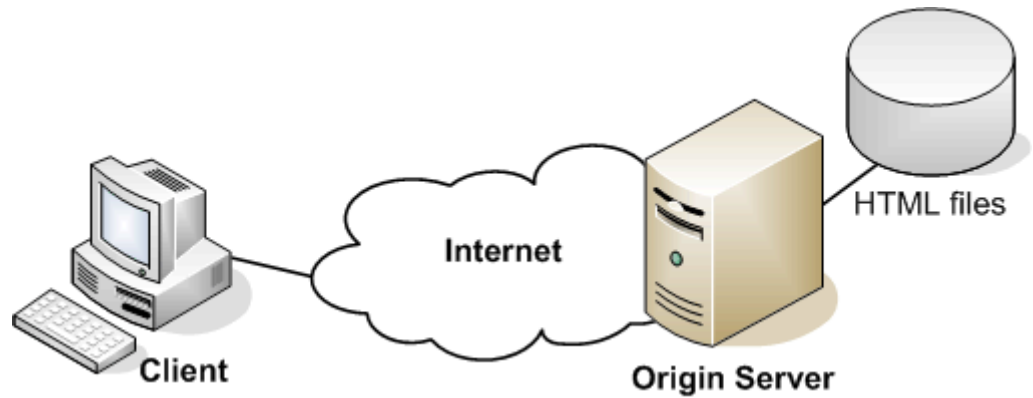
1.2.2. WAP Proxy, WAP Gateway và WAP Server

Ba thuật ngữ này rất thường được sử dụng thay thế cho nhau. Thế nhưng, trong môi trường mạng thì chúng lại hoàn toàn khác nhau với những chức năng cũng khác nhau:

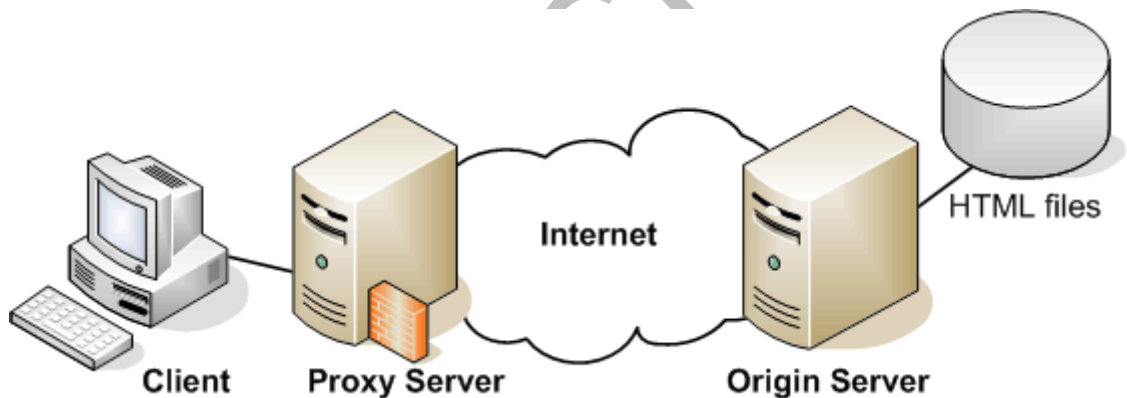
- Server nội dung/gốc/ứng dụng: đây là phần tử trong mạng chứa thông tin hoặc các ứng dụng web/WAP.
- Proxy: là một phần tử trung gian, hoạt động cả như client lẫn server trên mạng. Nó nằm giữa các client và các server gốc (origin server); các client gửi yêu cầu đến cho proxy, nó sẽ truy xuất và lưu trữ thông tin cần thiết bằng cách giao tiếp với trình duyệt gốc.
- Gateway: đây là một thành phần trung gian **thường được dùng để nối kết hai loại mạng khác nhau.** Nó nhận yêu cầu trực tiếp từ các client như thể

nó chính là một server gốc mà client muốn truy xuất thông tin. Các client này thông thường không nhận ra rằng mình đang giao tiếp với gateway.

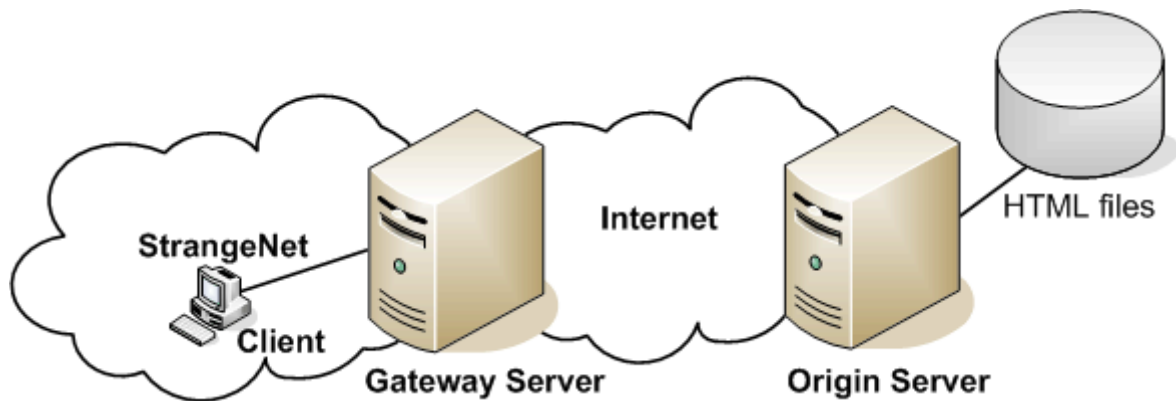
Ba thuật ngữ này được minh họa trong các hình vẽ sau đây:



Hình 1.2-5: Server gốc kết nối trực tiếp với Internet



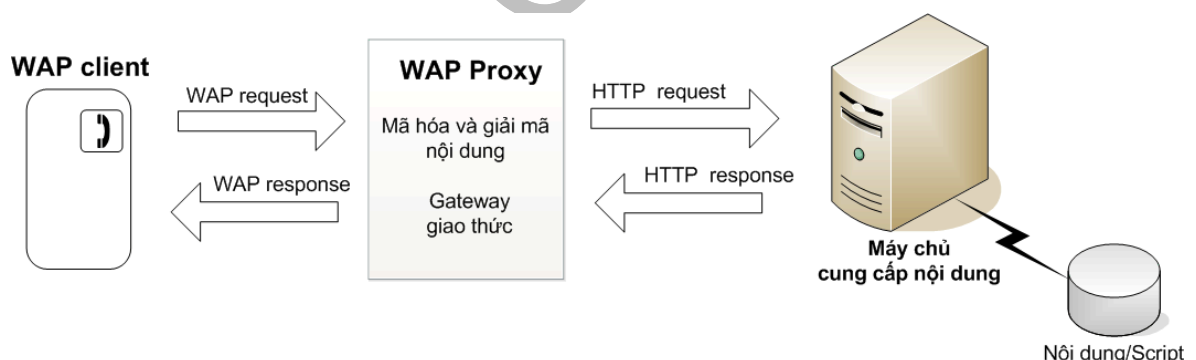
Hình 1.2-6: Truy cập Internet thông qua proxy server



Hình 1.2-7: Gateway server nằm giữa hai loại mạng khác nhau

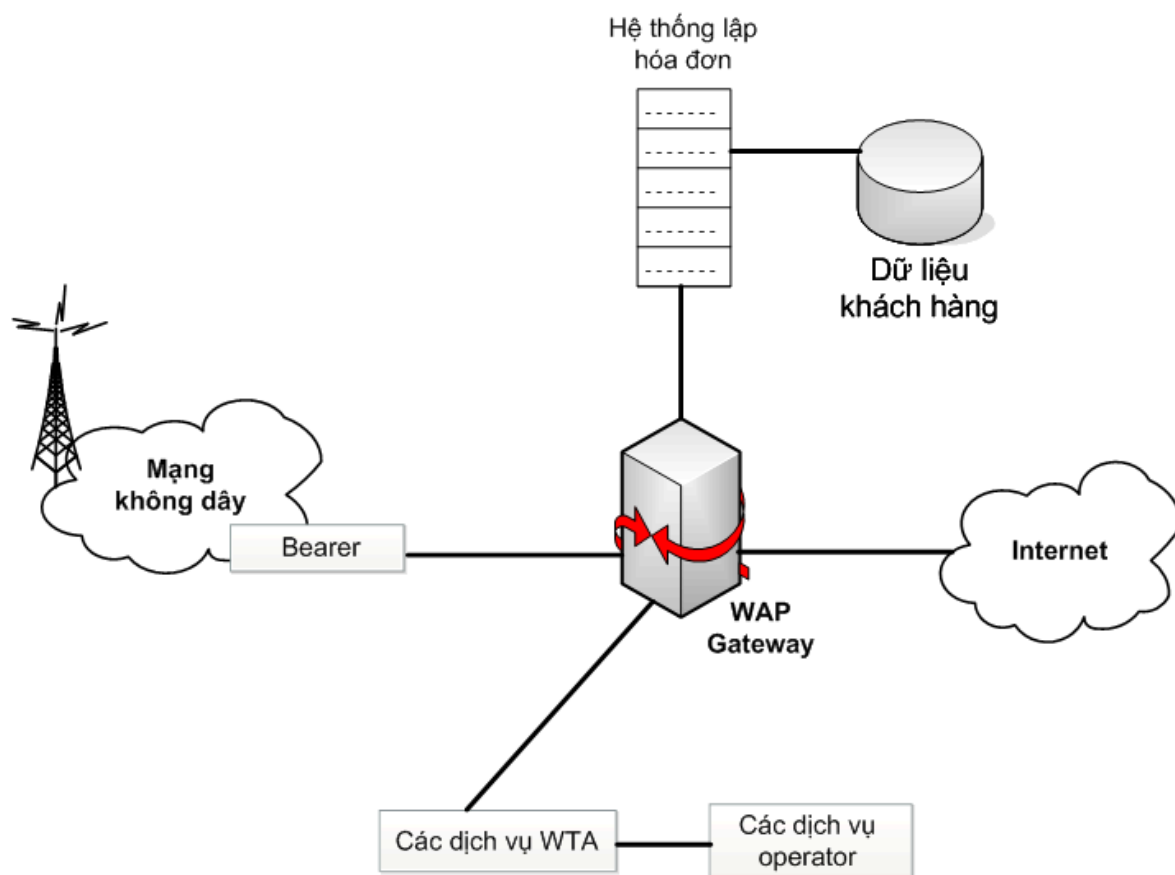
Trong kiến trúc WAP, một WAP gateway thật ra là một **proxy**. Nó được dùng để nối một vùng mạng không dây (wireless domain) với mạng Internet. Tuy nhiên, nó có thêm chức năng của *gateway chuyển đổi giao thức (protocol gateway)* và chức năng *mã hoá/giải mã*.

Hình 1.2-8 mô tả việc sử dụng một WAP proxy/gateway:



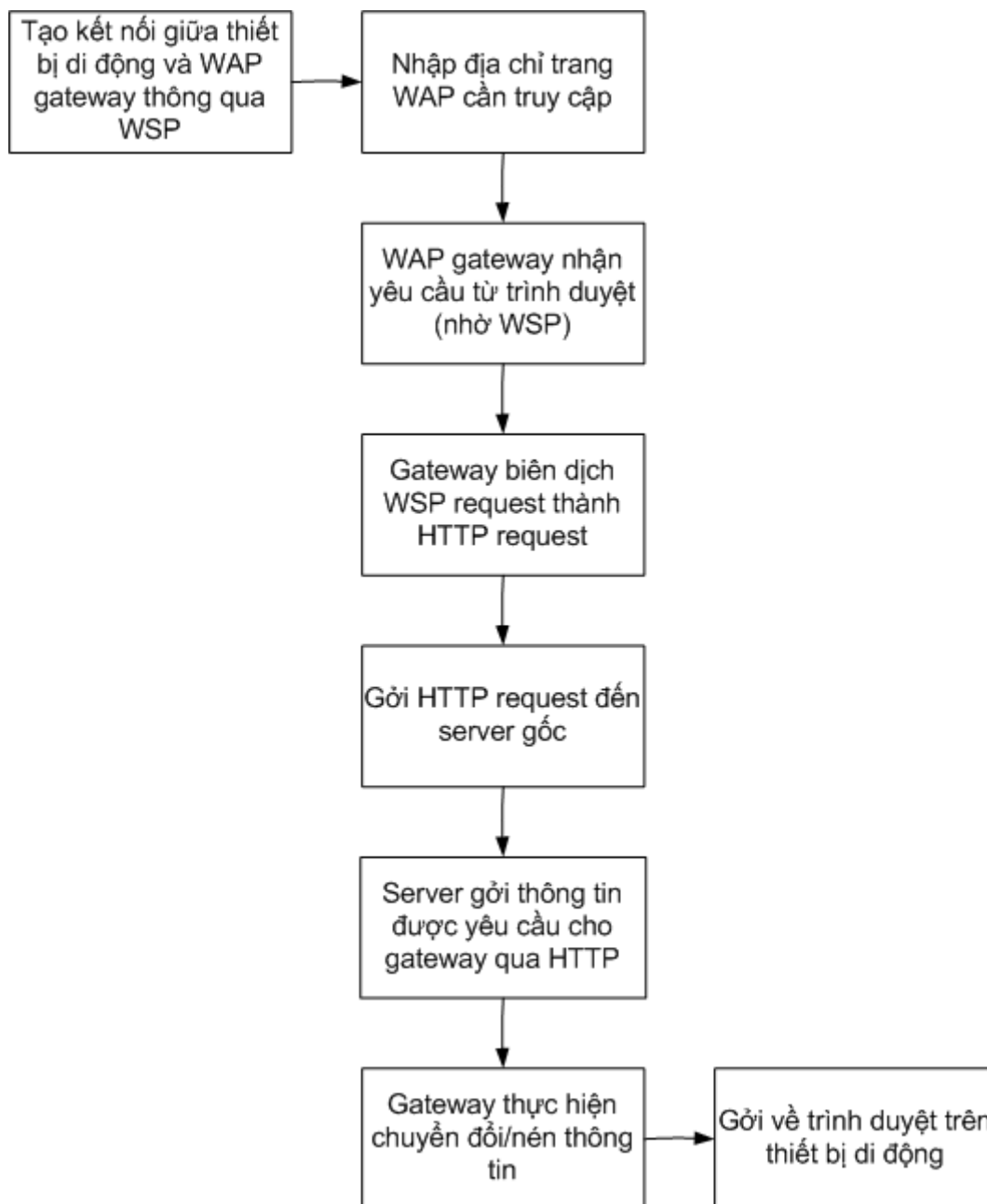
Hình 1.2-8: Sử dụng WAP proxy/gateway

Còn ở Hình 1.2-9 là một WAP gateway cùng với các phần tử khác trong mạng không dây. Nó mô tả cách mà một WAP gateway cộng tác và giao tiếp với tất cả các phần tử khác để cung cấp một dịch vụ phù hợp:

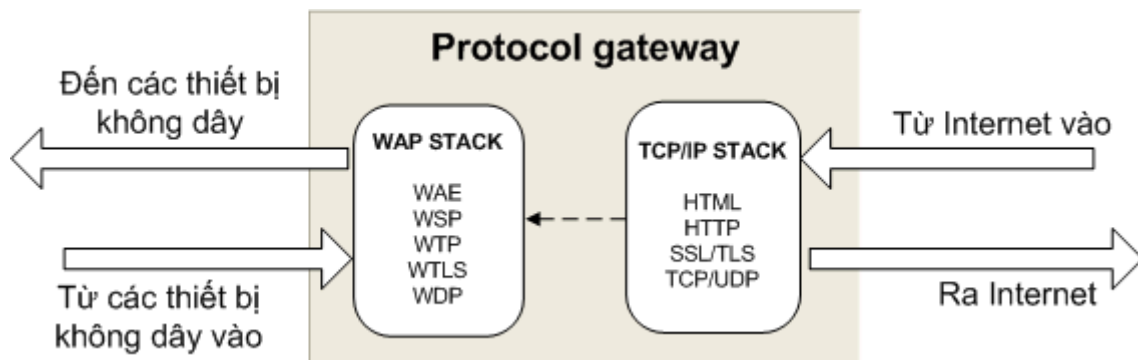


Hình 1.2-9: WAP gateway trong mạng không dây

Mỗi khi bắt đầu một phiên WAP (WAP session) trên điện thoại di động chúng ta đều phải thực hiện theo các bước như sau:

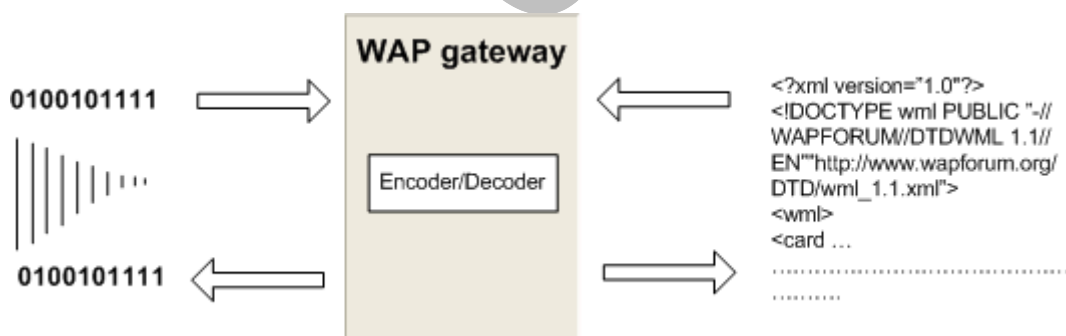


Hình 1.2-10: Các bước thực hiện khi tiến hành một phiên giao dịch WAP



Hình 1.2-10 mô tả quá trình biên dịch tại gateway chuyển đổi giao thức các yêu cầu được gửi và nhận về giữa thiết bị di động và mạng Internet.

Chức năng mã hoá/giải mã (CODEC) bên trong gateway được dùng để chuyển đổi nội dung dạng WML và WML Script thành một dạng phù hợp với các mạng có băng thông thấp (thường ở dạng nhị phân). Quá trình này được mô tả trong *Hình 1.2-11* dưới đây.

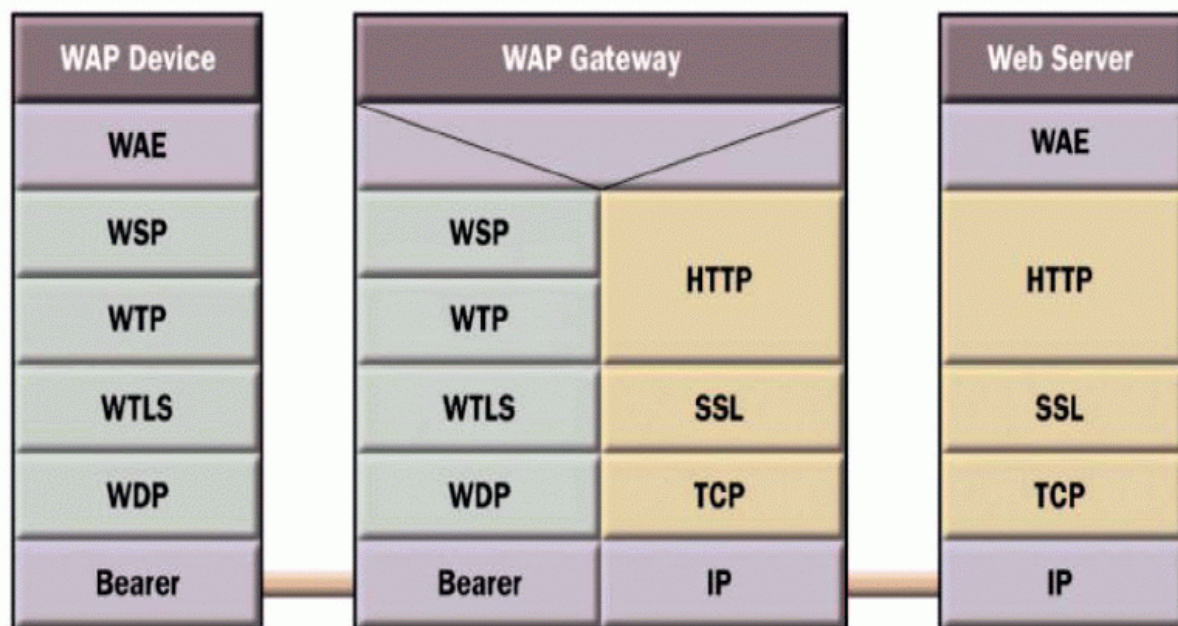


Một dịch vụ khác mà chức năng CODEC có thể cung cấp là biên dịch HTML hay văn bản thành WML/XHTML. Tuy nhiên, việc sử dụng gateway như thế này còn rất nhiều giới hạn. Mặc dù HTML và WML/XHTML đều được xây dựng dựa trên các nhưng HTML lại cho phép hiển thị các nội dung động cũng như các dạng dữ liệu đa truyền thông (multimedia) như hình ảnh, âm thanh, đồ họa, hay các cấu trúc phức tạp như các khung, các bảng lồng nhau...do đó với những giới hạn của

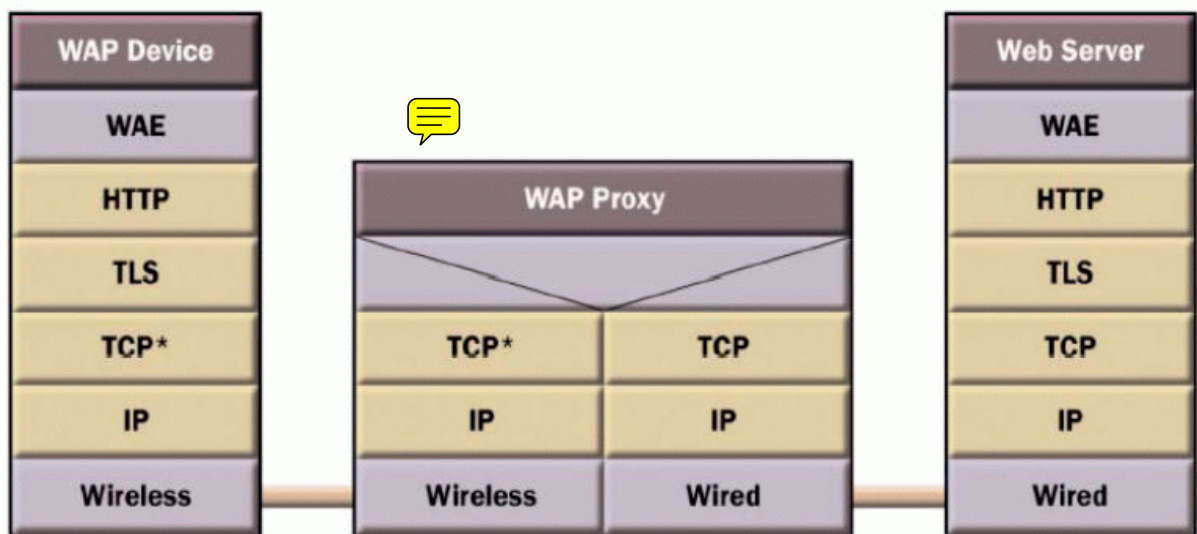
thiết bị di động (bộ nhớ nhỏ, băng thông thấp, độ trễ cao) thì việc chuyển đổi đơn thuần sẽ gây không ít khó khăn cho việc hiển thị.

1.3. Ngăn xếp giao thức WAP – WAP Protocol stack

Có hai phiên bản WAP đang được sử dụng hiện nay đó là WAP 1.x và WAP 2.0 [8].



Hình 1.3-1: Ngăn xếp WAP 1.x

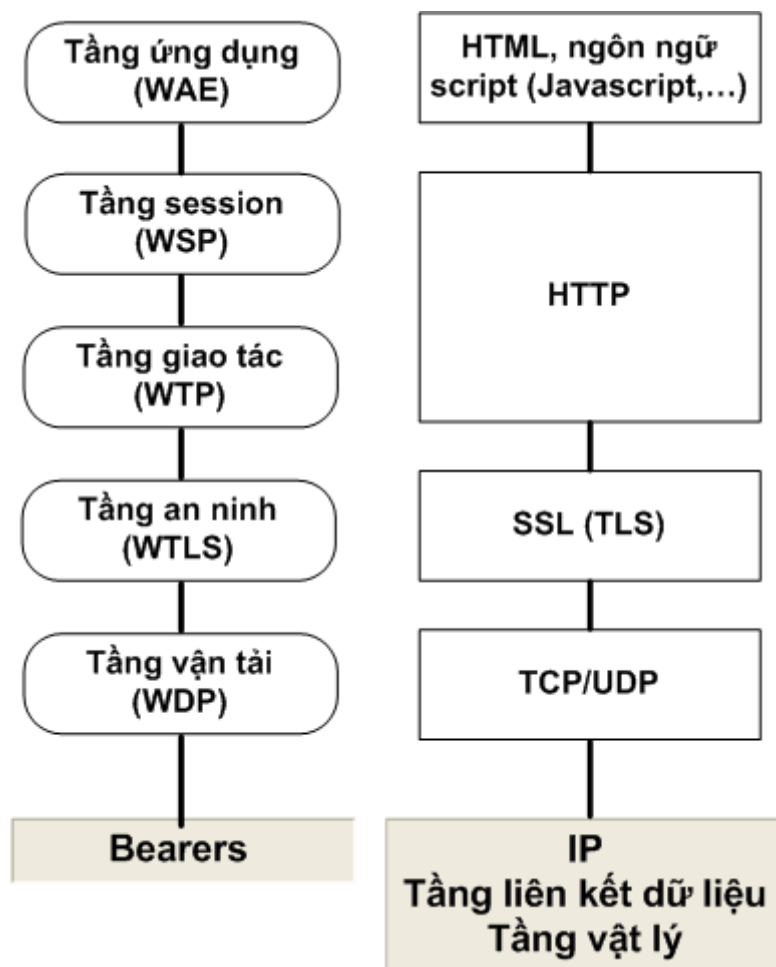


Hình 1.3-2: Ngăn xếp WAP 2.0

Ngăn xếp WAP bao gồm 5 tầng khác nhau:

- **Tầng ứng dụng (Application Layer):** WAE (Wireless Application Environment) cung cấp một môi trường ứng dụng với mục đích phát triển và thực thi các ứng dụng cũng như dịch vụ di động.
- **Tầng phiên (Session Layer):** WSP (Wireless Session Protocol) cung cấp các phương thức hỗ trợ việc trao đổi thông tin một cách có tổ chức giữa các ứng dụng dạng client/server.
- **Tầng tác vụ (Transaction Layer):** WTP (Wireless Transaction Protocol) cung cấp các phương thức khác nhau cho việc thực hiện các giao tác.
- **Tầng an ninh (Security Layer):** WTLS (Wireless Transport Security Layer) là tầng tùy chọn hỗ trợ tính bảo mật cho nối kết giữa các ứng dụng.
- **Tầng vận chuyển (Transport Layer):** WDP (Wireless Datagram Protocol) là tầng dưới cùng của ngăn xếp WAP, nó được xem là chỗ dựa cho các tầng bên trên.

Hình 1.3-3 cho thấy sự kế thừa của ngăn xếp WAP từ mô hình OSI:




Hình 1.3-3: Sự kế thừa của ngăn xếp WAP từ mô hình OSI

Cả hai tầng ứng dụng trong mô hình web và WAP đều cung cấp một ngôn ngữ đánh dấu và một ngôn ngữ điều khiển (scripting language) riêng cho việc phát triển các ứng dụng. Trong mô hình web, Tầng Phiên và Tầng Tác Vụ được kết hợp lại vào cùng một tầng đó là HTTP, trong khi chúng là hai thực thể riêng biệt trong ngăn xếp WAP (WSP và WTP). Tầng Vận Chuyển (WDP) và Tầng An Ninh (WTLS) được xếp trực tiếp lên các tầng TCP/UDP và TLS/SSL trên web.

Sau đây chúng ta sẽ xét đến từng thành phần trong ngăn xếp của WAP[9]:


1.3.1. Wireless Application Environment – WAE

Tầng ứng dụng của WAP (WAE) cung cấp tất cả các thành phần liên quan đến việc phát triển và thực thi ứng dụng. WAE cho phép những nhà phát triển có

thể sử dụng các định dạng và dịch vụ riêng biệt để tạo ra hoặc tùy biến việc hiển thị nội dung và tương tác với các thiết bị di động vốn có nhiều giới hạn. WAE gồm có hai tác nhân người dùng khác nhau được đặt ở phía client đó là: tác nhân người dùng WAE (WAE user agent) – bao gồm trình duyệt (microbrowser) và trình soạn thảo thông điệp dạng văn bản – và tác nhân người dùng WTA (WTA user agent). (WTA sẽ được đề cập chi tiết hơn trong phần 1.3). 

1.3.2. Wireless Session Layer – WSP

Wireless Session Protocol cho phép các dịch vụ trao đổi dữ liệu với các ứng dụng theo một cách có tổ chức. Nó bao gồm hai giao thức khác nhau:

- **Dịch vụ phiên hướng kết nối (Connection oriented session services)** – hoạt động nhờ vào Wireless Transaction Protocol (WTP)
- **Dịch vụ phiên phi kết nối (Connectionless session services)** – hoạt động trực tiếp trên Wireless Transport layer (WDP) 

Các **dịch vụ phiên (session service)** là những chức năng giúp cho việc thiết lập kết nối giữa một client và một server. Dịch vụ này được phân phối thông qua việc dùng các ‘primitives’ mà nó cung cấp.

Primitives là các thông điệp được định nghĩa mà một client dùng để gọi cho server yêu cầu dịch vụ. Chẳng hạn như trong WSP, một trong những primitives là SConnect, với nó chúng ta có thể yêu cầu việc tạo lập một nối kết với server.

Dịch vụ phiên **hướng kết nối** (Connection-oriented session service) cung cấp khả năng quản lý một phiên làm việc và vận chuyển dữ liệu tin cậy giữa client và server. Phiên làm việc tạo ra có thể được hoãn lại và phục hồi sau đó nếu như việc truyền tải dữ liệu không thể thực hiện được. Trong kỹ thuật push, dữ liệu không mong muốn có thể được gửi đi từ server đến client theo hai cách: được xác nhận hoặc là không được xác nhận.

- Trường hợp được xác nhận (**confirmed push**), client sẽ thông báo cho server khi nhận được dữ liệu.

- Trường hợp không được xác nhận (**unconfirmed push**) server không được thông báo khi dữ liệu push được nhận.

Phần lớn các chức năng được cung cấp bởi dịch vụ phiên hướng kết nối (connection-oriented session service) đều được xác nhận: client gửi các thông điệp yêu cầu (Request primitive) và nhận lại thông điệp xác nhận (Confirm primitive), server gửi các thông điệp phản hồi (Response primitive) và nhận lại các thông điệp chỉ dẫn (Indication primitive).

Dịch vụ phiên **phi kết nối** (Connectionless session service) chỉ cung cấp các dịch vụ không được xác nhận (non-confirmed services). Trong trường hợp này các client có thể chỉ sử dụng thông điệp yêu cầu (Request primitive) và các server cũng chỉ có thể dùng thông điệp chỉ dẫn (Indication primitive).

Để bắt đầu một phiên làm việc mới, client yêu cầu một WSP primitive cung cấp một số tham số như địa chỉ server, địa chỉ client và các client header. Các tham số này có thể được liên kết với các tiêu đề HTTP của client, và có thể được server dùng để nhận ra loại tác vụ người dùng bên trong WAP client (có thể là phiên bản và loại của trình duyệt). Điều này có ích khi ta muốn định dạng lại phần đầu ra khác đi, tùy thuộc vào loại thiết bị ở phía client. Chẳng hạn như một điện thoại có thể có một màn hình hiển thị chứa được 20 ký tự; nhưng thiết bị khác thì lại chỉ hiển thị được 16 ký tự.

WSP về cơ bản đó chính là một dạng nhị phân của HTTP. WSP cung cấp tất cả các phương thức được định nghĩa bởi HTTP/1.1 và cho phép đàm phán nhằm đạt được sự tương thích với chuẩn HTTP/1.1 này.

1.3.3. Wireless Transaction Layer – WTP

Wireless Transaction Protocol cung cấp các dịch vụ nhằm thực hiện các giao tác tin cậy và không tin cậy, nó làm việc trên tầng WDP hay tầng an ninh WTLS. Cũng như tất cả các tầng khác trong WAP, WTP được tối ưu cho phù hợp với băng

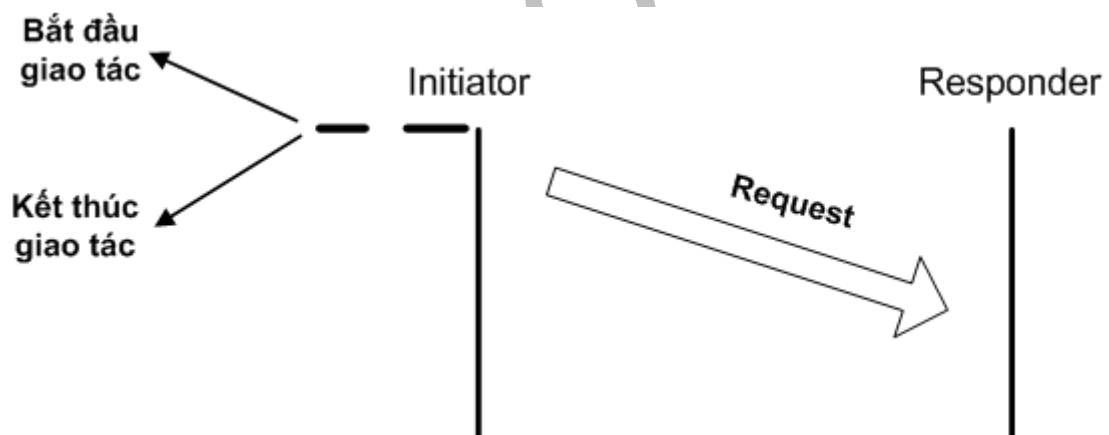
thông nhỏ của giao tiếp trên sóng vô tuyến, cố gắng giảm số lượng các giao tác thực hiện lại giữa client và server.

Cụ thể, có ba lớp khác nhau của các dịch vụ giao tác cung cấp cho các tầng bên trên là:

- Các yêu cầu không tin cậy – Unreliable requests
- Các yêu cầu có thể tin cậy – Reliable requests
- Các yêu cầu tin cậy với một thông điệp kết quả

1.3.3.1. Yêu cầu không tin cậy – Unreliable request

Trình khởi đầu (Initiator) (trong trường hợp này là một server chứa nội dung – content server) gửi yêu cầu đến trình đáp ứng (Responder) (tác nhân người dùng), và không có một thông điệp xác nhận nào được gửi trả về. Giao tác này không có trạng thái và kết thúc ngay thông điệp yêu cầu được gửi đi:

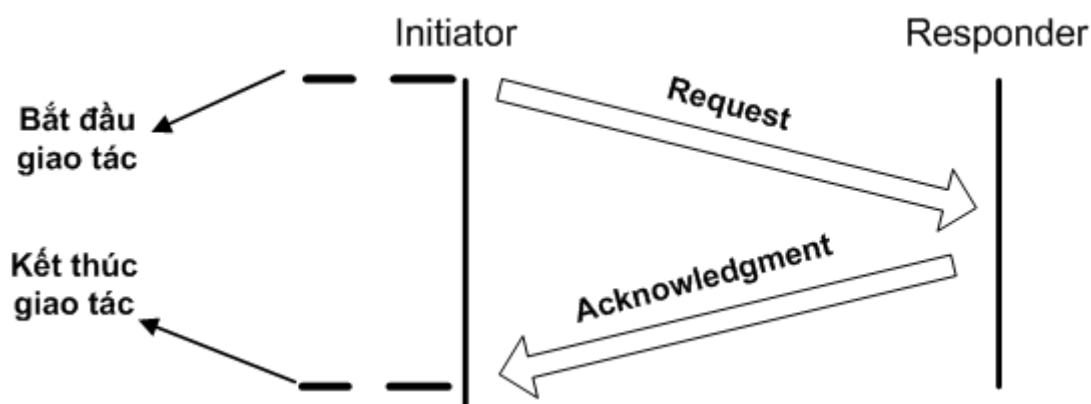


Hình 1.3-4: Unreliable request

1.3.3.2. Yêu cầu có thể tin cậy – Reliable request

Trình khởi đầu gửi một yêu cầu đến cho trình đáp ứng, trình này sẽ trả lời lại khi nhận được yêu cầu. Trình đáp ứng lưu trữ thông tin trạng thái của giao tác trong một thời gian để nó có thể gửi lại thông điệp xác nhận (acknowledgement

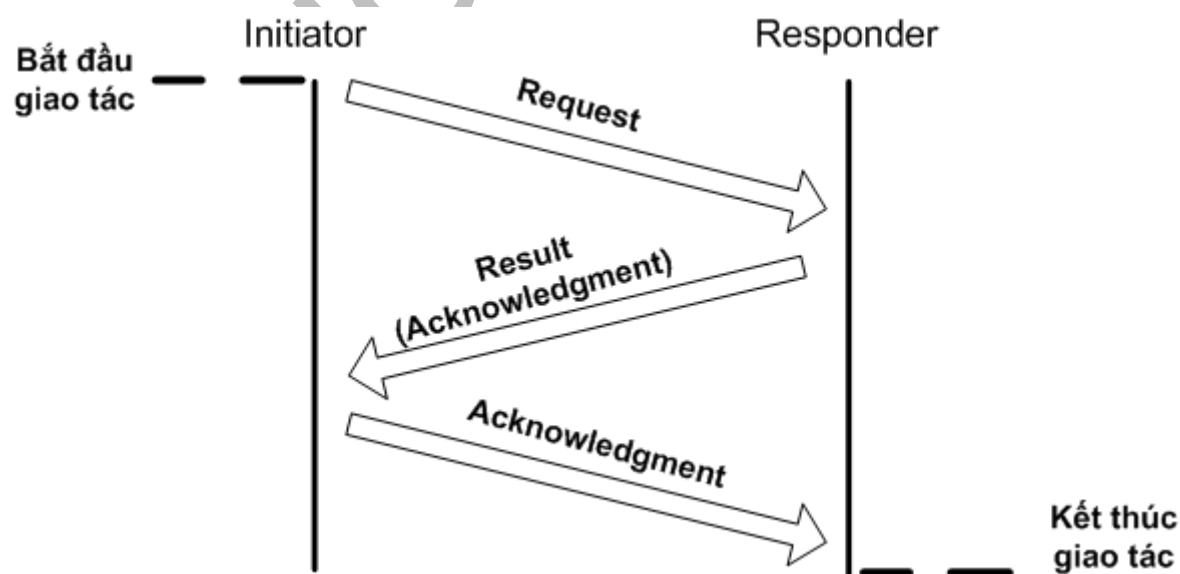
message) nếu như server có yêu cầu lại lần nữa. Giao tác kết thúc tại trình khởi đầu khi trình này nhận được thông điệp xác nhận:



Hình 1.3-5: Reliable request

1.3.3.3. Yêu cầu tin cậy với một thông điệp kết quả

Trình khởi đầu gửi yêu cầu đến cho trình đáp ứng, khi nhận được yêu cầu trình này sẽ gửi trả lại một thông điệp kết quả. Trình khởi đầu nhận thông điệp này, duy trì thông tin trạng thái của giao tác trong một thời gian sau khi xác nhận được gửi đi, phòng trường hợp thông báo gửi đi không đến được đích. Giao tác kết thúc tại trình đáp ứng khi nó nhận được thông điệp xác nhận.



Hình 1.3-6: Reliable request với thông điệp kết quả

1.3.4. Wireless Transport Layer Security – WTLS

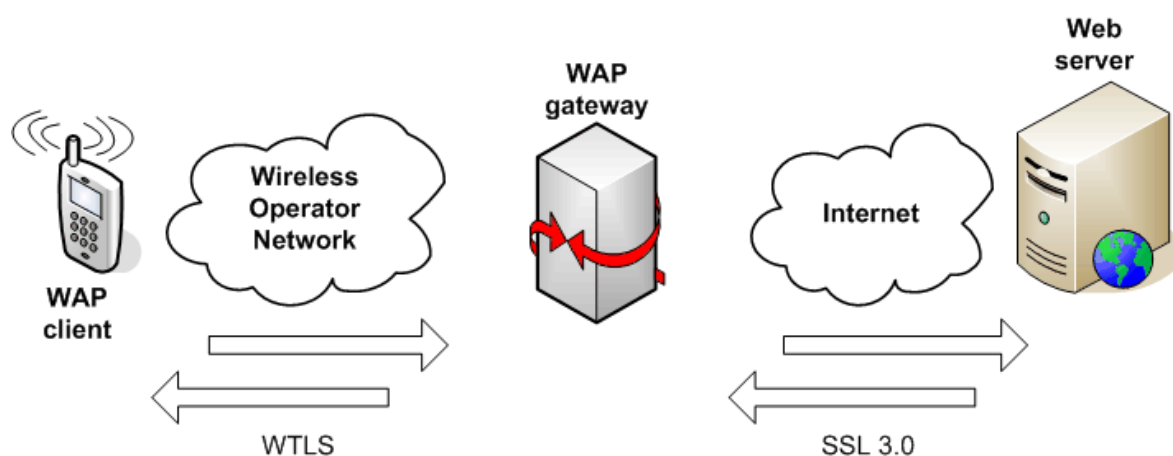
WTLS được cung cấp bởi WAP Forum, đây là một giải pháp cho vấn đề bảo mật trên WAP. WTLS là một tầng tùy chọn hoạt động trên tầng vận chuyển (WDP), và được xây dựng dựa trên hai giao thức Internet đó là TLS (Transport Layer Security) v1.0, tầng này cũng dựa trên một tầng khác đó là SSL (Secure Sockets Layer) v3.0.

WTLS cũng có cùng các đặc điểm cơ bản như tất cả các tầng trước đây trong ngăn xếp WAP: nó là điều chỉnh của một giao thức Internet cho phù hợp với điều kiện độ trễ cao, băng thông thấp, cùng với bộ nhớ và khả năng xử lý giới hạn của các thiết bị WAP. WTLS cũng cố gắng giảm bớt chi phí liên quan đến việc thiết lập một kết nối an toàn giữa hai ứng dụng. WTLS cung cấp cùng một mức độ bảo mật như ở SSL 3.0 nhưng giảm đi khoảng thời gian giao tác. Các dịch vụ mà nó cung cấp là:

- **Tính bảo mật (Privacy)** bảo đảm dữ liệu gửi đi giữa server và client không thể được truy cập từ bất kỳ người nào khác. Không ai có thể giải mã thông điệp cho dù họ có thể nhìn thấy các thông điệp này ở dạng đã được mã hóa.
- **Định danh server** đảm bảo một server thật sự.
- **Định danh client** giúp server gốc giới hạn khả năng truy cập đến những nội dung mà nó cung cấp. Xác định chỉ một số client nào đó mới có thể truy cập vào những trang nào đó cho phép mà thôi.
- **Bảo toàn dữ liệu** sẽ đảm bảo nội dung dữ liệu trên đường truyền giữa server và client sẽ không bị chỉnh sửa mà không được thông báo.

Hình 1.3-7 dưới đây mô tả cách WAP gateway điều khiển các phiên làm việc an toàn. Một phiên SSL chuẩn được mở ra giữa web server và WAP gateway và một phiên WTLS được khởi tạo giữa gateway và thiết bị di động. Nội dung mã hóa

được gửi đi thông qua nối kết này từ server đến gateway, gateway biên dịch và gửi nó đến cho điện thoại di động.



Hình 1.3-7: WAP gateway điều khiển phiên an toàn

Sau đó WTLS giao quyền lại cho giao thức SSL làm việc trên Internet.

Việc chuyển đổi giữa SSL và WTLS thực hiện bên trong bộ nhớ của WAP gateway. Điều quan trọng là các thông tin không được mã hóa sẽ không được lưu trữ bên trong gateway, vì như thế sẽ làm mất tác dụng tất cả các phương pháp bảo mật được dùng để bảo vệ dữ liệu lưu trữ với những người không được định danh.

Mặc dù các WAP gateway được cung cấp nhiều chức năng để bảo đảm ở cấp an toàn cao nhất, thế nhưng vẫn còn nhiều vấn đề liên quan đến giải pháp an toàn cho WAP.

WTLS là một tầng tùy chọn trong ngăn xếp WAP. Điều này có nghĩa là cơ chế bảo mật trong WAP chỉ có giá trị khi được yêu cầu và không được xây dựng như là một chức năng trong kiến trúc WAP. Do đó, thông tin lưu chuyển đến và đi qua WAP gateway thường không được mã hoá, trừ phi chúng ta dùng các kết nối SSL để giao tiếp giữa các server gốc và gateway.

1.3.5. Wireless Datagram Protocol – WDP

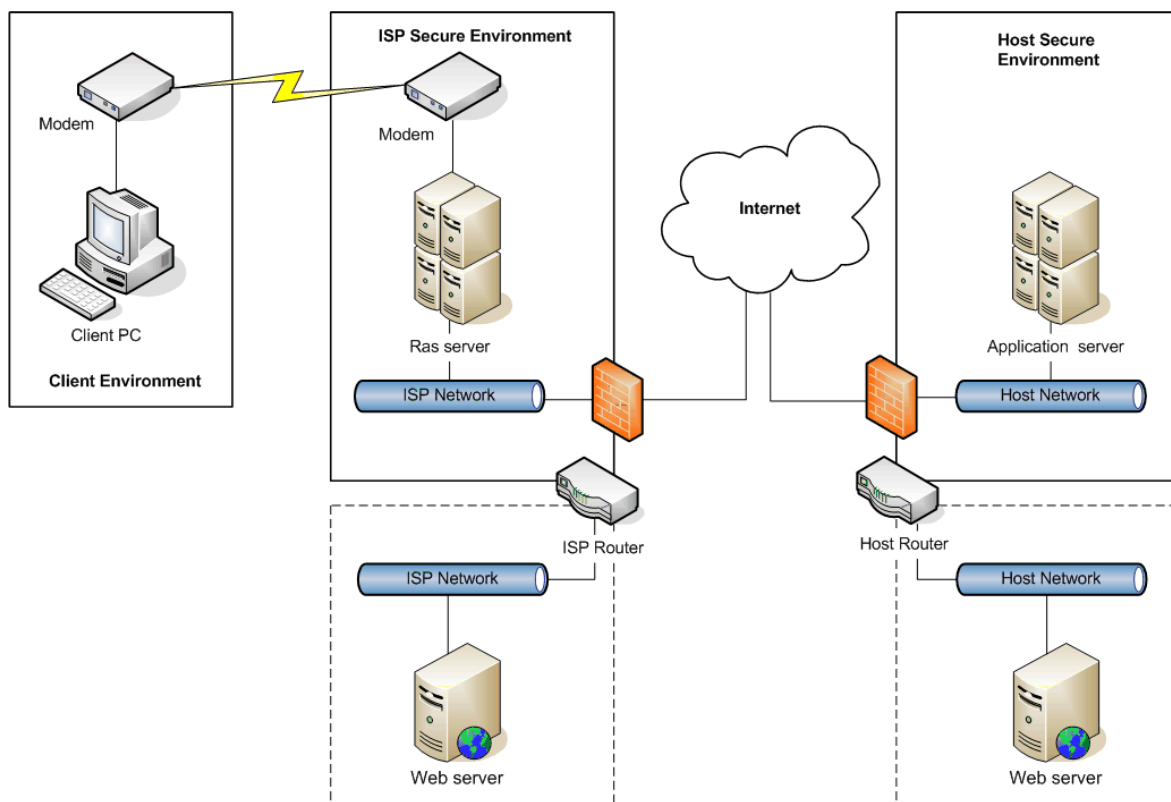
WDP là lớp dưới cùng trong ngăn xếp WAP và là một trong những phần tử làm cho WAP trở thành một giao thức cực kỳ di động, có thể thực thi trên nhiều loại mạng di động khác nhau. WDP che chở các tầng bên trên nhờ vào các dịch vụ nền mà mạng cung cấp. Các dịch vụ nền bao gồm: SMS, CSD, USSD, DECT, và CDMA.

1.4. Vấn đề bảo mật trên WAP

Bảo mật luôn là vấn đề được nhiều rất người quan tâm. Tuy nhiên, phạm vi đề tài này không cho phép thực hiện một tìm hiểu hoàn chỉnh về vấn đề bảo mật cho các ứng dụng mạng không dây. Thay vào đó là thực hiện việc so sánh vấn đề bảo mật trong môi trường WAP với một môi trường khá quen thuộc đó là Internet [9].

1.4.1. So sánh các mô hình bảo mật

1.4.1.1. Bảo mật trên Internet



Hình 1.4-1: Mô hình giao tiếp cơ điển trên Internet

Trong *Hình 1.4-1*, giả sử thiết bị ở phía client là một PC được kết nối với Internet thông qua một ISP dùng modem và giao thức PPP (point-to-point protocol).

Người dùng được ISP định danh trước khi cho phép sử dụng mạng do mình cung cấp. Các giao thức định danh này thực hiện bằng cách sử dụng tên và mật khẩu mà người dùng cung cấp.

Khi việc định danh hoàn tất, thiết bị ở phía client được cung cấp một địa chỉ IP, và đăng ký địa chỉ này với ISP. Tiếp theo, RAS server sẽ đóng vai trò như là một proxy đối với thiết bị client, thực hiện gửi đi các gói IP nhận được từ client và thu thập các gói tin gửi thẳng đến server và chuyển tiếp chúng thông qua nối kết PPP.

Mạng của ISP sẽ kết nối với mạng trục Internet (Internet backbone) thông qua một router hoặc là một gateway. Đồng thời với sự có mặt của bức tường lửa

(firewall), nó sẽ bảo vệ mạng của ISP với những lưu chuyển bên ngoài mạng Internet (bức tường lửa có thể nằm độc lập hoặc tính hợp ngay vào trong router).

Khi ra được bên ngoài Internet, dữ liệu sẽ đi qua nhiều mạng chuyển mạch (circuit-switched) và chuyển gói (packet-switched), lưu chuyển từ router này qua router khác trước khi đi đến đích.

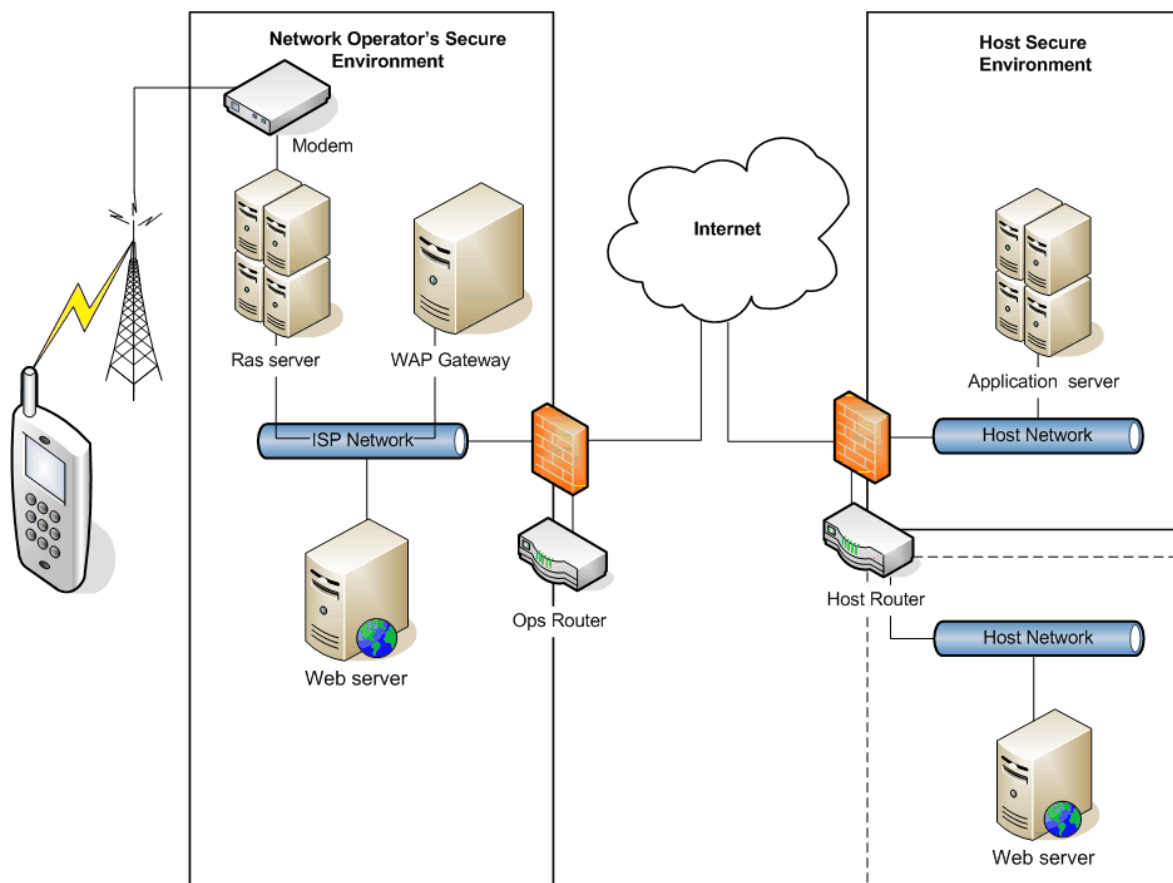
Phương thức bảo mật phổ biến nhất được dùng để bảo vệ đó là giao thức TLS (Transport Layer Security), trước đây là SSL (Secure Sockets Layer). Đây là một giao thức ở tầng vận chuyển.

Khi client yêu cầu một phiên làm việc an toàn với server, các tham số của phiên sẽ được trao đổi giữa client và server trước khi phiên làm việc an toàn được thiết lập giữa chúng. Tất cả các giao tiếp giữa client và server đều được mã hóa bằng các thuật toán và khóa được trao đổi như là một phần của việc thiết lập phiên làm việc. Mặc dù kẻ nghe trộm có thể sẽ chặn được các gói tin thiết lập phiên, nhưng với sự có mặt của khóa đủ để đảm bảo rằng phiên làm việc không bị ảnh hưởng. Điều này đạt được là do các khóa phiên được hình thành nhờ vào sự phối hợp của các khóa chung và riêng (public key, private key) lại với nhau. Như vậy, để có được khóa của phiên giao dịch, kẻ nghe trộm phải sở hữu một trong số các khóa riêng này.

TLS cung cấp các giao tiếp an toàn dạng end-to-end giữa client và server. Với hướng giao tiếp này, tất cả dữ liệu được mã hóa và không thể được giải mã bởi bất kỳ trạm trung gian nào giữa client và server.

1.4.1.2. Bảo mật trên WAP

Cũng giống như Internet, bảo mật được thực hiện ngay trên Tầng Vận Chuyển: mô hình trên mạng Internet thực thi phần lớn các chức năng bảo mật của mình trong TLS, còn WAP thì thực hiện phần lớn trong WTLS (WTLS dựa trên nền của TLS).



Hình 1.4-2: Mô hình giao tiếp trên WAP

Trong mô hình này, nối kết được thiết lập thông qua điện thoại di động, nhưng lúc này kết nối được quản lý bởi người điều khiển mạng chứ không phải từ ISP. Khi điện thoại thực hiện cuộc gọi, tín hiệu sẽ được truyền đến cho người quản lý, nó thực hiện việc tìm đường đi thông qua một trong những modem của mình và nối kết với RAS server cũng giống như trong mô hình mạng Internet.

RAS server cũng sẽ thực hiện việc định danh, nhưng một khi gói tin đi qua RAS server thì mọi thứ bắt đầu khác đi. Thay vì tìm đường trên Internet đến web server, dữ liệu được định tuyến đến WAP gateway. Tại đây, dữ liệu sẽ được biên dịch thành dạng nhị phân (nếu cần), sau đó được chuyển đi trong không khí. Gateway cũng hoạt động như là một proxy đối với điện thoại, việc giao tiếp với web server được thực hiện thông qua các giao thức HTTP 1.1. Web server không quan

tâm rằng mình đang giao tiếp với một WAP gateway, nó xem gateway đơn giản như là một thiết bị client khác.

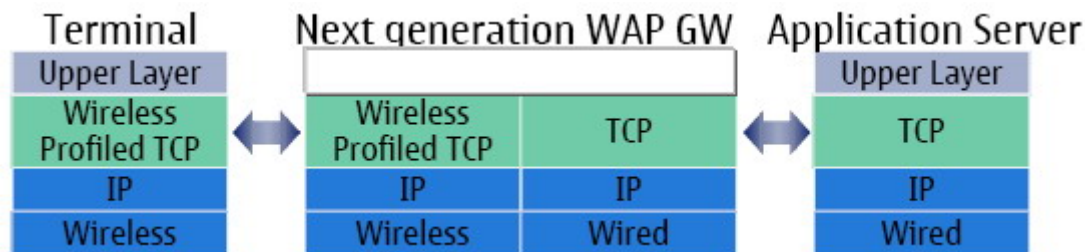
Web server có thể nằm ngay bên trong mạng, hay cũng có thể thuộc một tổ chức bên ngoài khác. WAP gateway sẽ gửi các gói tin HTTP của mình qua bức tường lửa đến với web server thuộc mạng cần đến.

Nếu như WAP gateway hoạt động như là một proxy đối với điện thoại di động và sử dụng các giao thức HTTP 1.1 thông thường thì không có lý do gì TLS không được dùng đến để đảm bảo an toàn cho tất cả các giao tiếp giữa WAP gateway và web server, giống như trên Internet. Nhưng với hai chuẩn WAP đang được áp dụng hiện nay – WAP 1.x và WAP 2.0 – thì các giao thức được dùng cho việc bảo mật khác nhau:

- WAP 1.x: do TLS đòi hỏi một truyền tải tin cậy – thường là TCP – còn điện thoại thì lại không sử dụng TCP để giao tiếp với WAP gateway nên TLS không thể dùng để bảo mật các giao tiếp giữa điện thoại di động và WAP gateway. Thay vào đó là sử dụng một giao thức mới có tên là WTLS (có khả năng hoạt động trên WDP và UDP). Giao thức này được phát triển dựa trên TLS và cung cấp cùng một mức bảo mật giống như trong TLS.

Như vậy, hệ thống phải sử dụng hai cơ chế bảo mật: một được đặt từ thiết bị đến WAP gateway, một thì từ gateway đến web server. Điều này có nghĩa là phải có một sự chuyển đổi từ WTLS sang TLS tại gateway.

- WAP 2.0: do kiến trúc của ngăn xếp WAP 2.0 gần giống với kiến trúc trên web, giao thức được sử dụng trên Tầng vận chuyển là wTCP/IP (Wireless Profile TCP/IP). wTCP/IP được tối ưu hóa từ TCP/IP nhằm vào mục đích phục vụ cho hoạt động trên môi trường di động, giao thức này có thể phối hợp tốt giữa hai môi trường mạng đó là: di động và mạng Internet [8].



Hình 1.4-3: Giao thức sử dụng trên Tầng Vận Chuyển của WAP 2.0

Do đó, cơ chế bảo mật được dùng trong WAP 2.0 cũng chính là TLS như trên môi trường web.

1.4.2. Vấn đề bảo mật trên WAP

So sánh trên đây cho thấy mô hình giao tiếp trên môi trường không dây cũng tương tự với mô hình giao tiếp trên Internet. Tuy nhiên, có hai điểm trên WAP mà ta cần quan tâm đó là: sự hiện của WAP gateway và vấn đề chứng thực người dùng [9].

1.4.2.1. Chứng thực người dùng

Khi muốn nối kết với ISP thì chúng ta cần phải cung cấp ID và mật khẩu người dùng để ISP thực hiện việc chứng thực. Hầu hết mọi người đều lưu trữ những thông tin này bên trong máy tính của mình, và chúng sẽ đại diện cho người dùng mỗi khi cần đến.

Sẽ không có vấn đề gì nếu như mỗi người có một máy tính cho riêng mình, nhưng điều gì xảy ra khi có nhiều người cùng truy cập trên cùng một chiếc máy tính? Khi đó, người sử dụng sau có thể sử dụng thông tin của người sử dụng trước đó để truy cập Internet, gửi nhận email, hay thậm chí có thể sử dụng cả những chứng nhận (certificate) của người dùng trước. Trường hợp này đòi hỏi hệ thống cần được quản lý bằng một cơ chế bảo mật nào đó.

Những vấn đề này lại nhỏ đủ có thể được bỏ qua trong môi trường có dây thông thường, trong thế giới không dây thì lại là cả một vấn đề. Có sự khác nhau rõ

ràng giữa việc chứng thực thiết bị sử dụng và chứng thực người dùng, sự khác nhau này quan trọng hơn trong trường hợp có nhiều ứng dụng.

Mặc dù vấn đề này tồn tại trên môi trường thương mại điện tử cũng như trên môi trường di động, nhưng trong môi trường di động nó lại cao hơn, đơn giản chỉ bởi vì các thiết bị này di động. Khi số lượng điện thoại di động cũng như các thiết bị di động khác tăng lên thì tỷ lệ bị mất cắp cũng sẽ tăng theo. Một số tổ chức thậm chí còn không dùng các máy laptop cho đội ngũ bán hàng của mình, vì các máy laptop rất dễ bị mất cắp và dẫn đến việc mất thông tin quan trọng có trên máy.

Bảo mật không chỉ dùng giao thức mà trong nhiều hệ điều hành còn cung cấp nhiều dạng khác, chẳng hạn như bảo mật ở cấp tập tin thông qua việc sử dụng các danh sách điều khiển truy xuất ACL (Access Control Lists). Nhưng nếu ACL được lưu trữ dưới dạng tập tin thì cũng có thể hệ thống khác sẽ đọc được nội dung này. Về bản chất đây không phải là một vấn đề của WAP, nhưng nó lại là một vấn đề về di động và cần phải được quan tâm đến nếu như các thiết bị di động chứa các thông tin quan trọng.

Một cách để tránh được trường hợp này đó là không bao giờ lưu các thông tin quan trọng trên thiết bị di động nếu có thể. Một khả năng khác là thực hiện việc chứng thực người dùng. Sử dụng cách chứng nhận sẽ định danh một cách hiệu quả các thiết bị và thiết lập một kết nối an toàn, và sau đó tất cả dữ liệu được truyền đi dưới dạng được mã hoá, yêu cầu người dùng nhập vào ID và mật khẩu. Chúng ta có thể dùng bất kỳ một kỹ thuật thông thường nào để xác nhận ID và mật khẩu này như: Kerberos, LDAP, hay một sản phẩm chứng thực người dùng nào đó.

1.4.2.2. WAP Gateway

Vấn đề trên WAP gateway có thể nhận thấy rõ ràng nhất là trên chuẩn WAP 1.x, do chuẩn này đòi hỏi WML và WMLScript phải được chuyển thành dạng nhị phân cho phù hợp với đặc điểm vận chuyển trên môi trường di động – có nhiều thách thức về băng thông và tài nguyên của thiết bị. WAP gateway chịu trách nhiệm thực hiện công việc này. Tuy nhiên:

- Một phiên bảo mật WTLS được thiết lập giữa điện thoại và WAP gateway, chứ không phải là trực tiếp với web server. Như vậy, dữ liệu chỉ được mã hoá giữa điện thoại và gateway, khi đến gateway chúng được giải mã trước khi lại được mã hóa và gửi đến cho web server qua một kết nối TLS.
- Tại WAP gateway toàn bộ dữ liệu có thể được thấy một cách tường minh. Điều này cũng có nghĩa là tại gateway dữ liệu có thể sẽ bị mất mát.

1.4.2.3. TLS và WTLS

❖ Giống nhau:

Cùng khái niệm phân biệt một phiên (session) và một kết nối (connection).

- Kết nối được đánh giá là ngắn hơn so với phiên
- Trong trường hợp mạng không dây, thời gian sống của một kết nối có thể tùy thuộc vào chất lượng thông tin nơi mà người dùng sử dụng (vị trí địa lý, khí hậu...)
- Các phiên bền hơn là các kết nối và có thể tồn tại qua nhiều kết nối và được xác định bằng một số ID của phiên (session ID)
- Các tham số bảo mật cho mỗi phiên được sử dụng để bảo mật một kết nối, có nghĩa là khi một kết nối bị phá vỡ, phiên có thể vẫn tồn tại và có thể được phục hồi sau đó.
- Phiên có thể được phục hồi, có nghĩa là một phiên đang được thiết lập có thể sử dụng cùng một tập tham số bảo mật với phiên trước đó. Phiên đó có thể là từ một kết nối hiện đang hoạt động, một kết nối khác cũng đang hoạt động hay là một kết nối đã hoạt động rồi. Việc phục hồi các phiên có thể được sử dụng để tạo nên các kết nối đồng thời cùng chia sẻ một tập tham số chung. Điều đó còn tùy thuộc vào server, vì server có quyền quyết định xem có cho phép phiên được phục hồi hay không

❖ **Khác nhau:**

TLS	WTLS
Thực chất là một tầng thêm vào Tầng Vận Chuyển dùng để can thiệp giữa tầng ứng dụng và tầng vận chuyển ‘thực sự’ nhằm vào mục đích bảo mật.	Thuộc tầng vận chuyển , nhưng bên trên nó là WTP và WSP và Tầng Phiên. Cách sắp xếp này cho phép chúng có thể độc lập với các dịch vụ được ứng dụng yêu cầu.
Đòi hỏi giao thức vận chuyển tin cậy (TCP). Không dùng trường số tuần tự (sequence number field).	Không đòi hỏi giao thức vận chuyển tin cậy (UDP, WDP). Dùng trường số tuần tự: cho phép WTLS làm việc với các vận chuyển không tin cậy.
Cho phép phân đoạn, lắp ghép dữ liệu dưới dạng các gói tin nhận được từ các tầng trên.	Không hỗ trợ phân đoạn, lắp ghép dữ liệu dưới dạng các gói tin.

Bảng 1.4-1: Một vài điểm khác nhau giữa TLS và WTLS

WTLS cho phép chứng thực cả client và server gồm ba lớp thực hiện cùng với các đánh dấu chức năng là: bắt buộc, tùy chọn hay loại trừ.

Lớp 1 chỉ yêu cầu hỗ trợ trao đổi khoá chung (public key exchange), mã hoá và MACs (kiểm soát truy cập môi trường truyền thông), các chứng nhận bên phía client và server và một tùy chọn bắt tay bí mật có chia sẻ (Một bắt tay bí mật có chia sẻ là trường hợp mà cả client và server đều đã biết được bí mật và chúng không cần trao đổi với nhau nữa).

Các thuật toán nén và giao tiếp thể thông minh không được dùng trong quá trình thực hiện của Lớp 1. Các thực thi trên Lớp 1 có thể vẫn chọn hỗ trợ cho việc

chứng thực cả hai phía client và server thông qua các chứng nhận, nhưng nó không cần thiết.

Lớp 2 hỗ trợ chứng nhận phía server là cố định.

Lớp 3, hỗ trợ cho cả client và server là cố định.

Hỗ trợ việc nén và giao tiếp thể thông minh là một tùy chọn ở Lớp 2 và 3.

Quá trình thực hiện:

Client bắt đầu tiến trình thiết lập một phiên bảo mật bằng cách gửi thông điệp đến cho server yêu cầu đàm phán thiết lập phiên bảo mật.

Server cũng có thể gửi thông điệp yêu cầu phía client bắt đầu một phiên đàm phán, thể nhưng nó còn tùy thuộc vào phía client có đồng ý hay không.

Tại bất kỳ thời điểm nào trong phiên làm việc, phía client cũng có thể gửi thông điệp này để yêu cầu đàm phán lại các thiết lập này. Đàm phán lại các thiết lập giúp giới hạn lượng dữ liệu có thể thấy được khi kẻ nghe trộm tấn công bằng cách tạo ra một khoá an toàn mới.

Khi client yêu cầu đàm phán một phiên bảo mật, nó cung cấp một danh sách các dịch vụ bảo mật mà nó có thể hỗ trợ. Phía client cũng cho biết rằng sau bao lâu thì các tham số bảo mật phải được làm mới lại. Trong phần lớn các trường hợp, phía client có thể yêu cầu các tham số này được làm mới qua mỗi thông điệp.

Nếu cơ chế trao đổi khoá chung xác định không phải là kẻ mạo danh thì phía server phải gửi cho client một chứng nhận để xác định chính mình. Chứng nhận được gửi đi phải phù hợp với thuật toán trao đổi khoá đã được đồng ý.

Chứng nhận ở phía gửi phải đến đầu tiên trong danh sách, và mỗi chứng nhận đến tiếp theo phải chứng thực chứng nhận đến trước đó. Chứng nhận của CA gốc có thể được bỏ qua trong danh sách, về cơ bản có thể chấp nhận chứng nhận của CA gốc có giá trị tùy ý, và có thể đã có sẵn ở phía client. Nếu không thì client cũng có thể dễ dàng quản lý được.

Nếu việc trao đổi khóa không là ẩn danh, thì server cũng có thể yêu cầu một chứng nhận từ phía client. Nếu client không có chứng nhận, nó có thể gửi một thông điệp mà không chứa chứng nhận nào. Tùy thuộc vào server quyết định xem có muốn tiếp tục không với một chứng nhận có giá trị từ phía client. Bằng cách gửi thông điệp này, client chứng minh nó có một khoá riêng tương ứng khóa chung chứa trong chứng nhận mà nó gửi cho server. Client gửi thông điệp chứa tất cả các thông điệp bắt tay được trao đổi trước đó giữa client và server, và được đánh dấu bằng khoá riêng của nó. Việc này cho phép server thực hiện một tính toán tương tự ở phía của nó và kiểm tra tập phân loại thông điệp mà nó nhận được như là một phần của chữ ký so với cái mà nó tạo ra. Nếu chúng phù hợp, server biết client đó là thật.

Chương 2. SỰ PHÁT TRIỂN CỦA CÁC NGÔN NGỮ ĐÁNH DẤU PHỤC VỤ CHO WAP

Tương lai mà công nghệ WAP nhắm đến là tiến gần hơn với các chuẩn hiện đang được sử dụng rộng rãi trên Internet. Và để có thể “hội tụ” các nội dung phát triển trên di động và Web truyền thống, WAP Forum đã thông qua chuẩn XHTML với Cascading Style Sheets (CSS) từ W3C như là cơ sở cho WAP 2.0. Việc chuyển đổi sang XHTML MP và WAP CSS đã củng cố thêm vị trí của các trình duyệt di động trong xu hướng phát triển Internet và mở ra khả năng xa hơn trong việc thiết kế cách thức định dạng so và trình bày với trước đây:

- Các nội dung đang tồn tại trên mạng được viết theo định dạng HTML và XHTML, do đó việc định nghĩa chuẩn WAP 2.0 làm cho việc phát triển các nội dung trao đổi trên thiết bị di động trở nên dễ dàng hơn, nhanh hơn và ít tốn kém hơn.
- XHTML cùng với CSS cho phép sắp xếp chính xác vị trí của văn bản, hình ảnh, các đường viền và những phần tử khác, do đó tất cả các thiết bị di động có thể trình bày nội dung XHTML một cách đồng nhất, tránh được nhiều vấn đề trước đây vẫn xảy ra trong WML.
- Các nhà phát triển đã có kinh nghiệm làm việc với XHTML và CSS, các công cụ phát triển Web và các máy chủ cũng đã hỗ trợ các chuẩn này.
- Sự phát triển trong tương lai của các chuẩn Web sẽ được áp dụng cho cả Internet truyền thống và trên các thiết bị di động.

2.1. Ngôn ngữ đánh dấu (Markup-Language)

SGML (Standard Generalize Markup Language) là ngôn ngữ đánh dấu đa mục đích dành cho các tài liệu xử lý trên máy tính. Nó là một tập hợp bao gồm các thẻ (tag) đánh dấu, các phần tử, và nhãn. Các thẻ đánh dấu được giới hạn bằng hai ký tự “<” và “>” để phân biệt với phần dữ liệu. Ngôn ngữ này sử dụng DTD

(Document Type Definition), nó là một dạng tài liệu có cú pháp đặc biệt, được dùng để định nghĩa các phần tử có thể sẽ xuất hiện trong một tài liệu và trật tự sắp xếp của chúng.

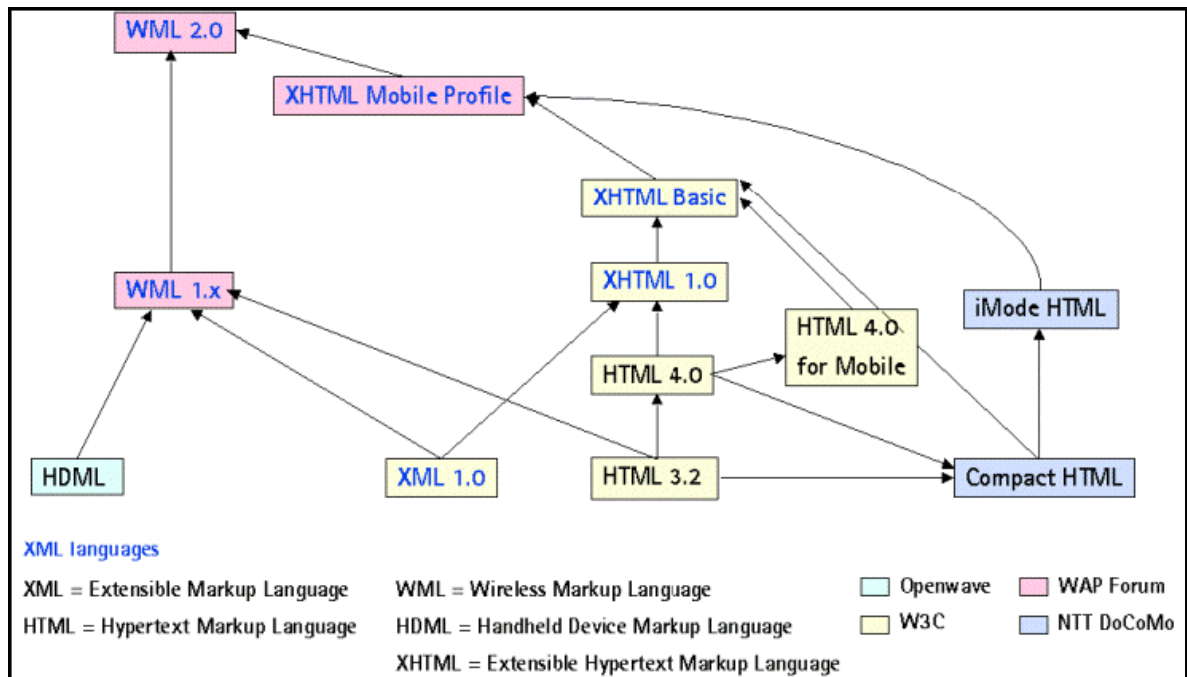
Mặc dù SGML là một ngôn ngữ đánh dấu chung nhưng ta có thể tạo nên nhiều loại tài liệu khác nhau bằng cách sử dụng những tài liệu DTD khác nhau.

HTML là một ngôn ngữ đánh dấu dựa trên SGML được dùng để tạo ra các tài liệu hiển thị trên các trình duyệt Web. Tuy được xây dựng dựa trên SGML nhưng ngôn ngữ được sử dụng rộng rãi này lại không hoàn toàn tuân theo SGML do không cần dùng DTD cũng như nó không đòi hỏi phải tuân theo các qui luật sử dụng các phần tử trong tài liệu. Ví dụ HTML cho phép sử dụng một thẻ mở (<p>) mà không cần phải có một thẻ đóng (</p>).

Do nhu cầu sử dụng ngôn ngữ đánh dấu cho những mục đích khác chứ không chỉ là hiển thị đơn thuần (như HTML), ngôn ngữ XML đã ra đời. Do XML có cấu trúc rất nghiêm ngặt và rất uyển chuyển, nên có thể được sử dụng cho nhiều mục đích khác như: thương mại điện tử, xử lý cơ sở dữ liệu v.v..

XHTML đại diện cho sự cách mạng của HTML, nó gần như giống với HTML nhưng lại tuân theo cú pháp nghiêm ngặt của XML. XHTML được sử dụng để trình bày nội dung trên các trình duyệt Web giống như HTML đã làm.

WML là một ngôn ngữ XML dùng một DTD riêng của nó. WML được sử dụng để soạn thảo các nội dung được sử dụng để hiển thị trên các thiết bị di động.



Hình 2.1-1: Mối quan hệ giữa các ngôn ngữ đánh dấu

2.2. WAP và WML

Vào năm 1998, WAP Forum đã cho ra đời một ngôn ngữ mới gọi là WML 1.1 và một giao thức mạng mới được gọi là WPS-Wireless Protocol Stack. WML 1.1 sau đó được nâng cấp lên thành WML 1.3, thế hệ WAP này được đề cập đến dưới tên WAP 1.x và ngôn ngữ để xây dựng là WML 1.x.

Hiện nay, WAP 1.x vẫn hoạt động tương đối tốt nhưng khả năng áp dụng WAP vào thương mại đã xuất hiện nhiều hạn chế. Đó là những giới hạn trong khả năng điều khiển việc sắp xếp các phần tử, kết quả là tạo nên các trang tương tự như các trang web thế hệ đầu tiên – không có màu sắc và hầu hết các phần tử được canh theo lề trái và cứ thế trải dài đến hết trang. Các trang này không những thiếu sự lôi cuốn trực quan so với trang web truyền thống mà còn khó sử dụng .

Nhưng vấn đề nghiêm trọng hơn với WML đó là bảng đặc tả không định nghĩa các qui định cho việc hiển thị các phần tử trên một trang nội dung. Do đó, với cùng một nội dung WML, mỗi nhà sản xuất trình duyệt trên di động sẽ hiển thị theo

một cách khác nhau mặc dù các nội dung đó phù hợp với các đặc tả của WML. Điều này đã gây nên nhiều vấn đề nghiêm trọng nhanh chóng lan rộng trong cộng đồng các nhà phát triển web cho di động, những nhà phát triển cần phải dò ra loại thiết bị đang xử lý nội dung để có thể biến đổi kết quả để phù hợp với các cách thức xử lý WML khác nhau trên các loại thiết bị di động. XHTML MP và WAP CSS đã đưa ra những phát triển to lớn trong việc điều khiển dàn trang và khắc phục được tính không nhất quán trong việc triển khai WML.

2.3. XHTML cơ sở

Theo bảng đặc tả XHTML của tổ chức W3C (World Wide Web Consortium), XHTML Basic DTD định nghĩa một loại tài liệu đủ mạnh để có thể xây dựng và định dạng các nội dung được chia sẻ giữa các nhóm thiết bị khác nhau như máy tính để bàn, thiết bị trợ giúp cá nhân (PDA), TV và điện thoại cầm tay. XHTML Basic là một phiên bản trên di động của XHTML 1.0 bao gồm tất cả các phần tử của XHTML 1.0 ngoại trừ các phần tử không phù hợp cho các thiết bị có màn hình nhỏ, ví dụ như khung (frame).. XHTML Basic lại tiếp tục được phát triển lên thành XHTML MP bằng việc thêm vào bảy phần tử và thuộc tính mới.

Theo tổ chức W3C, bản thân XHTML là đại diện cho sự thay đổi lớn đầu tiên của HTML kể từ phiên bản HTML 4.0 được công bố vào năm 1997. XHTML đã mang lại một cấu trúc rõ ràng cho các trang web, điều này cũng đặc biệt quan trọng cho các thiết bị di động vốn có giới hạn về màn hình hiển thị và khả năng xử lý. Tổ chức W3C hiện đang khuyến khích sử dụng XHTML cho sự phát triển các trang web dành cho các trình duyệt trên máy tính để bàn cũng như các thiết bị khác bao gồm cả điện thoại cầm tay.

Sử dụng XHTML đảm bảo rằng cùng một nội dung sẽ được hiển thị như nhau trên các hệ nền khác nhau, từ các nhà cung cấp khác nhau. Điều này không giống với trường hợp sử dụng WML 1.x, vì nó không đảm bảo tính nhất quán khi hiển thị nội dung trên các loại thiết bị khác nhau. Khi đó các nhà phát triển và quản

lý nội dung phải đối diện với nhiều khó khăn do sự khác nhau của giao diện người dùng trên các thiết bị cầm tay khác nhau.

Một số thuận lợi của XHTML so với HTML bao gồm:

- Các trang nội dung có thể được kiểm tra tính đúng đắn về cấu trúc một cách dễ dàng hơn.
- Các tài liệu có thể được chuyển đổi định dạng thông qua các công cụ như XSLT (Extensible Stylesheet Language Transformation)
- Các phần trong tài liệu có thể được tìm lại một cách nhanh chóng.
- Văn bản có thể được lưu trữ một cách hiệu quả trong cơ sở dữ liệu hướng đối tượng.

2.4. XHTML Mobile Profile

Tổ chức OMA đã lấy đặc tả XHTML Basic [XHTML] DTD từ tổ chức W3C và chuyên môn hóa nó cho mục đích sử dụng cho các trình duyệt di động bằng việc thêm vào các phần tử sau: `<hr>`, ``, `<i>`, `<big>`, `<small>`, `<style>` cũng như việc cho phép sử dụng thuộc tính định dạng trong các phần tử khác.

Với XHTML MP, một tài liệu có thể được hiển thị trên một số lượng lớn nhất các trình duyệt web bao gồm cả các điện thoại cầm tay với khả năng hiển thị khác nhau. Và bởi vì XHTML MP được phê chuẩn bởi cả W3C và OMA nên nó là một tập đầy đủ các chuẩn dành cho Internet và đảm bảo nó sẽ nhận được sự chấp nhận rộng rãi trong thời gian hiện tại, cũng như sẽ được tiếp tục phát triển trong một thời gian dài sắp tới.

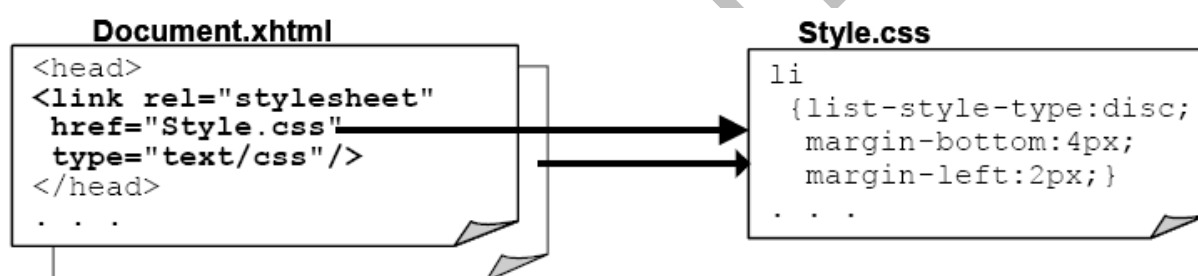
2.5. WAP CSS

Trung tâm của XHTML MP là sự hỗ trợ cho chuẩn CSS (Cascading Style Sheet) - CSS [CSS1] mô tả cách mà các tài liệu được hiển thị trên màn hình của trình duyệt. W3C đã chủ động xúc tiến việc sử dụng CSS trên Web dùng cho tất cả các trình duyệt trên máy để bàn và thiết bị di động. Thông qua việc sử dụng CSS,

những người phát triển có thể điều khiển cách trình bày của các tài liệu mà không phải phụ thuộc vào loại thiết bị hoặc thêm thẻ đánh dấu mới như đã làm trong WML 1.x.

WAP CSS là một bộ phận của CSS tương ứng cho các thiết bị nhỏ được định nghĩa bởi OMA; nó bỏ bớt các chi tiết không phù hợp với các thiết bị nhỏ và thêm vào một vài chi tiết phục vụ cho WAP.

Bằng cách sử dụng CSS, tác giả trang nội dung có thể xác định việc trình diễn toàn bộ ứng dụng Web tại một chỗ: bảng định dạng (style sheet). Nếu muốn thay đổi cách trình bày đó, ta có thể thực hiện thay đổi bảng định dạng và việc sửa đổi sẽ được áp dụng ngay lập tức trên tất cả các trang có tham khảo đến bảng định dạng đó trong hệ thống.



Hình 2.5-1: Minh họa tham chiếu bảng định dạng bên ngoài tài liệu

CSS tách biệt nội dung của tài liệu hoặc ứng dụng với việc trình bày. Điều này cho phép ta dễ dàng xây dựng được nhiều phiên bản tương ứng với các trình duyệt trên cùng một nội dung bằng cách tạo ra các bảng định dạng phù hợp. Ví dụ, khi một người dùng yêu cầu một tài liệu định dạng, Web server có thể nhận dạng được loại thiết bị gửi yêu cầu và gửi trả về bảng phiên bản định dạng phù hợp. Bảng định dạng sẽ được trình duyệt tải về một lần và được lưu lại (cache) để sử dụng cho các trang sau đó, điều này sẽ giúp cho nâng cao tốc độ xử lý tất cả các trang nội dung trong website.

Sức mạnh của CSS nằm ở khả năng điều khiển chính xác mà nó cung cấp cho những người tạo lập tài liệu và khả năng thay đổi sự hiển thị trên bất kỳ thiết bị

nào. Mỗi đặc điểm trình bày của một tài liệu – như định vị, font chữ, thuộc tính của văn bản, đường viền, canh lề, dàn trang – có thể được định nghĩa trong bảng định dạng. Khi cần thay đổi một đặc điểm nào đó trên toàn bộ website ta chỉ cần thực hiện một lần duy nhất.

2.6. So sánh XHTML với HTML, WML

2.6.1. Sự khác nhau giữa XHTML và HTML

XHTML tuân theo một cách nghiêm ngặt cú pháp của ngôn ngữ XML, trong khi HTML thì không. HTML không đòi hỏi phải bám chặt vào các tập luật của ngôn ngữ XML, do đó có thể việc hiển thị nội dung không giống nhau trên một số trình duyệt. Sự tuân thủ nghiêm ngặt các quy luật viết mã của XML đảm bảo một thể hiện thống nhất về nội dung trên phần lớn các trình duyệt và thiết bị. Nó cũng đảm bảo tất cả các tài liệu đều đúng định dạng và đúng theo XML, vì thế ta có thể sử dụng các công cụ XML chuẩn để quản lý, chuyển đổi, thao tác, và phân tích chúng.

HTML	XHTML MP
<pre> <I>item one</i> item two
 a link</pre>	<pre> <li class="i"> item one item two
 a link</pre>

Bảng 2.6-1: Các qui tắc XML có trong XHTML nhưng không có trong HTML

Minh họa trên đây sử dụng CSS thuần, không các phần tử `` và `<i>` lỗi thời (và thừa nhận các lớp CSS `i` và `b` được định nghĩa trong mẫu (style sheet) để áp dụng các thuộc tính CSS `font-style: italic` và `font-weight: bold`, theo thứ tự định sẵn). Trong thực tế, XHTML MP bao gồm các phần tử ``, `<i>`, `<big>`, và `<small>` giúp cho việc định dạng chữ tiện lợi.

Lưu ý những điểm khác nhau sau đây giữa HTML và XHTML MP [12]:

- HTML không phân biệt chữ hoa, chữ thường trong các thẻ tên (tag name). XHTML thì lại phân biệt dạng chữ trong tên tất cả các phần tử và thuộc tính. Hiện nay, tên của tất cả các phần tử trong XHTML MP là chữ thường.
- Trong cột HTML, thẻ `` đầu tiên thiếu thẻ đóng `` của nó. Tất cả các phần tử cần phải được đóng lại trong XHTML.
- Các thẻ `` và `<i>` không được lồng nhau trong mã HTML, việc lồng nhau này lại không được dùng trong XHTML MP.
- Dấu nháy không bao quanh giá trị của thuộc tính `href` trong mã HTML. Giá trị các thuộc tính trong XHTML luôn luôn phải nằm trong dấu nháy đơn hoặc kép.
- Tất cả các thẻ rỗng trong XHTML phải được “seft-closing” (tự đóng); ví dụ, `
` có một dấu / đóng bên trong thẻ để chỉ ra rằng sẽ không có thẻ đóng `</br>` tương ứng. Nó cũng được áp dụng ở nhiều thẻ khác như `<hr/>`, `<input/>`,...
- Các cặp giá trị thuộc tính không thể bị giảm thiểu như chúng có thể trong HTML. Trong HTML, khi một thuộc tính có thể mang chỉ một giá trị, nó được phép bỏ qua giá trị.

Ví dụ bên trong phần tử HTML <form> ở đoạn mã sau:

```
<option value="somevalue" selected>
```

thuộc tính "selected" vừa được cực tiểu hoá.

Trong ví dụ trên, trình duyệt HTML cho rằng phần tùy chọn nên được hiển thị là "selected". Tuy nhiên, XHTML lại yêu cầu phải cung cấp cả tên thuộc tính và giá trị của nó, ngay cả khi chỉ có một giá trị được cho phép:

```
<option value="somevalue" selected="selected">
```

Luật này cũng được dùng cho các phần tử khác. Ví dụ:

```
<input type="radio" ... checked="checked" />
```

```
<input type="checkbox" ... checked="checked" />
```

```
<dl compact="compact">
```

Các trình duyệt HTML có thể xử lý được những sự mơ hồ, không rõ ràng với những tài liệu không đúng theo chuẩn HTML. Tuy nhiên mọi thứ trong XHTML đều phải được rõ ràng vì nó được tuân theo các luật cú pháp nghiêm ngặt của XML.

Các trường nhập văn bản hỗ trợ thuộc tính CSS `-wap-input-format` để định nghĩa kiểu nhập, ví dụ *N cho kiểu nhập số. Điều này sẽ giúp người dùng tránh việc chuyển đổi qua lại giữa các kiểu nhập.

2.6.2. Sự khác nhau giữa XHTML và WML 1.x

Sự khác biệt giữa XHTML MP và WML 1.3 bao gồm những điểm sau đây[12]:

- Các không gian tên (namespace) của XML không được sử dụng trong WML 1.3, nhưng nó lại được dùng trong XHTML. Namespace mặc định được dùng là XHTML.

- Trong XHTML MP, phần tử root dạng WML <wml> đã được loại bỏ. Thay vào đó là phần tử của XHTML là <html> luôn được dùng như là root.
- Trong các mô-đun XHTML thì mọi thứ từ XHTML Basic, phần tử <hr>, thuộc tính <style>, các phần tử , <big>, <i>, và <small> được dùng đến.
- Phần tử <head> được dùng bắt buộc trong XHTML MP, thế nhưng nó lại là tùy chọn trong WML 1.3.
- Phần tử WML <element> không được dùng đến trong XHTML MP.
- Phần tử WML <card> cũng không được sử dụng trong XHTML MP, mà thay vào đó là <body>. Chỉ có thể có duy nhất một phần tử <body> trong mỗi file, không giống như <card> trong WML 1.x.3

Các đoạn neo (fragment anchor) được sử dụng để liên kết đến một phần khác thuộc cùng một trang, sau đó người sử dụng có thể quay trở về vị trí ban đầu của mình bằng cách sử dụng thanh cuộn trên trang. Ví dụ như đoạn code sau đây:

```
<a href="#h">Help</a>
...
<a name="h">Help text</a>
```

Liên kết đầu tiên sẽ xuất hiện là Help và khi nó được chọn sẽ ngay lập tức sẽ chuyển đến phần trang với anchor có tên “h”.

Thuộc tính WML order không được dùng trong XHTML MP.

Các sự kiện trong WML không được hỗ trợ trong XHTML MP, ví dụ như: <do>, <ontimer>, <onenterforward>, <onenterbackward>, <onevent>, <onpick>, <go>, <pre>, <noop> và <postfield>.

Phần tử WML <do> không được sử dụng trong XHTML MP.

Trình duyệt Nokia Mobile Browser có một điểm đặc biệt là thường đặt các liên kết quan trọng vào trong phần menu Options của điện thoại (các nhà sản xuất khác nhau có thể sử dụng những hệ thống menu khác tương ứng của mình). Đối với Nokia Mobile Browser, tất cả các liên kết và các nút Submit đều chứa thuộc tính `accesskey`, thuộc tính này được tự động thêm vào menu Option. Nhân dùng trong menu Options là nội dung những đề mục của phần tử `<a>` (hay nếu nó chỉ chứa một hình ảnh, khi đó sẽ là giá trị thuộc tính `alt` của `img`, hay là `value` của nút Submit. Phần chữ sẽ bị bỏ đi nếu như cần phải điều chỉnh lại menu cho vừa.

Đặc điểm này bảo đảm rằng người luôn có thể sử dụng các nối kết một cách nhanh chóng, mà không cần quan tâm đến là chúng nằm ở đâu trong trang hiện hành. Đặc điểm hay tương đương (và thậm chí còn đơn giản hơn) phần tử `<do>` trong WML 1.x.

Sẽ là hợp lệ trong WML khi các phần tử `<table>` nằm bên trong các đoạn văn. Thế nhưng điều này lại là không hợp lệ trong chuẩn XHTML. Tuy nhiên, nhằm để hỗ trợ cho sự chuyển đổi giữa từ WML 1.3 sang WML 2.0 các bảng nằm bên trong đoạn văn được cho phép dùng trong XHTML MP, và Nokia Mobile Browser có hỗ trợ tính năng này.

Phần tử `<pre>` có thể bao gồm các phần tử điều khiển trên form như `<input>` và `<select>`. Các phần tử này là không hợp lệ trong XHTML. Tuy nhiên, nhằm để biến đổi từ WML 1.3 sang XHTML MP, các phần tử bên trong `<pre>` đều được cho phép trong XHTML MP, và Nokia Mobile Browser hỗ trợ.

Các phần tử nhập liệu không được hỗ trợ các thuộc tính format của WML (dùng để việc nhập ký tự). Thay vào đó là sử dụng một đặc tính mới là `CSS -wap-input-format`, đặc tính này có cùng cú pháp với thuộc tính format của WML.

Sẽ luôn là một ý kiến hay khi xử lý các giá trị form trên server, và nếu như có lỗi xảy ra, quay trở lại form với các giá trị được chấp nhận và một thông điệp nhằm để hiệu chỉnh lỗi.

Với XHTML MP, thuộc tính accesskey được hỗ trợ trong các phần tử <a>, <input>, <label> và <textarea>. Tuy nhiên, cũng phải lưu ý là accesskey có thể không được hỗ trợ trong tất cả các thiết bị.

Phần tử định dạng <u> của WML không được hỗ trợ trong XHTML MP. Một cách được đề nghị để thực hiện được điều này trong XHTML MP là dùng một lớp CSS, bằng cách định nghĩa theo cách như sau trong mẫu ngoài (external style sheet):

```
.u {text-decoration:underline}
```

Sau đó sử dụng phần tử span với một lớp để sử dụng kiểu này; ví dụ:

```
<span class="u">This is underlined</span>
```

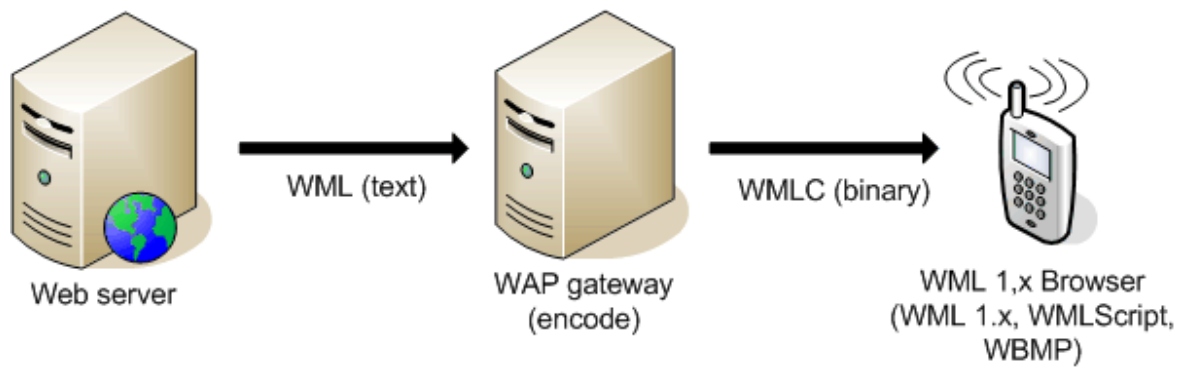
Các phần tử WML 1.x khác không được hỗ trợ trong XHTML MP. Diễn đàn về WAP đã định nghĩa một nhánh mở rộng dành cho XHTML MP gọi là WML 2.0. WML 2.0 bao gồm các phần tử “tương hợp” của WML 1.x (chẳng hạn như: <wml:card>, <wml:do>, v.v...), chúng được tự động dịch sang WML 2.0 nhờ vào các WAP gateway. Do Nokia Mobile Browser có hỗ trợ sẵn WML 1.x, nên việc biên dịch sẽ là không cần thiết [11] [12].

	WML 1.x	XHTML Mobile Profile và CSS
Sự chuẩn hoá	Chuẩn này được phát triển bởi WAP Forum	Chuẩn này được phát triển bởi W3C và được sự chấp nhận của OMA
Hiển thị nội dung trên thiết bị	Nội dung và cách trình bày được định nghĩa trong cùng một tài liệu, điều này khiến cho việc hiển thị bị biến đổi trên các thiết bị khác nhau.	Nội dung và cách trình bày được định nghĩa trong các tài liệu riêng, do đó cùng một nội dung sẽ thể hiện theo cách khác nhau nhờ vào các bảng định dạng khác nhau.
Mã hoá nội dung	Nội dung cần được mã hoá thành dạng nhị phân	Không cần mã hoá
Hỗ trợ màu sắc	Chỉ hỗ trợ các ảnh màu, không hỗ trợ cho font chữ, màu nền và các đường viền...	Đầy đủ các điều khiển màu sắc hỗ trợ font chữ, màu nền và đường viền... với CSS trên các thiết bị có hỗ trợ màu.

Bảng 2.6-2: Những khác nhau phổ biến giữa XHTML MP với CSS và WML 1.x

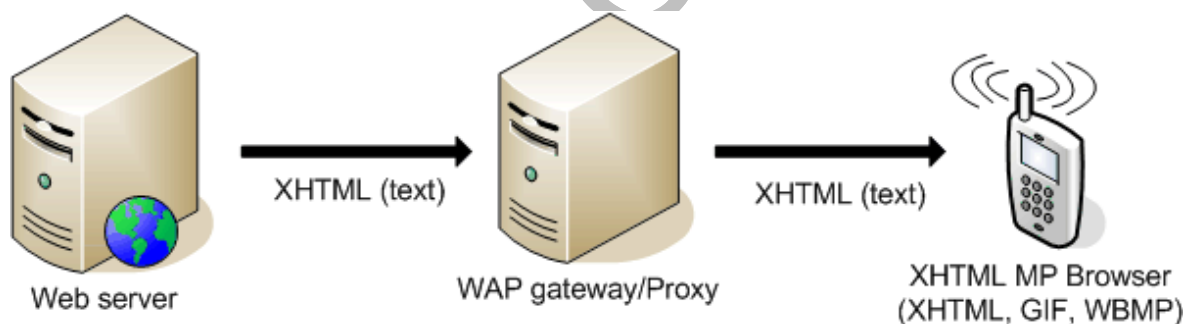
2.7. Các giao thức chuyển tải WML và XHTML

WAP 1.2.1 cũng như các site dành cho thiết bị di động được cấu hình để có thể nhận biết và xử lý các nội dung được viết bằng WML 1.x. WML cần một WAP gateway để mã hóa WML và WMLScript trước khi nó được gửi trên đường truyền (thường là sóng lan truyền trong không khí) đến thiết bị di động đầu cuối. Nội dung được mã hóa sang dạng nhị phân được gọi là WBXML.

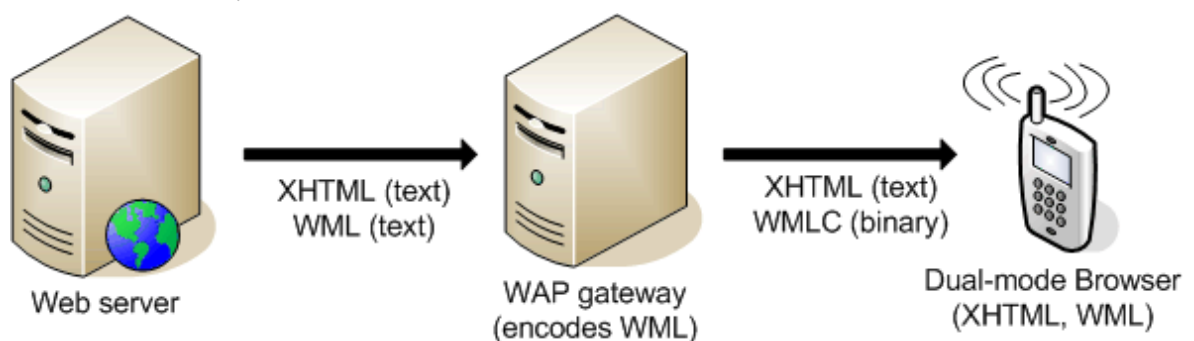


Hình 2.7-1: Sự truyền tải nội dung WML

Không giống như nội dung WML, các nội dung được viết bằng XHTML không cần phải được mã hóa tại WAP gateway mà sẽ chuyển thẳng đi mà không thay đổi gì cả. Bởi vì các nội dung WML đang tồn tại vẫn còn hữu dụng trong một thời gian nữa nên các trình duyệt hỗ trợ chuẩn WAP 2.0 có thể hiển thị cả các nội dung XHTML MP và WML 1.x. Những trình duyệt như thế được gọi là trình duyệt chế độ kép (dual-mode browser).



Hình 2.7-2: Sự truyền tải nội dung XHTML






Hình 2.7-3: Sự truyền tải phối hợp WML và XHTML

2.8. Cuộc cách mạng của trình duyệt WAP

Chính sự hiệu quả của chuẩn WAP 2.0, các thiết bị cầm tay mà cụ thể là các thế hệ điện thoại di động mới gần đây đã đẩy mạnh hỗ trợ WAP 2.0. Nokia là một trong những nhãn hiệu hàng đầu với phần lớn các model điện thoại có hỗ trợ WAP 2.0.

KHOA CNTT

Các thế hệ trình duyệt WAP	Các đặc điểm và tính năng chính
<p>WAP 1.0</p> 	<p>Màn hình trắng đen</p> <p>Sử dụng chuẩn WML, WAP stack</p> <p>Giá của dịch vụ được tính trên thời gian truy cập.</p> <p>Thời gian kết nối/thiết lập cao (15-30 giây cho trang đầu tiên)</p>
<p>WAP 1.2.1</p> 	<p>Màn hình màu</p> <p>Công nghệ mạng GPRS, WAP Push, Cookies</p> <p>Giá của dịch vụ được tính trên dung lượng sử dụng</p> <p>Thời gian kết nối nhanh (<3s cho trang đầu tiên)</p>
<p>WAP 2.0</p> 	<p>Màn hình màu kích thước lớn hơn, độ phân giải và số màu cao hơn</p> <p>Công nghệ WAP CSS, TCP/IP stack</p> <p>Khả năng hiển thị hình ảnh và điều khiển cao</p> <p>Dễ dàng chuyển đổi nội dung để phù hợp với các thiết bị di động khác nhau</p> <p>Hỗ trợ caching giúp tốc độ truy cập dịch vụ cao hơn</p> <p>Có thể download các file kích thước lớn.</p>

Bảng 2.8-1: Cuộc cách mạng trình duyệt WAP trên các thế hệ điện thoại di động của Nokia

Với khả năng thích ứng cao, XHTML hứa hẹn sẽ thúc đẩy lượng thông tin trao đổi thông qua các thiết bị di động.

KHOA CNTT

PHẦN II.
CÔNG CỤ TÌM KIẾM
SEARCH ENGINE

Chương 3. TỔNG QUAN VỀ MÁY TÌM KIẾM

3.1. Sơ lược về máy tìm kiếm

Máy tìm kiếm là một công cụ rất hữu ích giúp người dùng sử dụng nguồn tài nguyên trên Internet một cách hiệu quả nhất. Dựa vào máy tìm kiếm con người có thể tìm được rất nhiều thông tin liên quan đến mọi ngành, mọi lĩnh vực. Một hệ thống tìm kiếm thông tin bao gồm 3 mô-đun cơ bản:

- Thu thập thông tin từ internet.
- Phân tích, lượng hoá và rút trích thông tin cần thiết để lưu trữ vào cơ sở dữ liệu hệ thống
- Tiếp nhận yêu cầu từ người sử dụng, thực hiện tìm kiếm trong cơ sở dữ liệu và trả kết quả về cho người sử dụng.

Mỗi mô-đun có nhiều cách tiếp cận khác nhau, nên một hệ thống tìm kiếm cũng sẽ có phương thức xử lý và hoạt động khác nhau.

3.2. Phân loại máy tìm kiếm

Theo David P.Habib và Robert L.Barriot[6] có bốn loại công cụ tìm kiếm tất cả, đó là:

- Công cụ tìm kiếm thư mục (A directory search tool): tìm thông tin theo chủ đề, đó là dạng tìm phân cấp.
- Công cụ máy tìm kiếm : tìm thông tin theo từ khoá.
- Công cụ máy tìm kiếm cùng với thư mục: tìm kiếm dựa trên cả chủ đề và từ khoá .
- Công cụ máy tìm kiếm meta: sử dụng kết quả từ các máy tìm kiếm khác.

Vì mục tiêu chính của đề tài là nghiên cứu về công cụ máy tìm kiếm nên phần sau chỉ trình bày chi tiết hai loại, đó là máy tìm kiếm meta và máy tìm kiếm thông thường.

3.2.1. Máy tìm kiếm meta

Không giống như những máy tìm kiếm thông thường khác, máy tìm kiếm meta không có cơ sở dữ liệu riêng cũng như không có hệ thống lập chỉ mục tài liệu và robot. Chính vì vậy máy tìm kiếm meta hoạt động dựa vào cơ sở dữ liệu của những hệ thống khác. Tiếp nhận câu truy vấn của người dùng, dựa vào cơ sở dữ liệu của những hệ thống khác chọn ra những tài liệu có độ tương thích cao (cần phải chọn bao nhiêu, phải chọn thế nào tùy vào đặc điểm của từng hệ thống cụ thể) và trả kết quả cho người dùng[3][4].

Trong một máy tìm kiếm meta, ta nhập từ khoá cần tìm vào, nó sẽ chuyển từ đó đồng thời đến nhiều máy tìm kiếm cá nhân, trong một vài giây, ta nhận kết quả trả về từ tất cả những máy tìm kiếm được truy vấn.

Ý tưởng của việc tìm kiếm meta thì rất tốt, sẽ tiết kiệm nhiều thời gian bởi việc tìm kiếm chỉ ở một nơi và không cần đến việc sử dụng và học một số máy tìm kiếm khác nhau.

Có ba loại của máy tìm kiếm meta vào thời điểm này[3]:

- Trước tiên, những công cụ máy tìm kiếm meta sử dụng cho việc đào sâu trong nhiều nguồn tài nguyên, với khả năng hùng mạnh giúp tìm những gì cần thiết nằm trong kết quả tìm kiếm. Các công cụ dạng này[3] là:

Công cụ tìm kiếm meta	Công cụ thực hiện	Khả năng nâng cao	Kết quả hiển thị
SurfWax http://www.surfwax.com/	Vào bộ tìm kiếm và chọn từ một danh sách của các máy tìm kiếm gồm: AOL, Alltheweb, , Excite, Google, Hotbot, MSN, NBCi, Opendirectory, Yahoo! Có thể trộn với educational, US Govt tools, nguồn tin tức hoặc nhiều loại khác.	Chấp nhận “ “, +/-. Mặc định là AND giữa các từ.	Kết quả có thể được sắp xếp theo độ chính xác, hoặc theo thứ tự tiêu đề A-Z. Chọn liên kết nguồn để xem kết quả tìm được. FocusWords từ một trang đại diện ngữ cảnh. Thống kê hình ảnh và liên kết trên những trang chung.
Copernic Basic 2001 http://www.copernic.com/	Chọn Google và những công cụ khác từ danh sách những máy tìm kiếm bằng cách ấn vào nút Properties ở hộp tìm nâng cao. Một vài chọn lựa tốt: Altavista, AOL, Euroseek, Fast/alltheweb, Google, Hotbot.v.v.	ALL, ANY, Phrase, more. Tìm luận lí . Tích hợp với Internet Explorer , không dùng Netscape.	Phải được download và cài đặt, nhưng phiên bản cơ bản thì miễn phí. Có nhiều đặc tính có lợi, có thể thay đổi kết quả hiển thị, tìm lại mục trước.

Bảng 3.2-1: Các công cụ tìm kiếm meta dạng 1

- Thứ hai là máy tìm kiếm meta Good, chấp nhận những tìm kiếm phức tạp, tích hợp kết quả tốt, loại trừ trùng lặp, và những đặc tính

truyền thống như sắp xếp thông minh hoặc phân nhóm dựa vào đề tài trong kết quả tìm kiếm. Các công cụ dạng này[3] là:

Công cụ tìm kiếm meta	Công cụ để thực hiện tìm	Khả năng tìm phức tạp	Kết quả hiển thị
Ixquick http://www.ixquick.com/	AOL, All the web, Ask Jeeves/Direct Hit, Entireweb, Go, HotBot,.v.v.	Dịch những tìm kiếm phức tạp vừa phải thành cú pháp lệnh của máy tìm kiếm, hỗ trợ những dạng căn bản của luận lí và cụm từ nhưng không hỗ trợ sử dụng kí tự đại diện, dấu ngoặc đơn, NEAR.	Chọn 10 dòng đầu của kết quả tìm từ mỗi máy tìm kiếm và gộp chung kết quả(tập hợp kết quả). Loại bỏ trùng lặp
Vivisimo http://www.vivisimo.com/	Alltheweb/Fast. Yahoo!, MSN, AOL, Netscape và một vài khả năng khác cho những trang web chung,	Chấp nhận và dịch những tìm kiếm phức tạp với toán tử luận lí.	Kết quả đi kèm với sự chia nhỏ chủ đề dựa vào những từ trong kết quả tìm kiếm

Bảng 3.2-2: Các công cụ tìm kiếm meta dạng 2

- Thứ ba là máy tìm kiếm meta mà tìm một số nơi và trả kết quả không có những đặc tính trên[3].

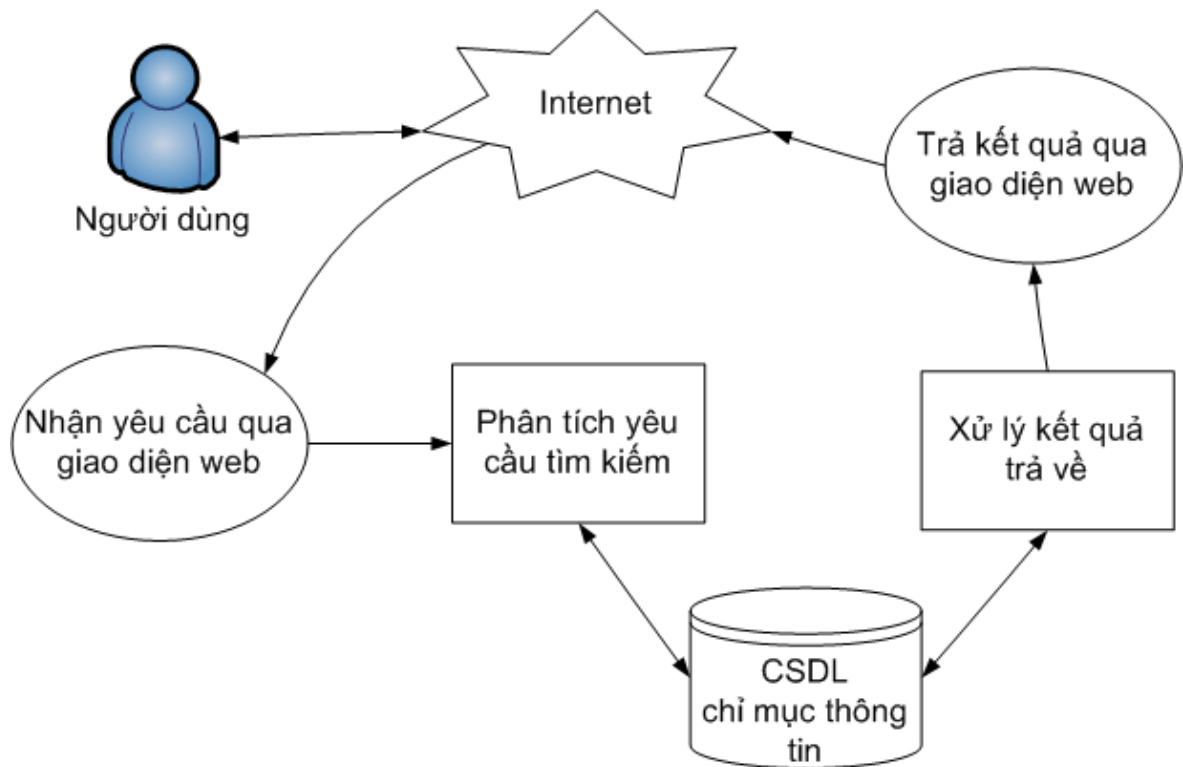
Những mặt không thuận lợi của loại này:

- Hầu hết những máy tìm kiếm meta có sẵn miễn phí đều tìm trên Google.
- Nếu kết quả không tìm thấy, điều mà ta có thể làm là thêm một thuật ngữ và tìm nơi nào máy tìm kiếm meta đang gởi nó.
- Không có máy tìm kiếm meta nào truy vấn đến tất cả máy tìm kiếm mà nó muốn truy vấn. Chúng tìm những gì có sẵn ở lúc ta submit truy vấn, và ta không biết chắc những gì nó truy vấn cho đến khi đọc được kết quả.

3.2.2. Máy tìm kiếm thông thường

3.2.2.1. Nguyên lý hoạt động của một máy tìm kiếm

Một máy tìm kiếm cơ bản gồm ba phần chính[2], mô tả như hình vẽ 3.2-1:



Hình 3.2-1: Mô hình hoạt động của máy tìm kiếm

3.2.2.2. Hệ thống thu thập dữ liệu (robot, spider, crawler...)

Chịu trách nhiệm tìm thông tin trên Internet từ địa chỉ URL cho trước. Thông thường hệ thống máy tìm kiếm có một server chịu trách nhiệm gửi những URL cho các robot. Robot tải tài liệu về và lưu trong kho dữ liệu, rồi lại tiếp tục nhận URL khác, tiến trình hoạt động liên tục từ khi khởi động hệ thống. Robot lấy những nội dung dạng văn bản, còn hình ảnh hay âm thanh hay những dữ liệu không thể đánh chỉ mục thì bỏ qua[5][1].

3.2.2.3. Hệ thống phân tích và lập chỉ mục dữ liệu

(Hay còn gọi là hệ thống lập chỉ mục(indexer)):

Chịu trách nhiệm phân tích tài liệu và lập chỉ mục thông tin cho tài liệu đó. Từ kho dữ liệu hệ thống lập chỉ mục tiến hành lấy từng tài liệu trong kho, lọc bỏ thông tin thừa(loại bỏ tag định dạng, ghi chú, script ...), phân tích các siêu liên kết

và lưu chúng vào danh sách URL, bỏ các từ stop word (từ không có nghĩa trong tài liệu), đưa tài liệu về dạng thuần văn bản, tiến hành phân tích từ sau đó tính trọng số của từ đó và lập chỉ mục cho chúng và lưu chúng vào cơ sở dữ liệu. Bên cạnh kỹ thuật trên Google áp dụng phương pháp tính độ quan trọng của tài liệu (pagerank) dựa vào số lượng liên kết chỉ đến tài liệu. Ý tưởng đó là nếu một tài liệu có nhiều liên kết (hyperlink), hoặc từ một tài liệu có độ quan trọng cao chỉ đến thì được coi như là tài liệu đó cũng có độ quan trọng cao. Dựa vào kỹ thuật mà Google tự hào rằng mình có thể tìm kiếm được file ảnh, âm thanh và những tài liệu phi văn bản khác[1].

3.2.2.4. Hệ thống tìm kiếm (truy vấn dữ liệu)

Tiếp nhận câu truy vấn của người dùng thông qua giao diện web, phân tích câu truy vấn và trả kết quả tìm kiếm. Hệ thống này phân tích câu truy vấn của người dùng cũng giống như phương pháp phân tích từ của hệ thống lập chỉ mục, điều này nhằm nâng cao độ tương tự và tính chính xác của hệ thống. Sau đó tiến hành tìm kiếm trên cơ sở dữ liệu đã được lập chỉ mục và trả kết quả tìm kiếm cho người dùng[2].

*** Ghi chú: Các thao tác như lập chỉ mục, cập nhật thông tin, bảo quản nội dung trong cơ sở dữ liệu phải diễn ra thường xuyên để đảm bảo thông tin luôn đúng.

Chương 4. MÁY TÌM KIẾM HỖ TRỢ THIẾT BỊ DI ĐỘNG

4.1. Tìm hiểu các dịch vụ tìm kiếm hỗ trợ thiết bị di động hiện có

4.1.1. Google Mobile Search

Tháng 6/2005 vừa qua, đại gia về lĩnh vực tìm kiếm Google đã chính thức thử nghiệm một công cụ tìm kiếm hỗ trợ cho các thiết bị di động. Với khả năng truy cập thông tin trực tuyến trên 8 tỉ trang web và 1 tỉ hình ảnh [13].



Hình 4.1-1: Google Mobile Search

Các ngôn ngữ đánh dấu hỗ trợ:

- XHTML (WAP 2.0)
- WML (WAP 1.2)
- iMode¹
- PDA Devices

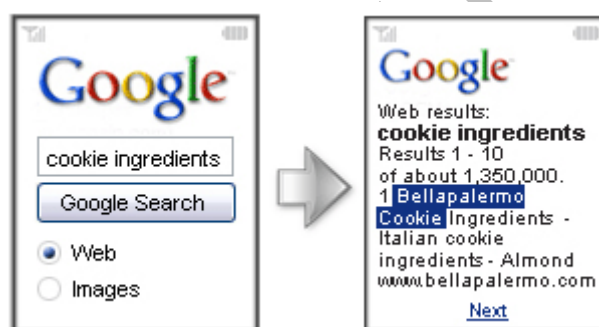
¹ iMode là một dịch vụ internet di động của NTTDoCoMo nhằm mục đích tạo ra các trang web sử dụng một dạng ngôn ngữ thuộc họ HTML (cHTML) hiển thị trên các trình duyệt Web di động.

Google hỗ trợ giao diện trên một số ngôn ngữ bao gồm: Trung Quốc, Đan Mạch, Hà Lan, Anh, Phần Lan, Pháp, Đức, Ý, Nhật, Hàn Quốc, Na Uy, Bồ Đào Nha, Rumani, Tây Ban Nha và Thụy Điển.

Với việc sử dụng ngôn ngữ XHTML cùng với WAP CSS, Google Mobile Search đã tạo nên một bước tiến mới trong việc hỗ trợ chức năng tìm kiếm hình ảnh và các trang web phức tạp, so với các công cụ đã có trước đây.

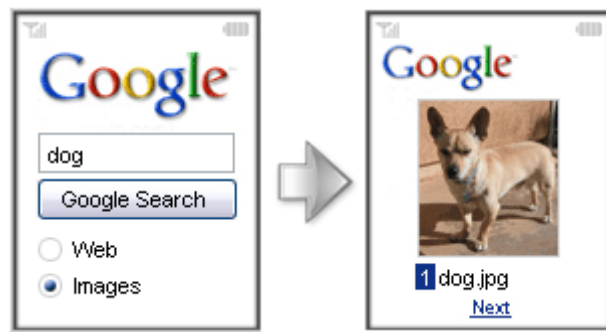
Google Mobile Search gồm có hai chức năng chính:

- **Tìm kiếm trang web:** khi chọn một liên kết trả về, Google sẽ định dạng lại trang nội dung này cho phù hợp với kích thước màn hình điện thoại của người sử dụng.



Hình 4.1-2: Chức năng tìm kiếm trang web của Google Mobile Search

- **Tìm kiếm hình ảnh:** dùng cho các thiết bị có hỗ trợ XHTML và iMode. Sau khi nhận được kết quả trả về, Google cũng định dạng lại cho phù hợp với màn hình hiển thị.



Hình 4.1-3: Chức năng tìm kiếm hình ảnh của Google Mobile Search

KHOA CNTT

4.1.2. Các máy tìm kiếm trên WAP hiện nay

URL	Chức năng
AllTheWeb mobile.alltheweb.com	Tìm kiếm HDML, WAP, i-mode, và các trang PDA.
Andamas	Máy tìm kiếm hiệu quả với chức năng tìm kiếm hình ảnh và một danh mục đường dẫn WAP hoàn chỉnh.
Cellno.net	Có khả năng tìm kiếm trên môi trường không dây và danh bạ điện thoại.
Google WAP Site www.google.com/wml	Máy tìm kiếm hỗ trợ thiết bị di động mới hỗ trợ XHTML.
Seek4Wap.com wap.seek4wap.com	Máy tìm kiếm cho WAP hỗ trợ 8 ngôn ngữ.
WAPALL http://www.wapall.com/ WAPALL	Danh mục WAP / máy tìm kiếm WAP
Yahoo UK WAP Site wap.yahoo.co.uk Yahoo UK Web Site Yahoo! Mobile Web Site	Portal WAP của Yahoo

Bảng 4.1-1: Các máy tìm kiếm hỗ trợ WML và WAP

4.2. Chuyển đổi các tài liệu sẵn có từ chuẩn web sang WAP

4.2.1. Nhu cầu chuyển đổi

Trong những năm gần đây việc truy cập thông tin di động ngày càng trở nên phổ biến. Ngày càng nhiều các thiết bị phần cứng hỗ trợ chức năng duyệt WAP, và thậm chí cả những trang web theo định dạng HTML (tuy vẫn còn rất hạn chế). Để

có thể đáp ứng được nhu cầu thông tin đang bùng nổ của công nghệ không dây thì làm sao trong thời gian ngắn có thể cung cấp lượng thông tin đủ để phục vụ cho nhu cầu của người truy cập WAP? Đây là một vấn đề đã và đang được các nhà phát triển rất quan tâm.

Hai khả năng có thể xảy ra. Đó là:

- Chỉ có một tập giới hạn các trang WAP sẽ được phát triển so với nhu cầu nếu chúng ta thực hiện xây dựng ngay từ đầu.
- Hoặc là số lượng các trang WAP sẽ tăng lên nhanh chóng nhờ vào các trang thông tin sẵn có (dạng HTML).

Trường hợp thứ nhất là một khả năng dễ nhận thấy. Nếu như chỉ dựa vào các URL và các thiết lập trên WAP gateway nó sẽ bó hẹp khả năng sử dụng của người dùng. Người dùng sẽ phải chấp nhận một lượng giới hạn các dịch vụ cung cấp cho thiết bị của họ, bởi vì mỗi một dòng sản phẩm của nó có những đặc tính riêng về kích thước màn hình, bộ nhớ cũng như khả năng hiển thị.

Trường hợp thứ hai đó là khi có mặt các công cụ chuyển đổi. Để có thể đáp ứng những dự đoán về sự phát triển to lớn của các thiết bị truy cập WAP, thì việc sử dụng những công cụ chuyển đổi là một giải pháp khả thi. Nếu được thực hiện tốt, nó sẽ giảm bớt vấn đề về hiển thị trên những thiết bị client khác nhau. Thế nhưng vẫn có một hạn chế, đó là nội dung trả về sẽ không được đảm bảo chất lượng, làm xảy ra các vấn đề như mất trang, hoặc một trang có thể phù hợp với thiết bị này nhưng lại không tương thích với thiết bị khác...

Hiện nay người ta vẫn đang dùng phối hợp cả hai phương pháp trên.

Một xu hướng mới đang rất được quan tâm đó là các phương pháp thiết kế tương thích cho việc hiển thị trên nhiều loại thiết bị, chẳng hạn như kỹ thuật XML-to-WML/XHTML, còn việc chuyển đổi các trang sẵn có sẽ ít được quan tâm hơn. Đối với giải pháp mới này, dữ liệu phụ trợ sẽ được hiển thị theo dạng XML và lưu trữ theo dạng XML, nó có thể dễ dàng được chuyển đổi sang HTML, WML hay bất

kỳ một ngôn ngữ đánh dấu nào khác. Do dữ liệu XML là một dạng trình diễn độc lập, các tiện ích dùng để chuyển đổi XML sẽ là các công cụ chuẩn được xây dựng ở phía server, ở các hệ điều hành và thậm chí là ở các client, chẳng hạn như IE hay Netscape.

Với việc sử dụng các công cụ chuyển đổi này giúp chúng ta có thể truy cập vào bất kỳ một trang HTML sẵn có ở bất kỳ đâu, và do đó nó cho phép thực hiện khi mà người phát triển:

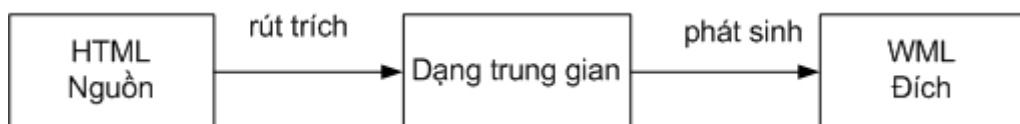
- Không biết gì về chi tiết thiết kế của trang web ban đầu.
- Không cần truy cập trực tiếp vào phía server.

Trong trường hợp thứ nhất, nó có thể hữu hiệu cho một công ty hay một nhà phát triển khác, họ có thể sử dụng công nghệ mà họ không hề có kiến thức về nó, chỉ muốn xây dựng lại nó mà thôi. Trong trường hợp thứ hai xảy ra khi người chủ của trang đó có thể không cho phép người khác truy cập trực tiếp vào cơ sở dữ liệu bên dưới của họ.

4.2.2. Hoạt động của các bộ chuyển đổi

Thông thường, các bộ chuyển đổi làm việc làm việc bằng các trích văn bản từ các trang nguồn (ví dụ các trang HTML), sau đó định dạng lại phần văn bản này về ngôn ngữ đánh dấu cần thiết, có thể là WML hoặc XHTML.

Sau đây là một ví dụ đơn giản:



Hình 4.2-1: Lược đồ mô tả quá trình chuyển đổi

Theo khuynh hướng thiết kế hướng đối tượng, trình chuyển đổi thực hiện việc chuyển đổi dữ liệu đã được định dạng thành dữ liệu trung gian. Dữ liệu trung gian này có thể được xử lý mà không cần đến các tiên trình phụ trợ chi phối.

Có hai hướng có thể sử dụng ở đây:

- Trích tất cả nội dung có thể trong một trang chẳng hạn như tiêu đề, thông điệp chào mừng, các liên kết ...
- Trích những phần riêng biệt của trang, ví dụ như các dòng tin, hay chỉ là các giá cả chính khoán.

Hai hướng này tương ứng với hai kiểu chuyển đổi là: tự động toàn bộ và có thể cấu hình được.

Trong quá trình chuyển đổi các phần tử văn bản trong trang, việc định vị từ trang này sang trang khác cũng rất cần thiết. Trong một trình chuyển đổi tự động hoàn toàn tốt, các liên kết và các form nhập liệu trên trang được chuyển đổi đều sẽ được giữ lại. Đối với trình chuyển đổi có thể cấu hình được, việc lựa chọn những phần nào sẽ được hiển thị tùy thuộc vào người phát triển.

4.2.3. Điều kiện quyết định khả năng chuyển đổi nội dung một tài liệu

Đặc điểm chính để quyết định xem một trang web có sẵn có phù hợp với cho việc chuyển đổi sang WAP hay không, đó là chúng phải có một lượng thông tin dạng văn bản tương đối nhỏ. Bên cạnh đó phải bảo đảm những yếu tố phù hợp với một dịch vụ WAP tốt đó là lượng thông tin phù hợp tối đa với thao tác nhập liệu tối thiểu. ‘Thông tin phù hợp tối đa’ có nghĩa là trang này cung cấp cho người dùng chỉ những thông tin mà họ quan tâm, với một lượng tối thiểu các phần phụ không liên quan khác. ‘Nhập liệu tối thiểu’ có nghĩa là có ít các định vị và việc nhập liệu từ phía người dùng.

Như vậy, khi cần thực hiện việc chuyển đổi thì các vấn đề mà ta cần quan tâm là:

- Trang cần chuyển đổi nên có lượng thông tin nhỏ.
- Các liên kết không quá nhiều (trong vòng bốn liên kết).
- Ít khung, ít hình ảnh và các ứng dụng đa truyền thông (multi-media)

Với thực tế các trang web rất phong phú, đa dạng như hiện nay sẽ có rất nhiều trang không thích hợp cho việc chuyển đổi.

4.2.4. Các ưu điểm

Những ưu điểm chính của việc chuyển đổi so với xây dựng một trang mới hoàn toàn là:

- Nhanh chóng hòa nhập vào thị trường
- Chi phí
- Độc lập với thiết kế ban đầu

Tính thị trường và chi phí là do nó giúp lập trình viên tốn ít thời gian hơn là bắt đầu lại từ điểm xuất phát. Một ưu điểm khác đó là người dùng có thể thực hiện chuyển đổi một trang nội dung mà không cần phải là một chuyên gia lập trình giỏi – một người thiết kế web với hiểu biết cơ bản về ngôn ngữ đánh dấu cũng có thể thực hiện một ứng dụng chuyển đổi dễ dàng.

Phần nội dung được rút trích từ trang gốc có thể được lưu ở một dạng độc lập, do đó nó có thể áp dụng được trên nhiều dạng client khác nhau. Với sự phát triển đa dạng của các thiết bị không dây như hiện nay thì vấn đề này đang rất được quan tâm.

Ngôn ngữ đích cần chuyển đổi không cần phải là WML – XML, nó có thể là bất kỳ một ngôn ngữ đánh dấu nào khác như: XHTML, CHTML, HDML...

Một ưu điểm khác nữa là khả năng tạo nên được một trang WAP mà không cần phải truy cập vào bên trong nguồn dữ liệu.

4.2.5. Các nhược điểm

Tuy nhiên phương pháp này cũng có nhiều nhược điểm.

Trong quá trình thực thi có hai vấn đề được đặt ra. Đó là:

- Vấn đề về các tiến trình liên quan đến server chứ không phải chỉ là phát sinh và cung cấp các trang WML từ một cơ sở dữ liệu nền. Giả sử khi 10.000 người dùng cùng truy cập vào trang cùng một lúc, nó sẽ làm cho server trở nên quá tải. Dẫn đến nhu cầu về nâng cấp phần cứng.
- Tài liệu HTML gốc có thể lớn hơn nhiều so với tài liệu WML tương đương.

Chi phí dành cho các công cụ để phát triển cũng là một hạn chế. Dù chúng ta có thể xây dựng được những bộ chuyển đổi đơn giản, thế nhưng để có được những trang kết quả hoàn chỉnh thì chi phí cho một bộ chuyển đổi đầy đủ là rất cao.

Ngoại trừ những nhược điểm được nêu ra ở trên, một số trang thậm chí còn không phù hợp cho việc chuyển đổi. Đó là chưa kể đến những trang đã phù hợp lại thường xuyên thay đổi định dạng hiển thị của mình.

PHẦN III.
ỨNG DỤNG MINH HỌA

Chương 5. PHÂN TÍCH THIẾT KẾ HỆ THỐNG “CÔNG CỤ TÌM KIẾM HỖ TRỢ THIẾT BỊ DI ĐỘNG”

5.1. Khảo sát hiện trạng

Theo dự đoán, trong tương lai số người sở hữu các thiết bị di động, nhất là điện thoại cầm tay, sẽ vượt xa số người sở hữu máy tính. Đồng thời với sự thay đổi mang tính cách mạng đó là sự phát triển mạnh mẽ của lượng thông tin trao đổi qua Internet, đặc biệt là trên các thiết bị di động.

Do đặc trưng của các thiết bị di động là sự hạn chế về dung lượng bộ nhớ, kích thước màn hình hiển thị, khả năng xử lý kém và băng thông thấp, do đó các thiết bị này khó có thể sánh ngang với máy tính để bàn và máy tính xách tay về phương diện lướt web tìm thông tin. Nhưng không vì thế mà thị trường cung cấp thông tin cho các thiết bị di động lại kém phát triển ngược lại đó là một thị trường đầy tiềm năng.

Ngày càng có nhiều người truy cập các thông tin hữu ích trên Internet thông qua các thiết bị cầm tay. Việc ra đời một tờ báo điện tử phục vụ cho điện thoại di động khá hiệu quả (ở Trung Quốc) và khả năng xử lý của điện thoại di động ngày càng được cải thiện sẽ giúp cho người dùng dần quen với việc đọc báo điện tử hoặc lấy thông tin từ Internet bằng các loại thiết bị cầm tay này. Từ đó sẽ nhanh chóng thúc đẩy nhu cầu tìm kiếm và khai thác thông tin trên Internet thông qua điện thoại di động nói riêng và các thiết bị cầm tay nói chung.

Với sự phát triển mạnh mẽ của các nội dung được trao đổi thì nhu cầu tìm kiếm thông tin của người dùng trên thiết bị di động cũng đang trở nên cần thiết. Tuy nhiên hiện tại chỉ có một số ít các nhà cung cấp dịch vụ tìm kiếm cho thiết bị di động, ví dụ như Google,... ; riêng tại Việt Nam vẫn chưa có hệ thống nào cung cấp dịch vụ này. (Xem phần 4.1 “Tìm hiểu các dịch vụ tìm kiếm hỗ trợ thiết bị di động hiện tại”). Và do đó đây vẫn là một thị trường đang còn bỏ ngỏ.

Một điều đặt ra cho các nhà phát triển là làm sao có thể giúp cho các thiết bị cầm tay có thể truy cập và hiển thị được các nội dung nằm trên các trang web truyền thống.

Có nhiều giải pháp để thực hiện vấn đề trên:

- Phát triển khả năng xử lý, hiển thị của phần cứng và nâng cao băng thông để có thể truy cập và hiển thị tốt các nội dung thuộc các trang web thuần túy – điều này phụ thuộc rất nhiều vào công nghệ mạng di động và công nghệ điện tử viễn thông. Đây là một phương án cực kỳ tốn kém và phải có một thời gian phát triển lâu dài. Đồng thời nó sẽ làm cho giá thành của thiết bị và dịch vụ tăng cao do đó sẽ không thu hút được khách hàng.
- Tạo ra một bộ chuyển đổi để chuyển đổi các thông tin từ trang web thuần túy sang các trang WAP (chuẩn WAP 1.x hoặc WAP 2.0) có thể hiển thị tốt trên thiết bị di động – phương án này sẽ hiệu quả và ít tốn kém hơn vì phần lớn các thiết bị di động hiện nay, đặc biệt là điện thoại di động, đã được tích hợp kèm theo một trình duyệt WAP, và các nội dung truyền tải trên trang WAP không đòi hỏi băng thông và khả năng xử lý của thiết bị di động cao.

5.2. Phân tích và xác định yêu cầu

Mục tiêu của đề tài là ứng dụng máy tìm kiếm, xây dựng một hệ thống hỗ trợ người dùng thiết bị di động tìm kiếm các thông tin trên Internet và giúp chuyển đổi nội dung trang web để có thể hiển thị trên bộ trình duyệt WAP của thiết bị di động, mà cụ thể là bộ trình duyệt WAP của các hệ điện thoại di động của Nokia và của một số hãng điện thoại di động khác.

Sau đây là danh sách các yêu cầu chức năng của hệ thống:

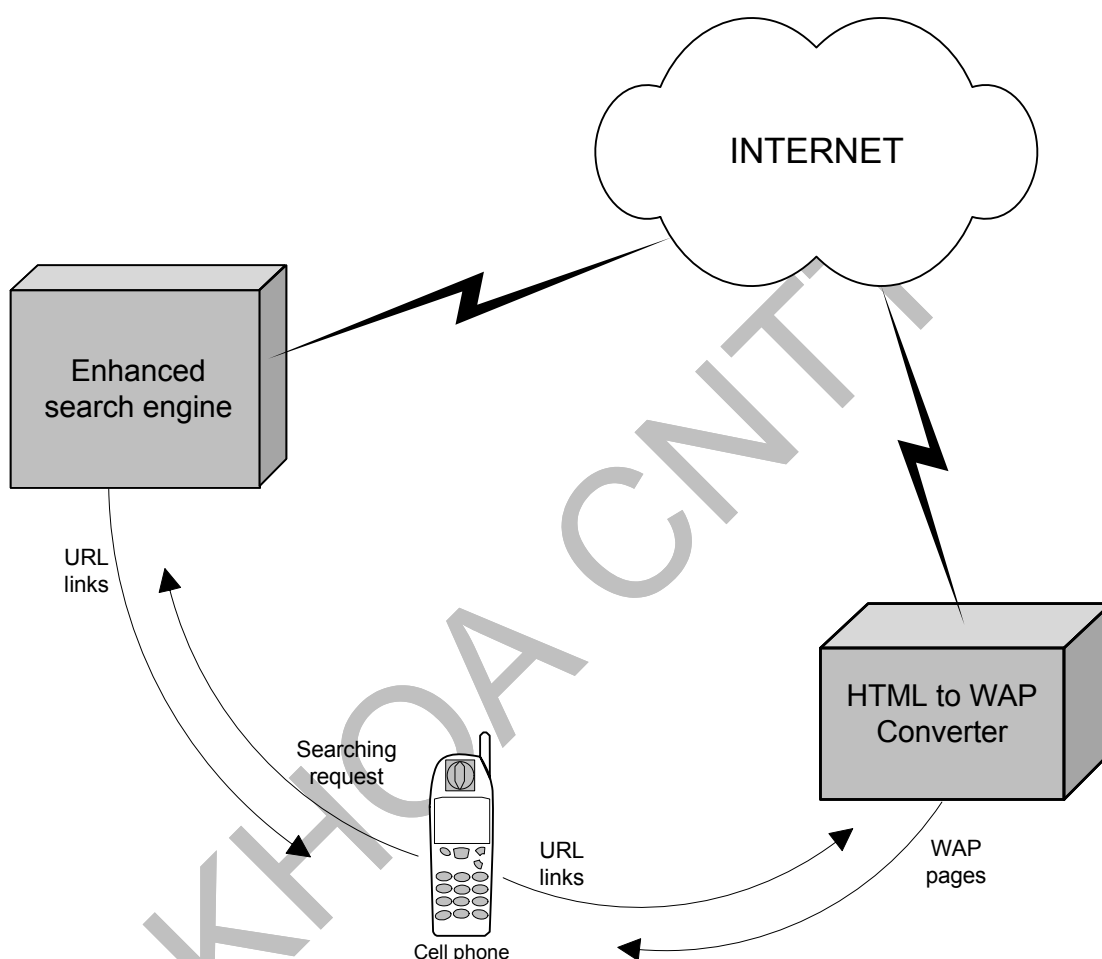
STT	Chức năng	Diễn giải
1	Xây dựng hệ thống cơ sở dữ liệu phục vụ cho việc tìm kiếm	Module được thực thi thường trực trên Server tải các trang web về, lập chỉ mục và đưa vào cơ sở dữ liệu phục vụ cho việc tìm kiếm. Đây là module của một máy tìm kiếm – search engine hoàn chỉnh.
2	Xác định khả năng xử lý của điện thoại	Hệ thống sẽ kiểm tra khả năng xử lý và hiển thị của điện thoại để chuyển đến trang dịch vụ thích hợp
3	Tiếp nhận chuỗi query để tìm kiếm	Nhận chuỗi tìm kiếm từ người truy cập thông qua giao diện wap.
4	Tìm kiếm	Tiến hành phân tích chuỗi query của người dùng và truy vấn cơ sở dữ liệu để tìm tập kết quả phù hợp với chuỗi query của người dùng.
5	Hiển thị kết quả	Trả kết quả tìm kiếm về cho người dùng thông qua giao diện wap
6	Chuyển đổi trang web	Kết quả tìm kiếm được liên kết đến chức năng chuyển đổi. Nếu người dùng click vào liên kết, chức năng chuyển đổi sẽ download trang web tương ứng và tiến hành chuyển đổi để có thể hiển thị trên điện thoại di động. Sau đó trả kết quả về cho người dùng.
7	Chức năng quản trị hệ thống tìm kiếm	Cung cấp giao diện cho người quản trị có thể quản lý hệ thống máy tìm kiếm để tiến hành các thao tác quản lý: thêm từ mới vào từ điển, chỉnh sửa từ điển, thêm URL

		download và lập chỉ mục...
--	--	----------------------------

Bảng 5.2-1: Xác định yêu cầu

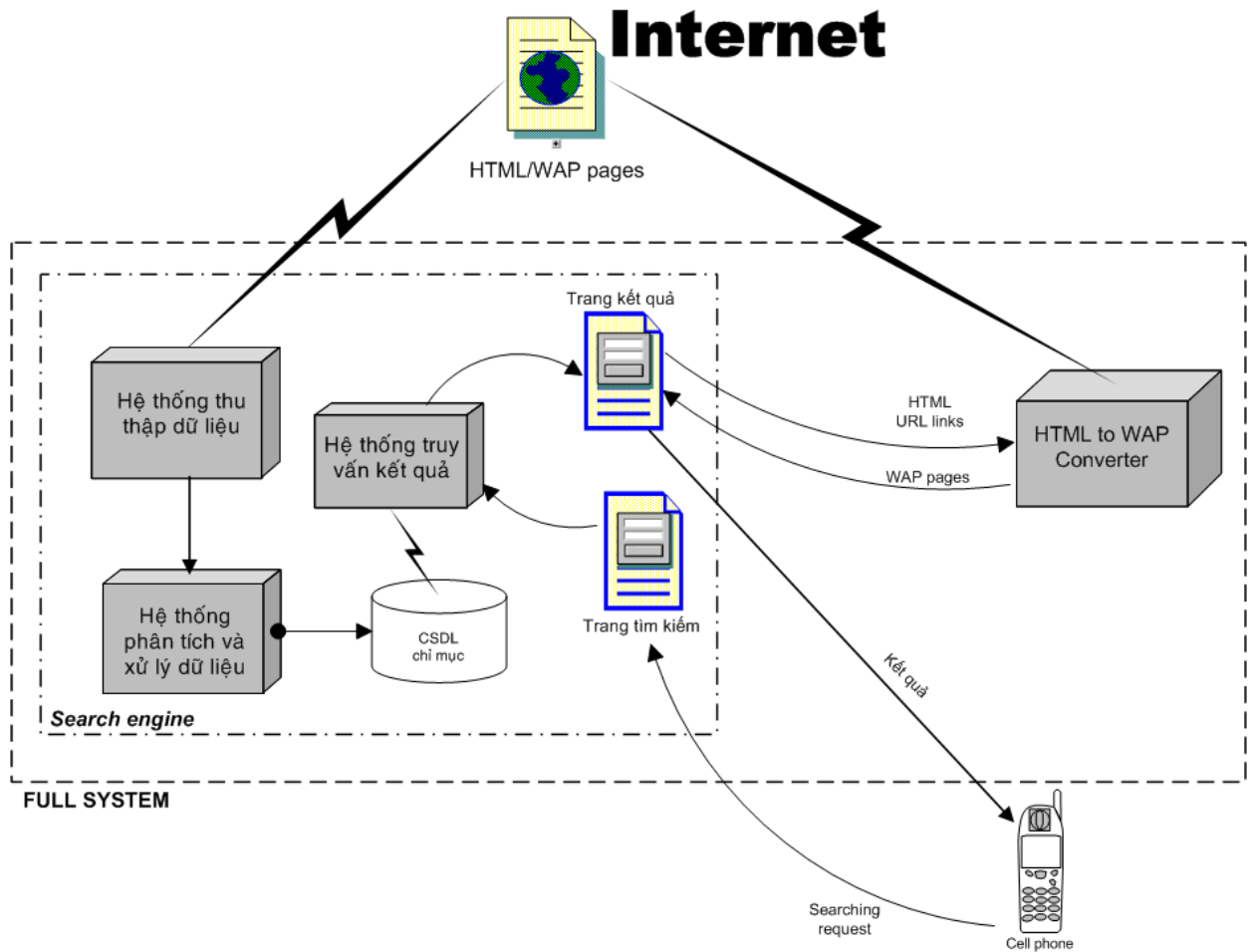
5.3. Mô hình hoạt động

5.3.1. Mô hình chung



Hình 5.3-1: Mô hình hoạt động chung

5.3.2. Mô hình chi tiết máy tìm kiếm



Hình 5.3-2: Mô hình hoạt động chi tiết

5.4. Mô hình Use-case

5.4.1. Xác định Actor và Use-case

Actor: Người sử dụng thiết bị di động truy cập vào hệ thống

Use-case:

- Connect – Kết nối:

Tiến hành truy cập vào hệ thống (được thực hiện bởi trình duyệt WAP được tích hợp trong thiết bị cầm tay)

- Search – Tìm kiếm:

Người dùng cung cấp từ khóa, hệ thống trả về kết quả tìm kiếm tương ứng với từ khóa đó.

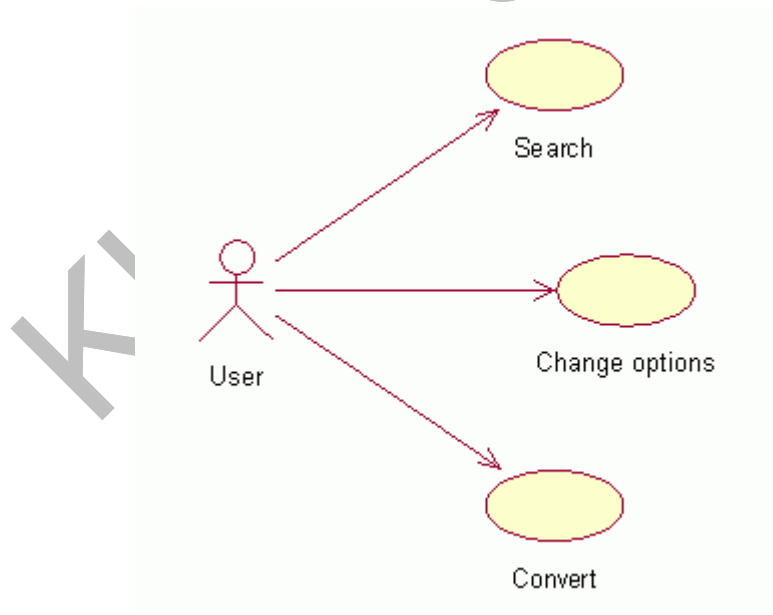
- Change options – Thay đổi thông số cho kết quả tìm kiếm

Người dùng chọn liên kết đến trang options để thay đổi thông số hiển thị cho kết quả.

- Convert – Chuyển đổi nội dung:

Người dùng chọn truy cập trang web từ trang kết quả, hệ thống tiến hành download và chuyển đổi trang web về dạng XHTML

5.4.2. Mô hình Use-case



Hình 5.4-1: Mô hình Use-case

5.5. Đặc tả Use-case

5.5.1. Tìm kiếm

- Tên Use-case: Search
- Mô tả

Người dùng cung cấp từ khóa, hệ thống trả về kết quả tìm kiếm tương ứng với từ khóa đó.

- Dòng sự kiện

- Dòng sự kiện chính

Người dùng nhập vào chuỗi query sau đó chọn tìm kiếm

Hệ thống nhận query, phân tích và kiểm tra trong từ điển

Truy vấn thông tin từ cơ sở dữ liệu để lấy ra tập kết quả

Trả kết quả tìm kiếm về cho người dùng

- Dòng sự kiện khác

Nếu người dùng không nhập query mà chọn tìm kiếm thì hệ thống sẽ không trả về kết quả.

- Các yêu cầu đặc biệt

Không

- Điều kiện tiên quyết

Không

- Điều kiện bổ sung

Không

- Điểm mở rộng

Không

5.5.2. Thay đổi thông số hiển thị kết quả tìm kiếm

- Tên Use-case: Change options

- Mô tả

Người dùng chọn liên kết đến trang options để thay đổi thông số hiển thị cho kết quả.

- Dòng sự kiện

- Dòng sự kiện chính

Người dùng chọn liên kết đến trang cung cấp tùy chọn

Người dùng chọn số kết quả trên một trang kết quả

Người dùng có thể chọn lưu thông số mới sau đó trở lại trang tìm kiếm chính

Người dùng có thể trở về trang tìm kiếm và giữ nguyên giá trị của thông số.

- Dòng sự kiện khác

Không

- Các yêu cầu đặc biệt

Không

- Điều kiện tiên quyết

Không

- Điều kiện bổ sung

Không

- Điểm mở rộng

Không

5.5.3. Chuyển đổi nội dung

- Tên Use-case: Convert

- Mô tả

Người dùng chọn truy cập trang web từ trang kết quả, hệ thống tiến hành download và chuyển đổi trang web về dạng XHTML.

- Dòng sự kiện

- Dòng sự kiện chính

Người dùng chọn click vào liên kết đến trang web được cung cấp trên trang kết quả

Hệ thống tiến hành download trang web tương ứng

Hệ thống kiểm tra và chuẩn hóa trang web sang dạng XHTML của chuẩn WAP 2.0

- Dòng sự kiện khác

Không

- Các yêu cầu đặc biệt

Không

- Điều kiện tiên quyết

Chỉ có thể thực hiện sau khi người dùng tìm kiếm và có kết quả trả về.

- Điều kiện bổ sung

Không

- Điểm mở rộng

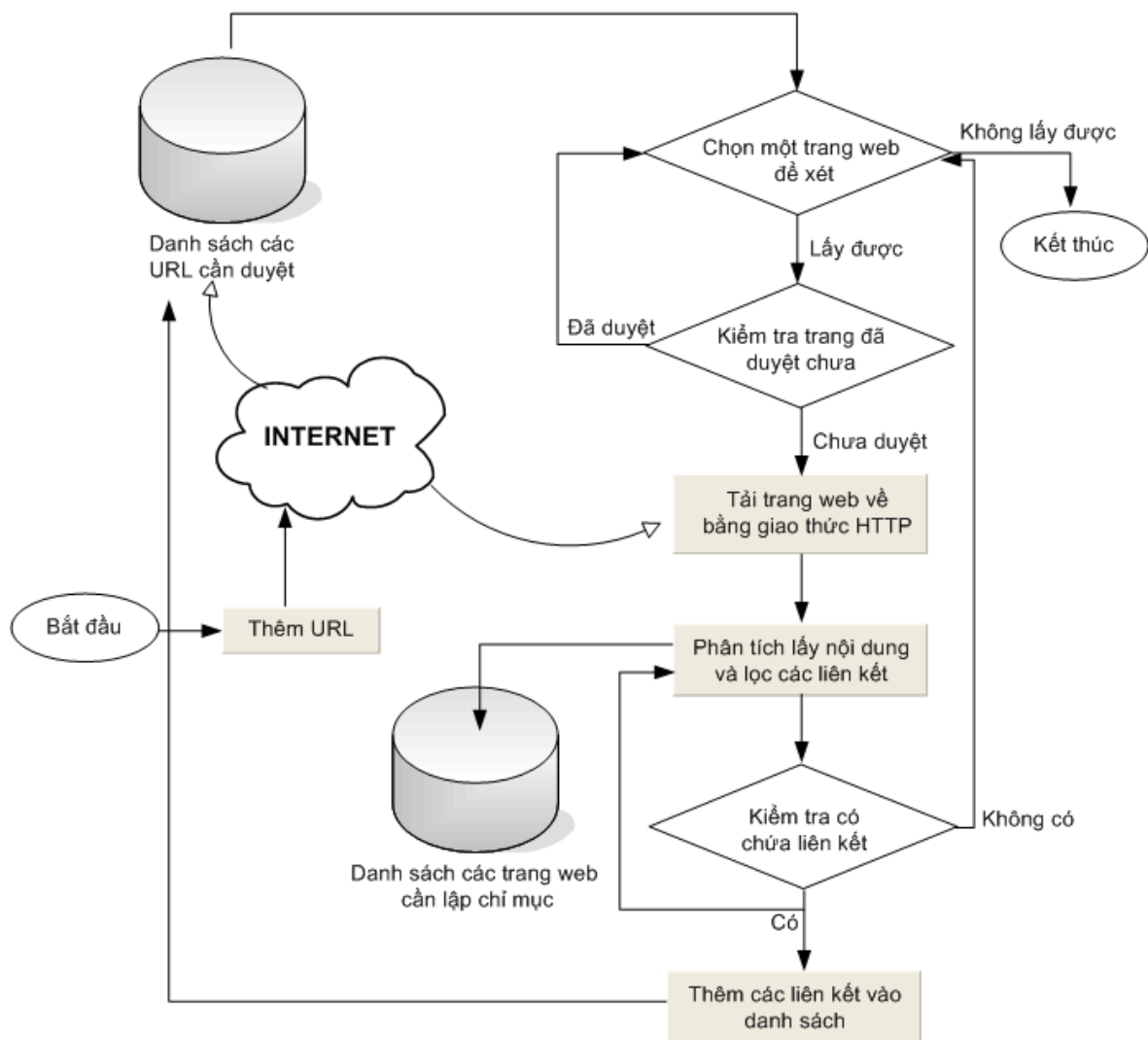
Không

5.6. Module Máy tìm kiếm – Search Engine

5.6.1. Phần thu thập dữ liệu

Trong hoạt động của một search engine thì công đoạn phải thực hiện đầu tiên khi khởi động hệ thống là thu thập tài liệu, đó là quá trình tập hợp tài liệu từ các nguồn khác nhau trên mạng. Việc thu thập tài liệu thường được cài đặt bằng các Web Robot hoặc các agent.

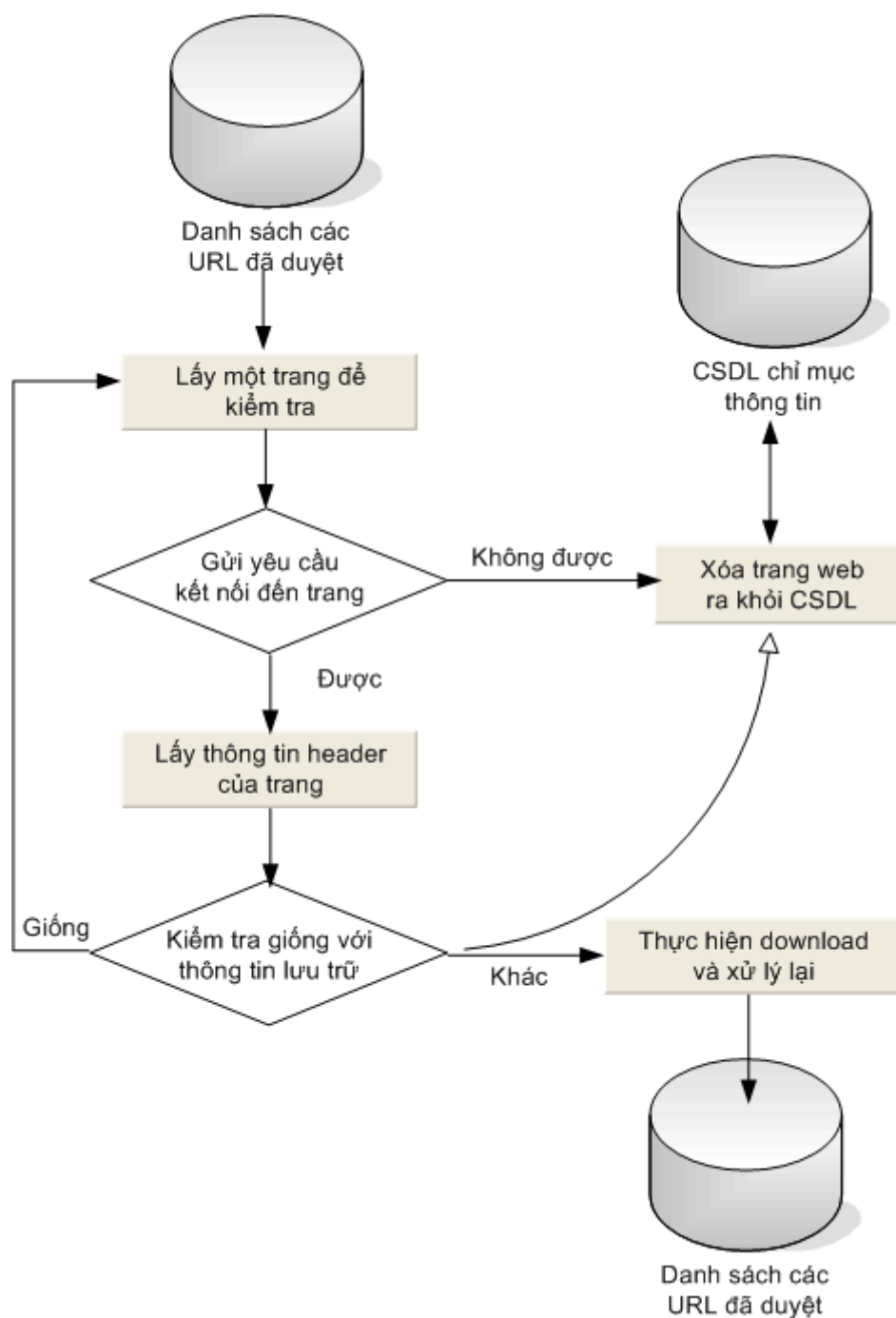
Web Robot là những chương trình có khả năng tự động dò tìm để đi đến các địa chỉ khác nhau trên mạng, tải các trang web về theo giao thức HTTP.



Hình 5.6-1: Lưu đồ xử lý quy trình thu thập thông tin

➤ Thuật toán duy trì thông tin cho máy tìm kiếm

Đối với các trang Web trên mạng Internet ngày nay, việc cập nhật và thay đổi dữ liệu rất thường xảy ra. Các thông tin trên mạng Internet luôn là những thông tin được cập nhật mới nhất. Có một số trang Web việc cập nhật xảy ra hàng ngày, thậm chí hàng giờ. Song song đó, cũng có một số trang web không còn tồn tại nữa, hoặc được chuyển đi nơi khác. Để đáp ứng nhu cầu có được những thông tin mới nhất cho người sử dụng và những gì tìm kiếm được qua search engine là tồn tại và có thể tham khảo tới được, cần phải liên tục duy trì và bảo quản các trang web đã tìm kiếm được.



Hình 5.6-2: Lưu đồ xử lý quy trình duy trì thông tin cho máy tìm kiếm

Thuật toán:

- Duyệt qua nội dung các trang web.
- Đối với mỗi trang web:
 - Kiểm tra xem tình trạng của trang web trên mạng Internet bằng cách đưa ra một yêu cầu kết nối đến trang web.
 - Nếu kết nối đến trang web thành công thì trang Web vẫn còn tồn tại. Qua bước kế tiếp
 - Nếu kết nối đến trang web không được , tức là trang web không còn tồn tại , thì thực hiện việc xóa trang web khỏi cơ sở dữ liệu.
 - Lấy các thông tin tiêu đề của trang web: Last Modified và Content-Length. Nếu một trong hai thông tin này khác với các thông tin được lưu trong cơ sở dữ liệu tức là nội dung trang web đã bị thay đổi. Thực hiện việc download lại trang web này về.
- Lặp lại công việc sau một khoảng thời gian qui định.

5.6.2. Phân thu thập dữ liệu

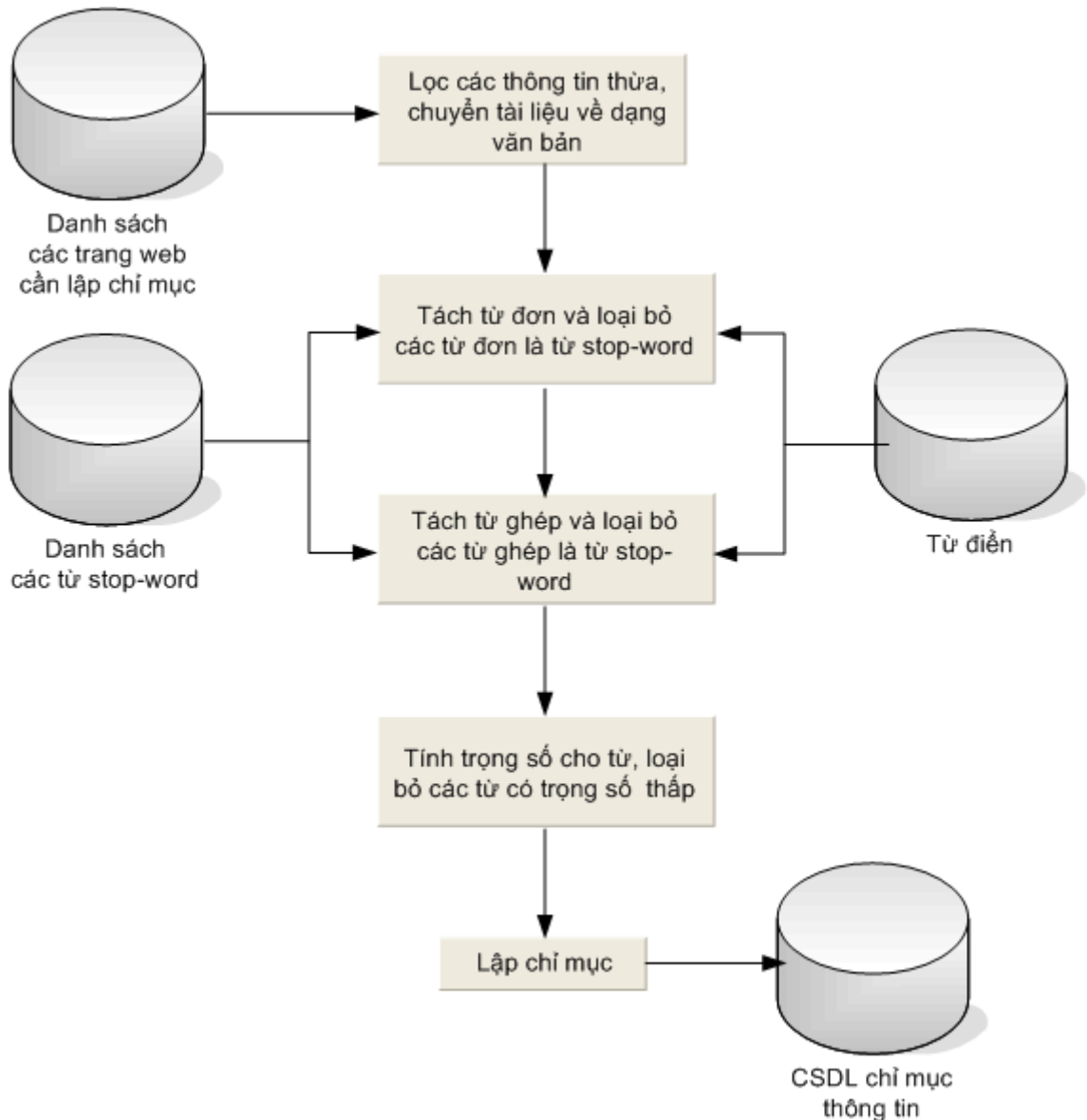
Các tài liệu thu thập về sẽ được hệ thống lập chỉ mục lọc các thông tin thừa, chuyển thành những tài liệu dưới dạng thuần văn bản. Trang web qua bước tiền xử lý để lọc bỏ các tag, các thông tin không liên quan tới nội dung tài liệu như chú thích, meta data, xử lý các kí tự đặc biệt (các kí tự được thay thế bằng chuỗi bắt đầu bằng dấu kí tự "&" và kết thúc bằng kí tự ";") để chuyển về dạng thông thường.

Ví dụ: chuỗi ">" tương ứng với kí tự ">"

chuỗi "&#" ứng với kí tự "&"

chuỗi """ ứng với kí tự "\"

Sau đó thực hiện công đoạn tách từ. Việc lập chỉ mục sẽ trích ra các mục từ quan trọng của tài liệu và ghi các mục từ này vào cơ sở dữ liệu của hệ thống với tần suất xuất hiện tương ứng của nó.



Hình 5.6-3: Lưu đồ xử lý qui trình lập chỉ mục

Khi hệ thống tìm kiếm đã khởi động thì mô-đun phân tích, xử lý dữ liệu sẽ hoạt động liên tục, thực hiện phân tích nội dung các trang web, bỏ các dữ liệu thừa và xử lý nội dung văn bản của tài liệu. Hoạt động này là một phần tách biệt trong hệ thống, không ảnh hưởng đến tốc độ và thời gian xử lý hay tìm kiếm thông tin

chung của toàn hệ thống. Mục đích chính là làm tăng khối lượng các trang web mà hệ thống đã xử lý, lưu trữ vào cơ sở dữ liệu để tìm kiếm sau này.

5.6.2.1. Thiết kế dữ liệu

Dữ liệu của hệ thống được xây dựng dựa trên mô hình vector, sử dụng phương pháp tổ chức tập tin nghịch đảo như đã trình bày trong các phần trước, bên cạnh đó có một số phần cải tiến nhằm đạt được kết quả tốt về thời gian xử lý cho hệ thống. Dữ liệu của hệ thống gồm 3 phần chính sau:

-Bảng định danh tài liệu: dùng hệ quản trị cơ sở dữ liệu Oracle quản lý, lưu trữ thông tin về các URL như: định danh, địa chỉ, tiêu đề,

-Tập tin nghịch đảo: chứa thông tin về các tài liệu và trọng số của mục từ trong tài liệu đó.

-Từ điển chỉ mục: chứa thông tin về một từ như số lần xuất hiện, số tài liệu có chứa từ đó, vị trí trong tập tin nghịch đảo

5.6.2.1.1. Bảng định danh tài liệu

Bảng định danh tài liệu lưu trữ danh sách các trang web đã được lập chỉ mục cùng các thông tin kèm theo của từng trang. Vì số lượng trang web lập chỉ mục lớn nên hệ thống sử dụng hệ quản trị cơ sở dữ liệu Oracle để quản lý. Cấu trúc bảng định danh tài liệu gồm các trường sau:

STT	TÊN TRƯỜNG	KIỂU	MÔ TẢ
1	DOCID	Number	Số định danh cho mỗi tài liệu, do hệ thống gán tự động
2	URL	Chuỗi	Địa chỉ của trang web
3	TITLE	Chuỗi	Tiêu đề của trang web
4	NGAY	Ngày	Ngày tạo lập của tài liệu
5	TRICHDAN	Chuỗi	Nội dung trích dẫn (chưa cài đặt trường này)

Bảng 5.6-1: Bảng định danh tài liệu

Các tác vụ chính trên bảng này là :

- Thêm một tài liệu

- Xóa một tài liệu
- Tìm kiếm tài liệu dựa trên DOCID. Tác vụ tìm kiếm được thực hiện thường xuyên, nên cài index cho trường DOCID để tăng tốc độ tìm kiếm.

5.6.2.1.2. Cấu trúc từ điển chỉ mục

Từ điển chỉ mục chứa danh sách các mục từ, hệ thống thực hiện yêu cầu tìm kiếm dựa trên các mục từ này. Số lượng từ trong từ điển rất lớn và từ điển thường xuyên được truy xuất nên cần phải có một cấu trúc hợp lý sao cho việc tìm kiếm một mục từ là nhanh nhất.

Từ điển có thể tổ chức theo danh sách tuyến tính được sắp xếp của các mục từ và thực hiện tìm kiếm nhị phân. Tuy nhiên, khi muốn thêm một mục từ vào cần phải sắp xếp lại từ điển, điều này rất khó cho việc quản lý từ điển và chi phí tốn kém. Giải pháp là xây dựng từ điển thành cây n phân biến thể thành cây nhị phân để dễ dàng cho cài đặt và xử lý.

Mỗi mục từ trong từ điển có một cấu trúc dữ liệu Info kèm theo, được gắn vào ký tự cuối cùng của mục từ. Cấu trúc Info gồm các trường sau:

```
struct Info {
    int n;
    int nDoc;
    int signal;
    int startPage;
    int endPage;
}
```

Trong đó:

- **n**: số lần xuất hiện của mục từ này trong danh sách các trang web mà hệ thống đã lập chỉ mục
- **nDoc**: số tài liệu có chứa mục từ này
- **signal**: xác định thuộc tính của mục từ này, có giá trị 0: mục từ là từ tiếng Anh hoặc từ tiếng Việt; giá trị 2: mục từ là từ stopwords

- **startPage**: trang bắt đầu trong chuỗi các trang trong file chỉ mục nghịch đảo thuộc về mục từ này
- **endPage**: trang cuối cùng trong danh sách các trang trong file chỉ mục nghịch đảo của mục từ này. Giá trị này sử dụng nhằm tăng tốc độ lập chỉ mục vì từ nó ta có thể truy xuất được đến trang cuối cùng nhanh nhất khi cần thêm tài liệu vào file nghịch đảo, chứ không cần phải duyệt tuần tự danh sách các trang từ trang bắt đầu.

Các thao tác chính trên tự điển gồm có

- Thêm một mục từ
- Xóa một mục từ
- Tìm kiếm một mục từ
- Cập nhật thông tin một mục từ

Tự điển được nạp hoàn toàn vào hệ thống.

5.6.2.1.3. Cấu trúc tập tin chỉ mục nghịch đảo

Tập tin nghịch đảo lưu trữ danh sách các tài liệu ứng với mục từ để cho hệ thống nhanh chóng tìm được danh sách các tài liệu có chứa một mục từ nào đó, tổ chức dưới dạng sau:

Mục từ	Tài liệu, trọng số
T_1	(2,w1), (3,w2), (4,w3)
T_2	(3,w4), (4,w5), (5,w6)
T_3	(2,w7)
T_4	(1,w9)

Bảng 5.6-2: Bảng mục từ

Bảng trên có nghĩa là mục từ T_1 xuất hiện trong các tài liệu 2, 3, 4 với các trọng số tương ứng là w1, w2, w3

Tập tin nghịch đảo được tổ chức bằng cách sử dụng một tập tin, trong đó dung lượng dành cho mỗi mục từ được cấp phát theo từng trang có kích thước cố định là 1024byte. Như vậy một trang có thể chứa $1024/8 - 1 = 127$ tài liệu (4 byte chứa DOCID, 4 byte chứa trọng số)

Cấu trúc một trang cấp cho từng mục từ trong file chỉ mục nghịch đảo như sau:

CHIỀU DÀI	TÊN TRƯỜNG
4	NextPage
4	NextPos
4	docId ₁
4	weighT ₁
4	docId ₂
4	weighT ₂
4	docId ₃
4	weighT ₃
.....
4	docId _n
4	weighT _n

Bảng 5.6-3: Cấu trúc trang cho từng mục từ trong tập tin chỉ mục nghịch đảo

- **NextPos**: chỉ đến vị trí trống tiếp theo chưa được sử dụng trên trang này, chỉ có ý nghĩa khi đây là trang cuối.
- **docId_i** ($i=1 \dots n$): định danh tài liệu có chứa mục từ sở hữu trang này, dựa vào định danh này có thể truy xuất thông tin liên quan đến tài liệu từ bảng định danh tài liệu.

- **weighT_i** (i=1...n): trọng số của mục từ trong từng tài liệu tương ứng docId_i

Như vậy có thể đọc toàn bộ danh sách các tài liệu có chứa một mục từ bằng cách đọc toàn bộ các trang được liên kết theo con trỏ nextPage. Vấn đề còn lại là xác định trang đầu tiên trong chuỗi danh sách này. Giá trị trang đầu tiên được lưu trữ trong từ điển chỉ mục đã trình bày ở trên. Các thao tác chính trên tập tin nghịch đảo chỉ mục là:

- **Thêm một tài liệu cho một mục từ:** khi một tài liệu được lập chỉ mục, nếu tài liệu có chứa một mục từ t nào đó thì tài liệu này sẽ được thêm vào danh sách các tài liệu ứng với mục từ t trong tập tin nghịch đảo. Tài liệu được thêm vào vị trí trống đầu tiên trong trang cuối của mục từ t.
- **Đọc danh sách của các tài liệu cho một mục từ:** kết quả tác vụ này được thiết kế trả về theo luồng (stream) dưới dạng (docId₁, weighT₁, docId₂, weighT₂,, docId_n, weighT_n) nghĩa là có thể đọc kết quả trả về theo từng tài liệu, xử lý xong tài liệu này mới đọc sang tài liệu tiếp theo. Việc trả kết quả theo luồng có ưu điểm lớn là giảm được dung lượng lưu trữ kết quả (chỉ cần một lượng bộ nhớ nhỏ bị chiếm dụng để là bộ đệm nhằm giảm bớt thao tác truy xuất file). Điều này là cần thiết vì danh sách các tài liệu trữ về có thể rất lớn (hàng chục ngàn trzng với hàng MB bộ nhớ) và hệ thống hoạt động trên môi trường mạng nên phải xử lý rất nhiều yêu cầu đồng thời, dẫn đến việc hệ thống có thể bị cạn kiệt tài nguyên bộ nhớ nếu phải lưu trữ toàn bộ kết quả trong bộ nhớ.

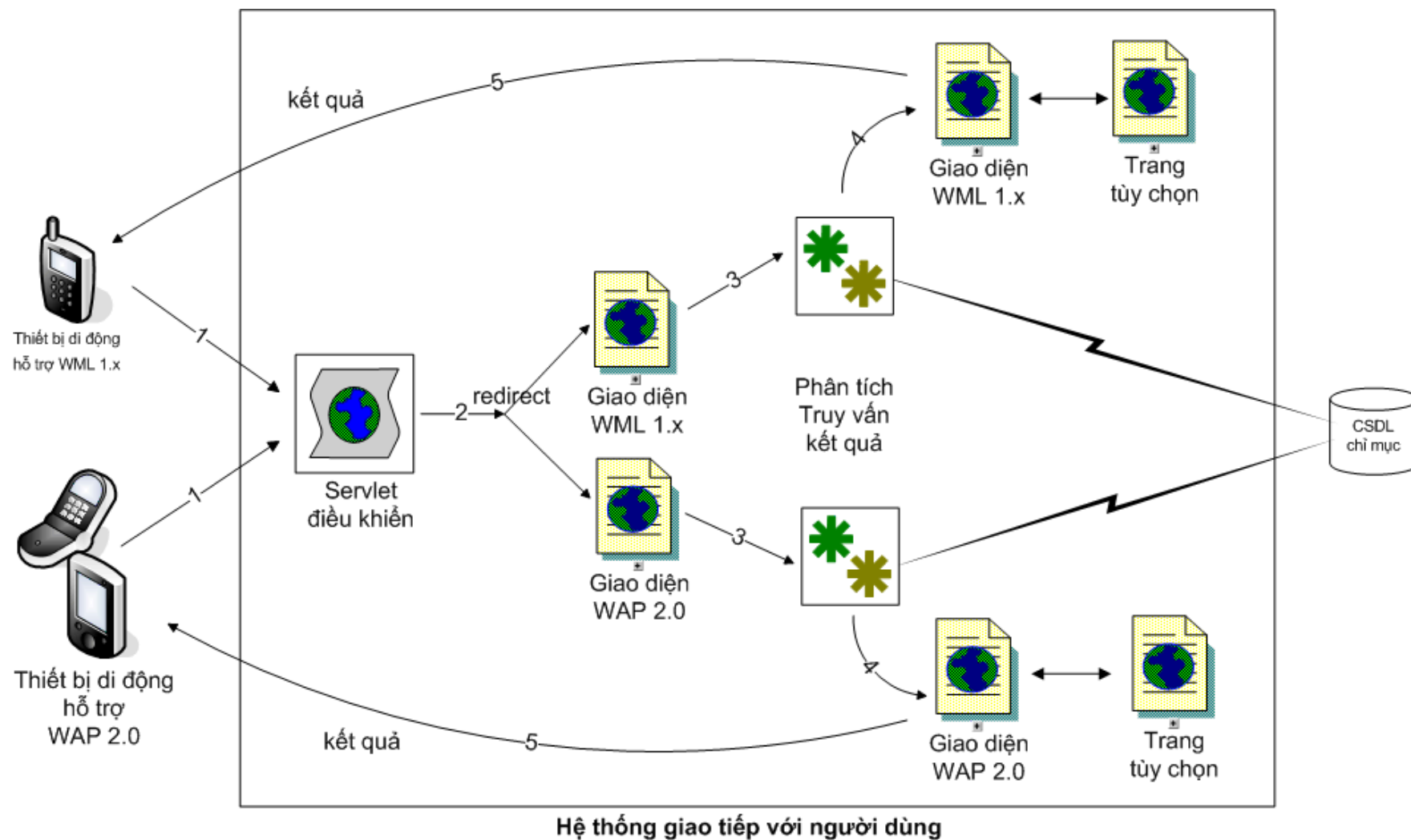
Với mô hình này dung lượng bộ nhớ bị chiếm dụng bởi một yêu cầu đọc danh sách các tài liệu trên file nghịch đảo không phụ thuộc vào số lượng kết quả trả về.

Mặt khác file nghịch đảo được truy cập thường xuyên khi xử lý yêu cầu tìm kiếm và thao tác đọc dữ liệu từ file nghịch đảo chiếm nhiều thời gian nhất trong tổng số thời gian cần thiết để hoàn tất một yêu cầu tìm kiếm. Vì dung lượng file

nghịch đảo thay đổi, tăng theo thời gian và số lượng file đã xử lý nên nó có thể trở nên quá lớn, dẫn đến tình trạng không thể lưu toàn bộ file nghịch đảo vào bộ nhớ. Do đó để tăng tốc độ tìm kiếm, thực hiện cấp phát một vùng nhớ đóng vai trò bộ đệm cho thao tác đọc file này. Bộ đệm chi thành các trang với dung lượng bằng dung lượng một trang cấp phát cho từng mục từ 1KB. Khi có yêu cầu truy xuất một trang trong file nghịch đảo, trang cần sẽ được nạp lên bộ đệm nếu chưa có trong bộ đệm và tồn tại ở đó để có thể sử dụng cho những lần truy xuất sau (không phải truy cập lại đĩa). Giải thuật thực hiện thay thế các trang trong bộ đệm là giải thuật lâu nhất chưa sử dụng.

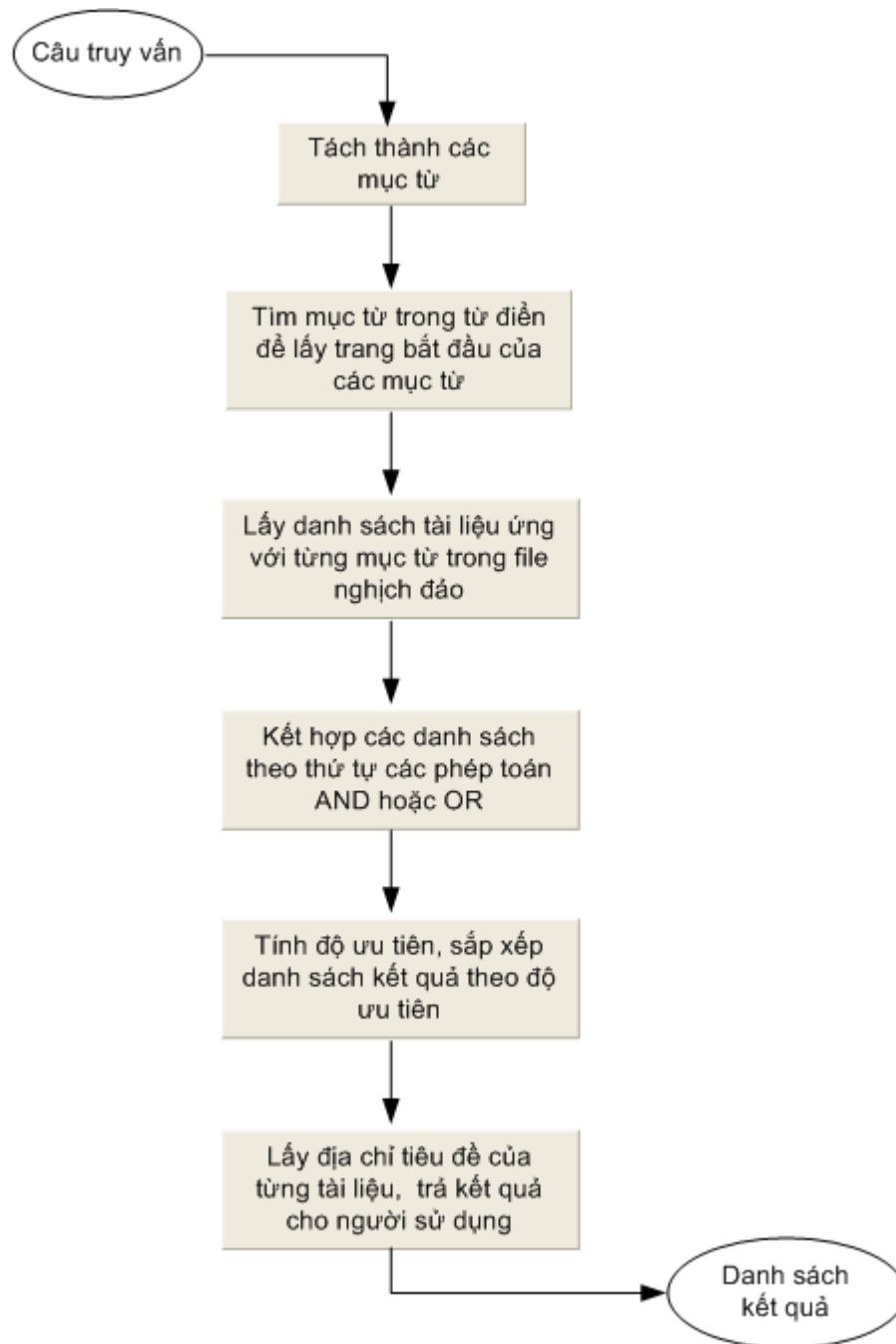
5.7. Module nhận và phân tích query từ người dùng

5.7.1. Mô hình hoạt động



Hình 5.7-1: Mô hình hoạt động module nhận và phân tích query từ người dùng

5.7.2. Mô hình xử lý



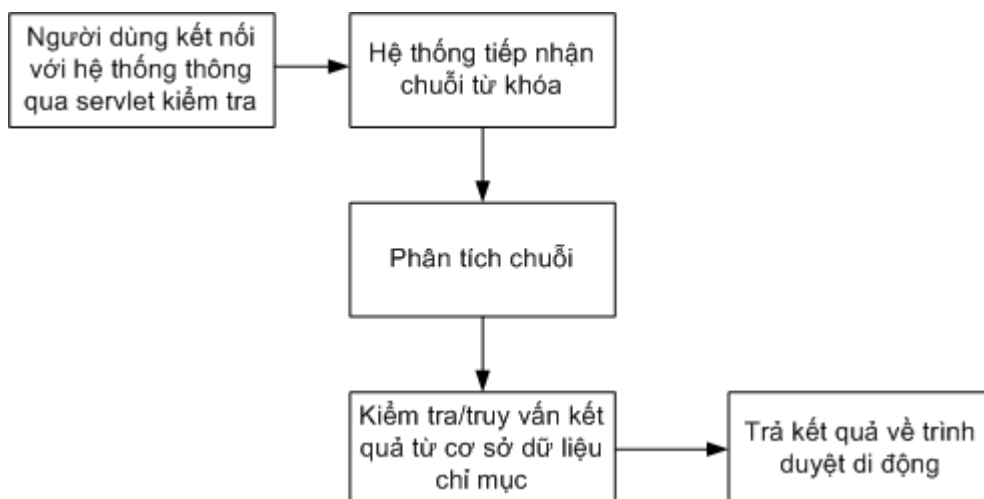
Hình 5.7-2: Mô hình xử lý quá trình nhận và phân tích query người dùng

5.7.3. Mô tả

Đây là hệ thống giao diện chính tiến hành giao tiếp với người dùng

Yêu cầu: Cung cấp giao diện, tiếp nhận yêu cầu tìm kiếm, phân tích yêu cầu và trả về kết quả tìm kiếm tương ứng

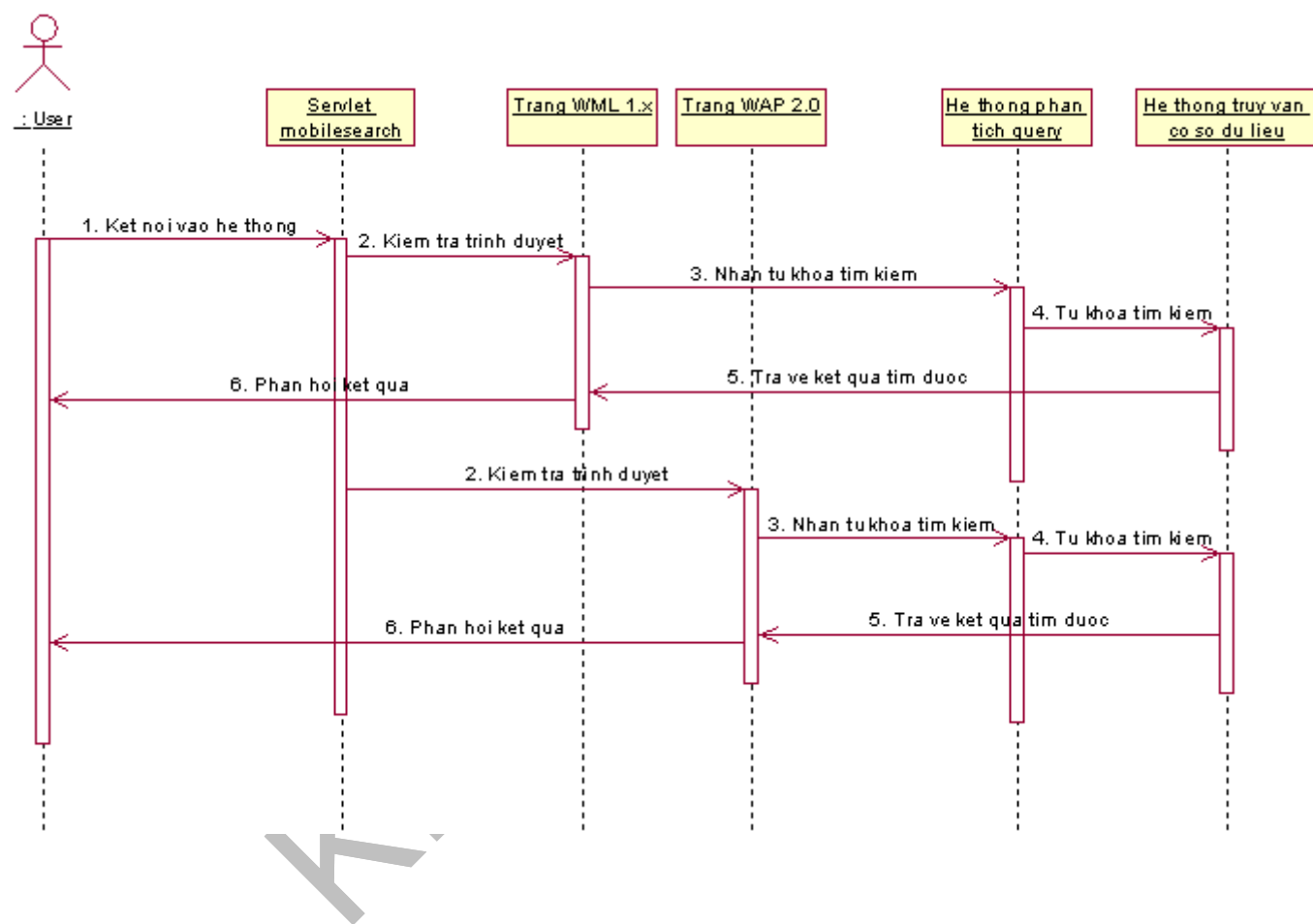
Các bước thực hiện:



Hình 5.7-3: Quá trình nhận và phân tích query từ người dùng

- Người dùng kết nối đến hệ thống thông qua một servlet kiểm tra: Servlet này sẽ kiểm tra khả năng hiển thị của trình duyệt đang kết nối sau đó chuyển yêu cầu đến trang giao diện tương ứng (WML 1.x hoặc WAP 2.0)
- Hệ thống tiếp nhận chuỗi từ khóa tìm kiếm
- Phân tích chuỗi
- Kiểm tra và truy vấn kết quả từ cơ sở dữ liệu chỉ mục
- Trả kết quả về cho người dùng thông qua giao diện tương ứng (được lưu theo session)

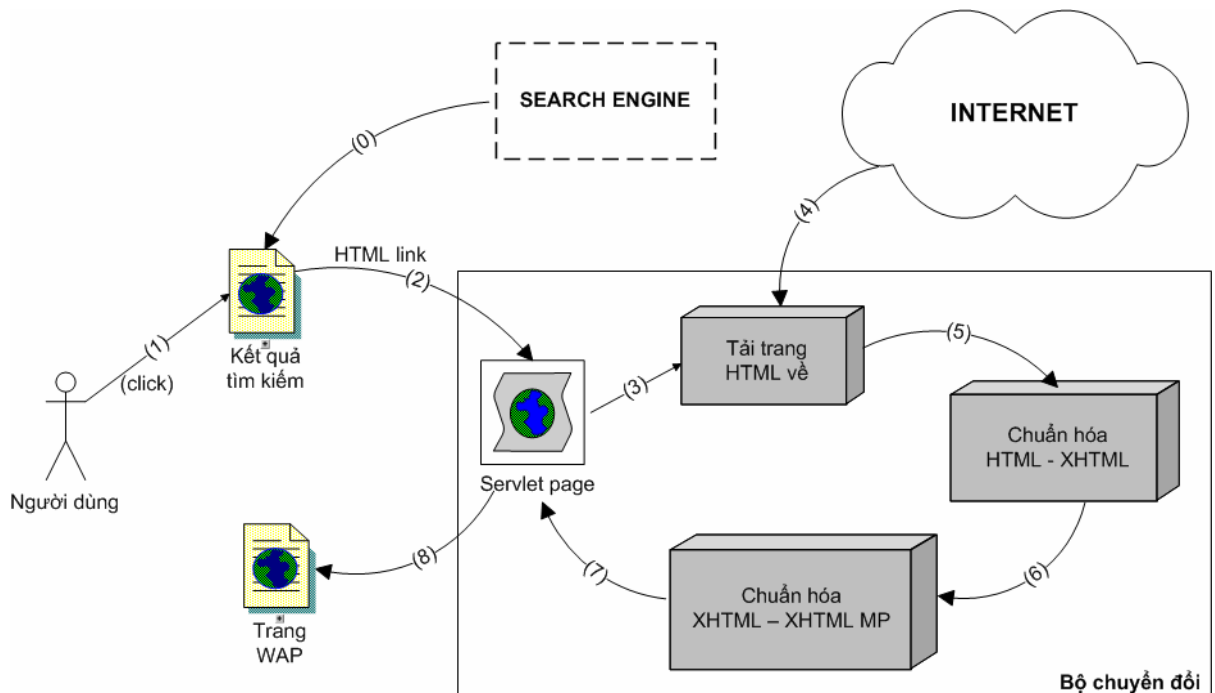
5.7.4. Mô hình sequence



Hình 5.7-4: Mô hình sequence

5.8. Module chuyển đổi trang web

5.8.1. Mô hình hoạt động



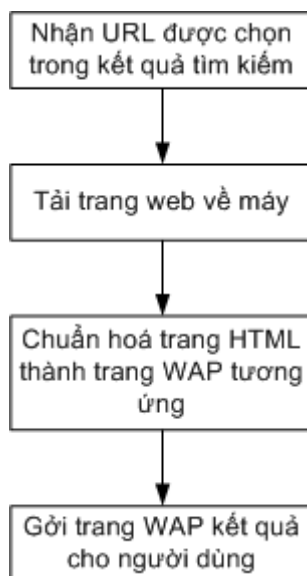
Hình 5.8-1: Mô hình hoạt động của bộ chuyển đổi

5.8.2. Mô tả

Module này sẽ thực hiện việc chuyển đổi các trang web thuần túy (chủ yếu là trang HTML) sang các trang thuộc chuẩn WAP (XHTML và WML) để có thể hiển thị trên các thiết bị di động giúp người dùng có thể xem được các thông tin hữu ích được trả về từ hệ thống tìm kiếm.

Yêu cầu: Nhận vào một liên kết và trả về một trang WAP theo chuẩn XHTML đối với trình duyệt 2.0 và WML đối với trình duyệt 1.x.

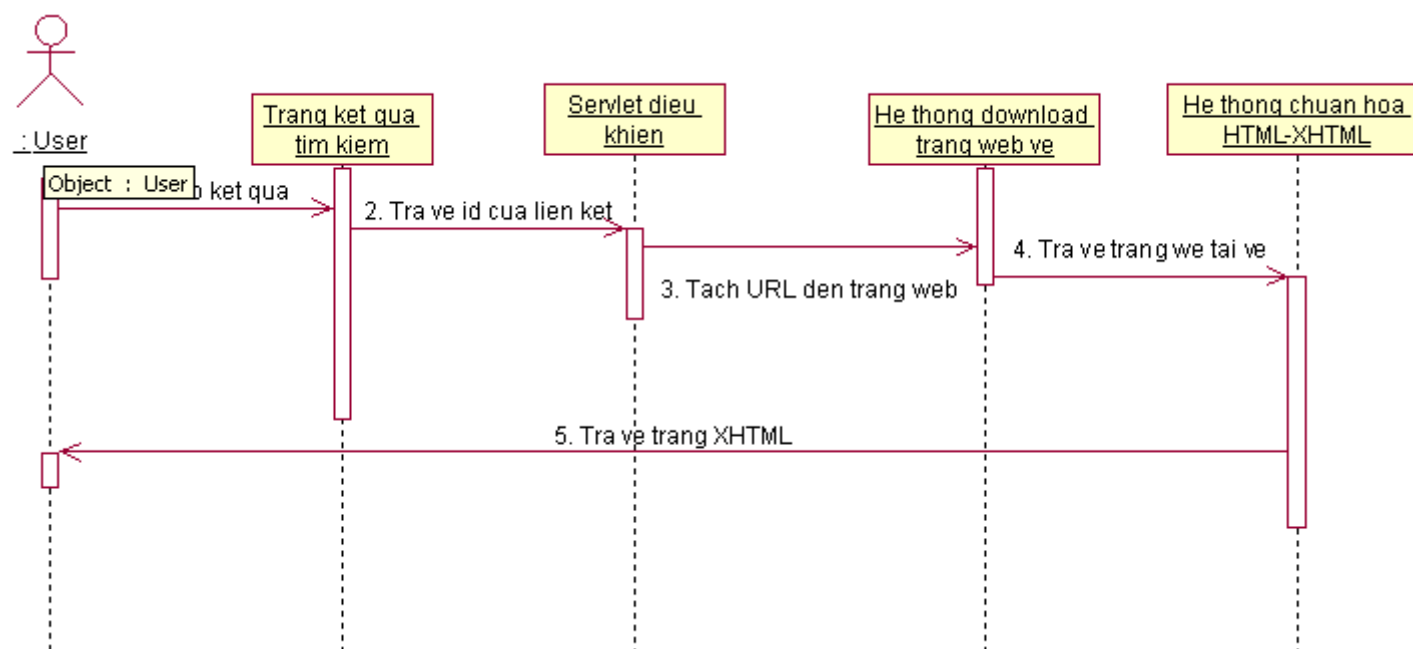
Các bước thực hiện:



Hình 5.8-2: Quá trình chuyển đổi trang web

- Nhận liên kết (URL) gửi về khi người dùng chọn một liên kết trong trang wap kết quả tìm kiếm trên trình duyệt wap của thiết bị di động.
- Tải trang web về máy: chỉ tải trang HTML và một số ít hình ảnh (bỏ qua các thông tin dưới dạng flash, movie, animation...)
- Chuẩn hóa trang HTML tải về thành trang WAP tương ứng: đảm bảo các ràng buộc của chuẩn XHTML
- Gửi trang WAP kết quả về cho người dùng

5.8.3. Mô hình sequence



Hình 5.8-3: Mô hình sequence của module chuyển đổi trang web

Chương 6. CÀI ĐẶT

Hệ thống được phát triển với các công cụ/môi trường

- Công cụ phân tích thiết kế: Rational Rose 2003
- Hệ điều hành: Windows XP Professional Service Pack 1
- Môi trường lập trình: Borland JbuilderX
- Môi trường thử nghiệm: Bộ giả lập của Nokia bao gồm WAP gateway và bộ trình duyệt WAP
- Cơ sở dữ liệu: Oracle 9i
- Các công cụ đồ họa hỗ trợ: Microsoft Frontpage 2003, Adobe Photoshop CS, Adobe Illustrator CS, ...

6.1. Hệ thống cơ sở dữ liệu chỉ mục

Sau khi tạo service phục vụ cho việc lập chỉ mục trên Oracle ta cần phải đăng ký service với ODBC (Xem phụ lục A.1.b “Hướng triển khai hệ thống”)

Các bảng trong cơ sở dữ liệu chỉ mục:

STARTPAGE (DOCID, URL, NGAYCAPNHAT)

Bảng chứa địa chỉ các trang web khởi đầu sẽ được download về để lập chỉ mục.

DADOWNLOAD (DOCID, URL, NGAYCAPNHAT)

Bảng chứa thông tin các trang đã được download về nhưng chưa được lập chỉ mục. Các trang đã được lập chỉ mục sẽ được xóa khỏi bảng này.

DALAPCHIMUC (DOCID, URL, TIEUDE, TRICHDAN, NGAYCAPNHAT)

Bảng chứa thông tin các trang đã được lập chỉ mục
THAMSO (TEN, GIATRI)

Bảng chứa các tham số: CurrentDocId, DownloadDocId

Các thao tác tạo bảng và tạo các procedure được thực thi tự động thông qua file script SCRIPT.SQL được đặt trong thư mục Working.

6.2. Module chuyển đổi trang HTML sang trang WAP

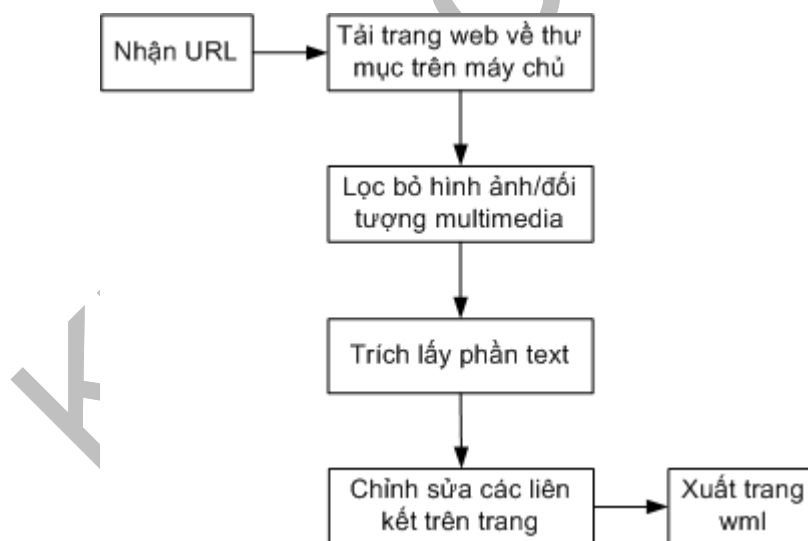
6.2.1. Các lớp cài đặt chính

Html2Wml: Servlet chuyển đổi trang HTML sang trang WML (chuẩn WAP 1.x)

Html2Xhtml: Servlet chuẩn hóa trang HTML sang trang XHTML (chuẩn WAP 2.0)

6.2.2. Phần chuyển đổi WAP 1.x – Servlet Html2Wml

- Đầu vào: Nhận 1 URL
- Đầu ra: trang wml theo chuẩn WAP 1.x
- Các bước tiến hành:



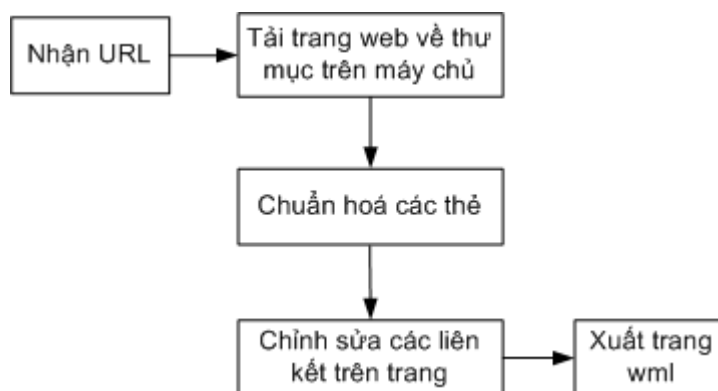
Hình 6.2-1: Các bước chuyển đổi WAP 1.x

1. Nhận URL
2. Tải trang web có URL vừa nhận về một thư mục trên máy chủ
3. Lọc bỏ hình ảnh và các đối tượng multimedia khác
4. Trích lấy phần text

5. Chỉnh sửa các liên kết trên trang (nếu có) : Chuyển hướng liên kết đi qua servlet chuyển đổi.
6. Xuất ra output trang wml

6.2.3. Phần chuyển đổi WAP 2.0 – Servlet Html2Xhtml

- Đầu vào: Nhận một địa chỉ URL của một trang web
- Đầu ra: trang XHTML theo chuẩn WAP 2.0
- Các bước tiến hành:



Hình 6.2-2: Các bước chuyển đổi WAP 2.0

1. Nhận URL
2. Tải trang web có URL vừa nhận về thư mục trên máy chủ
3. Chuẩn hóa các thẻ
4. Chỉnh sửa các liên kết trên trang (nếu có): Chuyển hướng liên kết đi qua servlet chuyển đổi.
5. Xuất ra output trang wml

6.3. Module nhận và phân tích query từ người dùng

6.3.1. Các lớp cài đặt chính

Search: Servlet quản lý việc gọi các hàm cung cấp giao diện từ lớp SearchUI để tạo nên trang tìm kiếm, trang kết quả, phân tích query của người dùng nhập vào và truy vấn cơ sở dữ liệu để lấy các kết quả tìm kiếm.

SearchUI: Cung cấp giao diện cho trang tìm kiếm

6.3.2. Phần xử lý detect trình duyệt

Để xác định khả năng hiển thị của trình duyệt trên thiết bị di động cần phải xây dựng phần kiểm tra phần HTTP header được trả về từ thiết bị di động với hai thuộc tính: accept và user-agent.

Đoạn mã kiểm tra:

```
String accept = request.getHeader("accept");
int userAgentClass = WML1_PHONE;
if (accept.indexOf("text/html") != -1){
    userAgentClass = accept.indexOf("application/vnd.wap.wmlc") != -1 ?
WAP2_PHONE : XHTML_BROWSER;
    if (userAgentClass == WAP2_PHONE &&
accept.indexOf("application/xhtml+xml") == -1){
        userAgentClass = WML1_PHONE;
    }
}
String userAgent = request.getHeader("user-agent");
if (userAgent.indexOf("Mozilla") != -1)
    userAgentClass = XHTML_BROWSER;
```

Sau khi xác định được khả năng của trình duyệt ta thiết lập Content Type cho phần response header:

```
response.setContentType(userAgentClass == WML1_PHONE ?
WML_CONTENT_TYPE : userAgentClass == WAP2_PHONE ?
XHTML_CONTENT_TYPE : HTML_CONTENT_TYPE);
response.setHeader("cach-control", "no-cache");
```

Ta lưu thông tin trình duyệt vào session người dùng

```
HttpSession session = request.getSession(true);
session.setAttribute("userAgentClass",
Integer.toString(userAgentClass));
```

Cuối cùng chuyển người dùng đến trang nội dung tương ứng:

```
if (userAgentClass == WML1_PHONE){
```

```

        response.sendRedirect("/www/wmlcontent");
    }else if (userAgentClass == WAP2_PHONE){
        response.sendRedirect("/www/xhtmlcontent");
    }else if (userAgentClass == XHTML_BROWSER){
        response.sendRedirect("/www/xhtmlcontent");
    }

```

6.3.3. Phân xử lý query

Nhận query của người dùng nhập vào

```
String query = request.getParameter("query");
```

Quá trình phân tích query

```

Vector v = manager.parseQuery(query);
Vector vword = new Vector(); //Vector chứa các stopwords
int size = v.size();
String word;
Word = (String)v.elementAt(i);
for (i=0;i<size;i++){
    word = (String)v.elementAt(i);
    if (!manager.isStopWord(word))
        vword.addElement(word);
}

```

Mỗi một word là một từ có nghĩa được phân tích từ chuỗi query

6.3.4. Phần truy vấn cơ sở dữ liệu tìm kiếm kết quả

```

int nword = 0;
StreamInverseFile streams[];
nword = vword.size(); //so luong stopwords trong query
if (nword>0)
{
    streams = new StreamInverseFile[nword];
    for (i=0;i<nword;i++)
        streams[i] =
manager.getStreamResult((String)vword.elementAt(i));
    JoinStream join = new JoinStream(streams);
}

```

```

        SortedResult sorted = new
SortedResult(join,nword);
        sorted.skip(start);
        int nItem,nPage;
        n=0;
        nItem = sorted.arr.size(); //nItem: so tai lieu
tim thay
    }

```

Mảng sorted chứa các tài liệu được tìm thấy, để lấy một tài liệu

```

Item item;
item=sorted.getItem ();

```

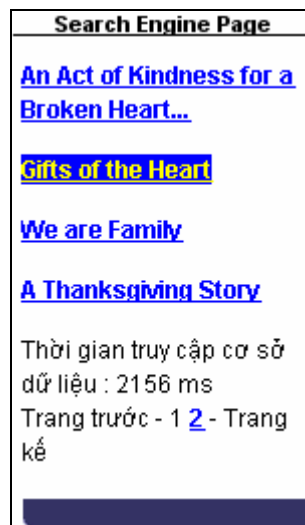
6.3.5. Giao diện tìm kiếm trên thiết bị di động

6.3.5.1. Giao diện cho trình duyệt hỗ trợ WAP 2.0

Trang chủ



Kết quả tìm kiếm



Trang tùy chọn kết quả hiển thị



6.3.5.2. Giao diện cho trình duyệt hỗ trợ WAP 1.x

Trang chủ



Trang kết quả



Chương 7. THỬ NGHIỆM

7.1. Thử nghiệm trên các bộ giả lập

Môi trường thử nghiệm

- Hệ điều hành: Windows XP Professional Service Pack 1
- Cơ sở dữ liệu: Oracle 9i
- Web server: Apache Tomcat 4.06 được chạy trên máy local
- Bộ giả lập Nokia: WAP gateway và bộ trình duyệt WAP

Kết quả thử nghiệm

- Máy tìm kiếm:
 - ❖ Có khả năng download các trang web trên mạng cục bộ cũng như trên Internet
 - ❖ Quá trình lập chỉ mục các trang web tải về tương đối nhanh.
 - ❖ Cung cấp kết quả tìm kiếm tương đối chính xác
 - ❖ Giao diện quản trị dễ hiểu, tiện dụng (Xem phần phụ lục B “Quản trị hệ thống Mobile Search Engine”)
- Hệ thống giao diện nhận từ khóa tìm kiếm của người dùng
 - ❖ Hoạt động hiệu quả trên các dòng trình duyệt khác nhau (IE 6.0, FireFox, trình duyệt WML 1.x giả lập của Nokia, trình duyệt WAP 2.0 giả lập của Nokia, trình duyệt WAP giả lập của Erricsion)
 - ❖ Đơn giản, dễ sử dụng
 - ❖ Hiển thị kết quả rõ ràng cho người dùng
- Hệ thống chuyển đổi:
 - ❖ Hoạt động tương đối tốt khi chuyển từ HTML sang XHTML

7.2. Thử nghiệm trên môi trường thực tế

Môi trường thử nghiệm

- Hệ điều hành: Windows XP Professional Service Pack 1
- Cơ sở dữ liệu: Oracle 9i
- Web server: Apache Tomcat 4.06
- Điện thoại di động Nokia 6610, 6600, Motorola V3
- Hệ thống được đưa lên mạng Internet thông qua đường truyền ADSL do FPT cung cấp (Xem phần phụ lục A.4 “Đưa trang web lên Internet qua đường truyền ADSL – Self-hosting”)

Kết quả thực nghiệm:

- Hệ thống có thể được truy cập từ các thiết bị di động thật có tích hợp sẵn trình duyệt WAP
- Thao tác tìm kiếm tiện lợi, nhanh chóng, kết quả tương đối chính xác.

Chương 8. TỔNG KẾT

8.1. Kết quả đạt được

Sau khi thực hiện đề tài, chúng em đã thu được một số kết quả sau:

- Tìm hiểu được công nghệ WAP, công nghệ thông dụng phục vụ cho việc cung cấp, trao đổi thông tin trên Internet thông qua các thiết bị di động. Nội dung tìm hiểu bao gồm tình hình phát triển của WAP, kiến trúc, đặc điểm kỹ thuật, đặc điểm triển khai, những hạn chế và tương lai phát triển của WAP. Bên cạnh đó nhóm còn tìm hiểu thêm được các kiến thức về mạng thông tin di động và mạng không dây.
- Tìm hiểu được mô hình hoạt động của một máy tìm kiếm – search engine và triển khai, áp dụng máy tìm kiếm vào mô hình ứng dụng của đề tài.
- Tìm hiểu được cách thức chuẩn hóa một tài liệu HTML theo chuẩn của XML (XHTML)
- Có được kiến thức triển khai một hệ thống sử dụng cơ sở dữ liệu Oracle 9i.
- Phát triển khả năng lập trình sử dụng ngôn ngữ Java và môi trường tích hợp Borland JbuilderX
- Xây dựng được hệ thống tìm kiếm phục vụ cho các thiết bị di động.
- Tìm hiểu và vận hành tốt các web server: Apache 4.1, Apache Tomcat 4.0.6, Resin
- Triển khai được hệ thống trên đường truyền ADSL, vốn có khó khăn do địa chỉ IP do nhà cung cấp phân phối là địa chỉ IP động – dynamic IP. Xem thêm phần phụ lục A.4. “Đưa trang web lên Internet qua đường truyền ADSL – Self-hosting”.

8.2. Hạn chế

Mặc dù nhóm đã có một thời gian học tập, làm việc tích cực nhưng do thời gian có hạn và không có điều kiện tốt để thử nghiệm nên vẫn còn tồn tại những hạn chế:

- Chưa tiến hành thực nghiệm được các hệ thống thực tế nhất là đối với kiến trúc hoạt động của WAP, WAP gateway và các hướng bảo mật của WAP.
- Máy tìm kiếm chưa hỗ trợ được tiếng Việt. Chưa tối ưu hóa được mô hình hoạt động của máy tìm kiếm để phục vụ riêng các nội dung phù hợp cho thiết bị di động.
- Hệ thống chuyển đổi hoạt động chưa hiệu quả nhất là các trường hợp trang web có frame, có nhiều table lồng vào nhau,...

PHẦN IV.
ĐÁNH GIÁ – HƯỚNG PHÁT TRIỂN

Với những kết quả đạt được và những hạn chế còn tồn tại, đề tài có thể mở rộng theo các hướng sau:

- Phát triển máy tìm kiếm để có thể hỗ trợ tất cả các bảng mã tiếng Việt.
- Tối ưu hóa mô hình hoạt động của máy tìm kiếm để phục vụ tốt hơn cho thiết bị di động, ví dụ: cho phép chọn trọng số của từ khóa trong các kết quả tìm được, cho phép người dùng đánh giá độ phù hợp của kết quả để phục vụ cho yêu cầu tìm kiếm sau đó...
- Phát triển bộ chuyển đổi để có thể tiến hành phân tích và dàn trang lại trang web: sắp xếp các table, loại bỏ các frame, resize hình ảnh, chuyển đổi định dạng hình ảnh để có thể hiển thị tốt trên các thiết bị di động, loại bỏ các file mà hầu hết các trình duyệt WAP chưa hỗ trợ: Flash,...

TÀI LIỆU THAM KHẢO

- [1] Sergey Brin and Lawrence Page, The anatomy of a large-scale hypertextual web search engine, Computer Science Department, Stanford University.
- [2] Huỳnh Thụy Bảo Trân, Nghiên cứu một số mô hình Và Xây Dựng Thử Nghiệm Một máy tìm kiếm Tiếng Việt, Luận Án Thạc Sĩ Khoa Học, 2002.
- [3] <http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/MetaSearch.html>
- [4] http://www.lookoff.com/tactics/engines_basics.php3#metasearch
- [5] <http://www.cs.utexas.edu/users/mooney/ir-course/>
- [6] <http://www.searchtools.com>
- [7] <http://www.ovum.com>
- [8] WAP 2.0 with XHTML over TCP/IP
- [9] Professional WAP – FlyHeart.com
- [10] Next Generation Mobile Browsing White Paper v1.0
- [11] WML to XHTML Migration v2.1
- [12] XHTML Guidelines v1.2
- [13] <http://www.google.com.vn>
- [14] NMIT4.1 User Guide v1.0
- [15] Nokia Mobile Browser Simulator 4.0 User's Guide
- [16] Nokia WAP Gateway Simulator 4.0 User's Guide

PHỤ LỤC A
HƯỚNG TRIỂN KHAI HỆ THỐNG

1. Cơ sở dữ liệu

a) Cài đặt

Cài đặt hệ quản trị cơ sở dữ liệu Oracle 9i

Tạo mới một user với hệ thống phân quyền sau:

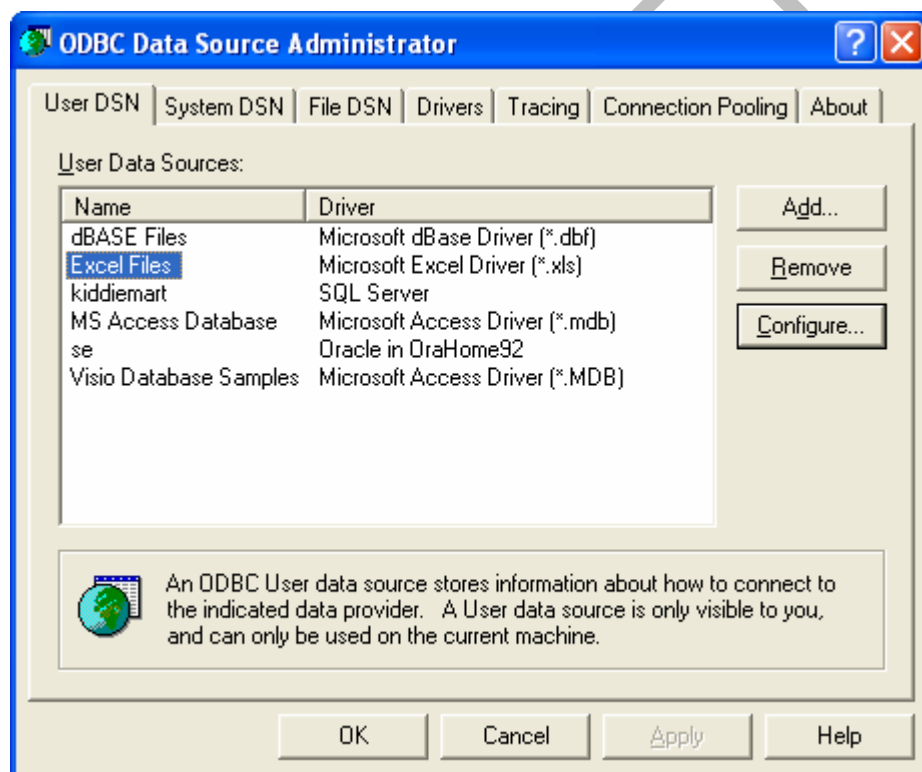
Role: CONNECT, RESOURCE

System: UNLIMITED TABLESPACE

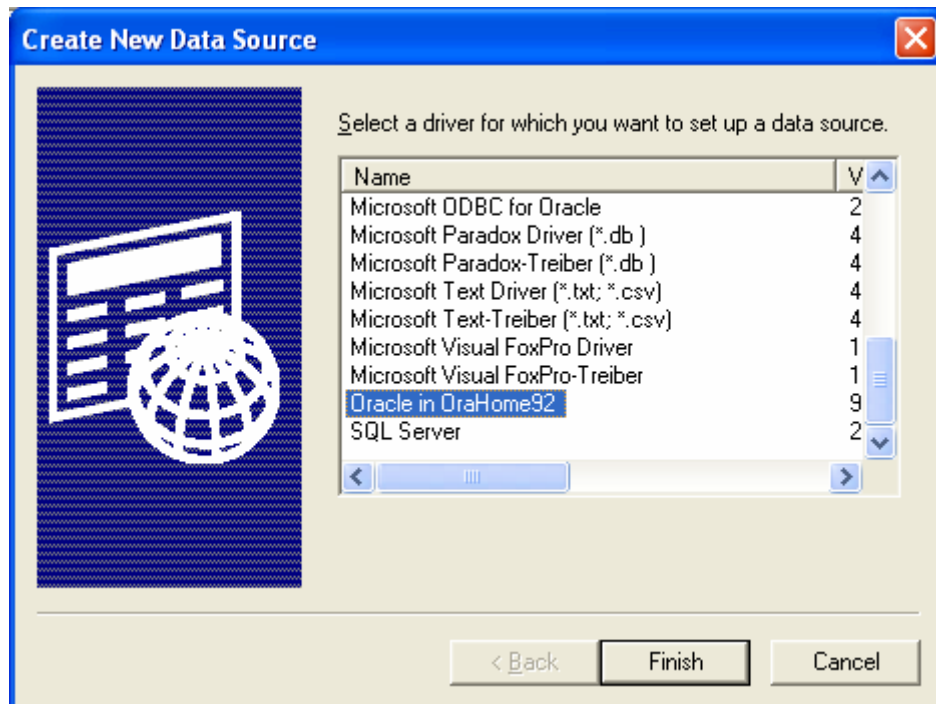
b) Đăng ký cơ sở dữ liệu với hệ thống ODBC

Tiến hành theo các bước sau:

Chọn Control Panel>Administrative Tools>Data Source (ODBC)



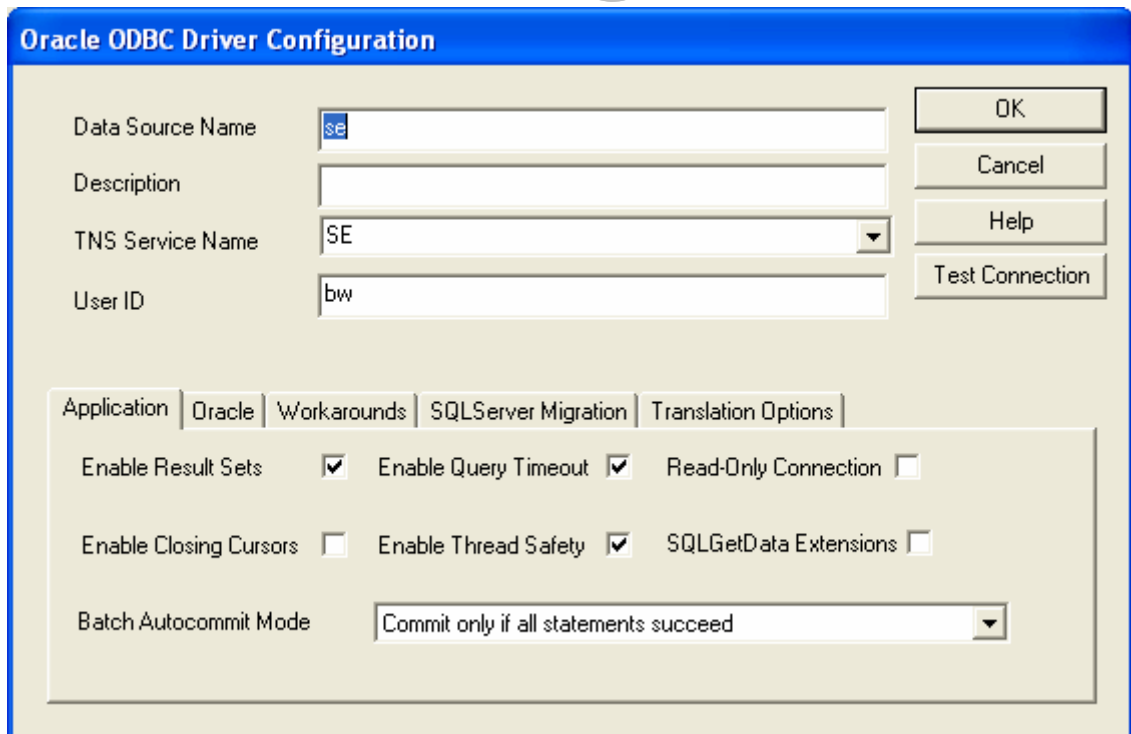
Chọn Add... để thêm mới một cơ sở dữ liệu. Xuất hiện hộp thoại Create New Data Source:



Chọn Oracle in OraHome92

Chọn Finish

Xuất hiện hộp thoại Oracle ODBC Driver Configuration



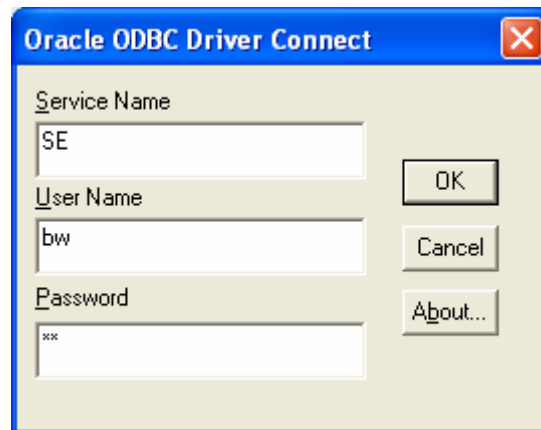
Nhập tên cơ sở dữ liệu (Data Source Name)

Nhập tên của dịch vụ cơ sở dữ liệu (TNS Service Name)

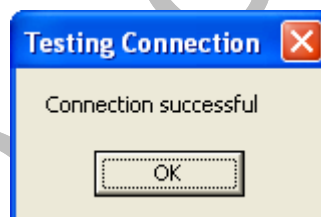
Nhập tên user (User ID)

Đến đây ta đã hoàn thành đăng ký cơ sở dữ liệu với hệ thống ODBC.

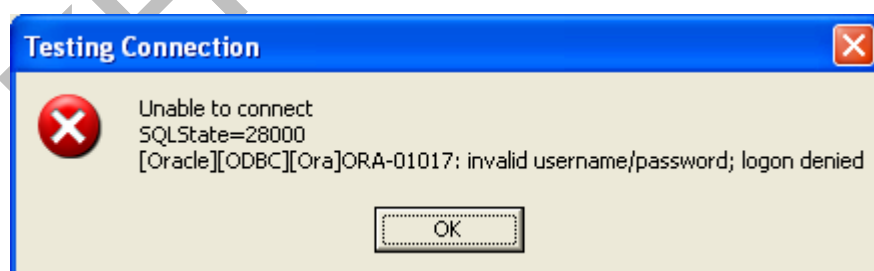
Ta có thể kiểm tra kết nối đến dịch vụ bằng cách chọn Test Connection. Khi đó sẽ xuất hiện hộp thoại Oracle ODBC Driver Connect



Nhập password tương ứng với người dùng và chọn OK. Nếu kết nối thành công sẽ xuất hiện hộp thoại



Nếu bị lỗi trong khi kết nối sẽ thông báo lỗi. Ví dụ khi nhập sai password



Đến đây thì cơ sở dữ liệu đã có thể phục vụ cho nhu cầu lập chỉ mục của hệ thống máy tìm kiếm.

Các bảng và các procedure được thực thi tự động khi tiến hành cấu hình máy tìm kiếm thông qua giao diện web. Nội dung của các script được lưu trong file SCRIPT.SQL trong thư mục SearchEngine/www/Working

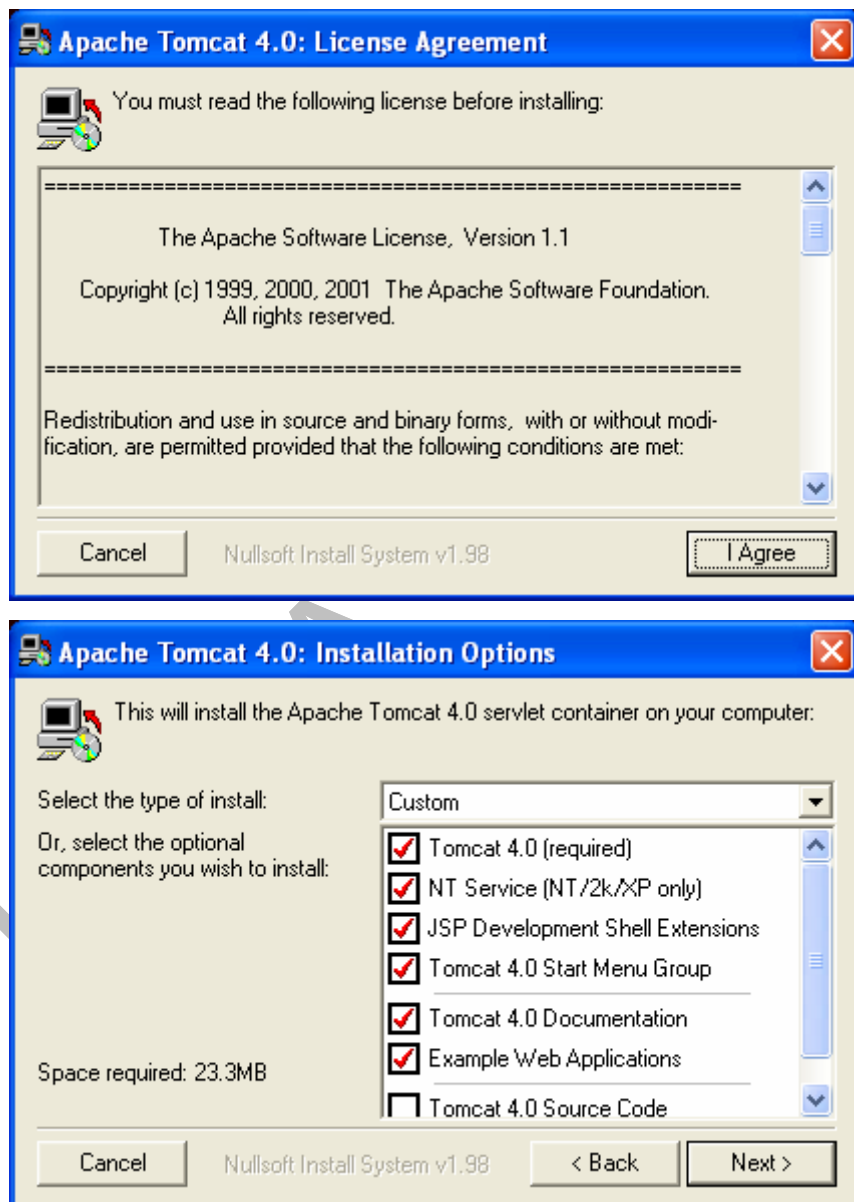
2. Web server

Hệ thống được triển khai trên Web server Apache Tomcat 4.0.6

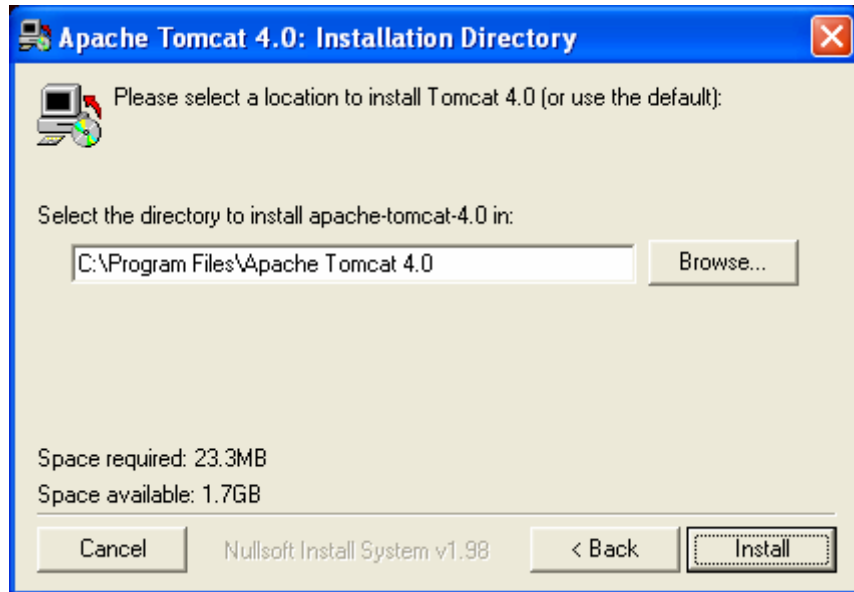
a) Cài đặt

Gói cài đặt có thể download từ trang web <http://jakarta.apache.org/tomcat>

Quá trình cài đặt được tiến hành tuần tự qua các bước



Chú ý: Chọn mục NT Service (NT/2k/XP only)



Chọn Install để tiến hành cài đặt.

b) Thiết lập server

Do hệ thống Oracle sử dụng web server Apache và được thiết lập để nhận kết nối ở cổng 8080 trong khi mặc định web server Apache Tomcat 4.0.6 cũng được thiết lập để nhận kết nối qua cổng 8080, ta cần phải thiết lập đổi cổng nhận kết nối của Apache Tomcat 4.0.6 sang cổng khác (ví dụ 8085) để tránh xung đột.

Để thiết lập đổi cổng nhận kết nối của Apache Tomcat 4.0.6 ta tiến hành chỉnh sửa trong file C:\Program Files\Apache Tomcat 4.0\conf\server.xml (có thể truy cập file thông qua đường dẫn Start Menu>All Programs>Apache Tomcat 4.0>Configuration>Edit Server Configuration. Ta đổi giá trị 8080 thành 8085 tại dòng sau:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<Connector
className="org.apache.catalina.connector.http.HttpConnector"
    port="8080" minProcessors="5" maxProcessors="75"
    enableLookups="true" redirectPort="8443"
    acceptCount="10" debug="0"
connectionTimeout="60000"/>
```

Kết quả sau khi thay đổi:

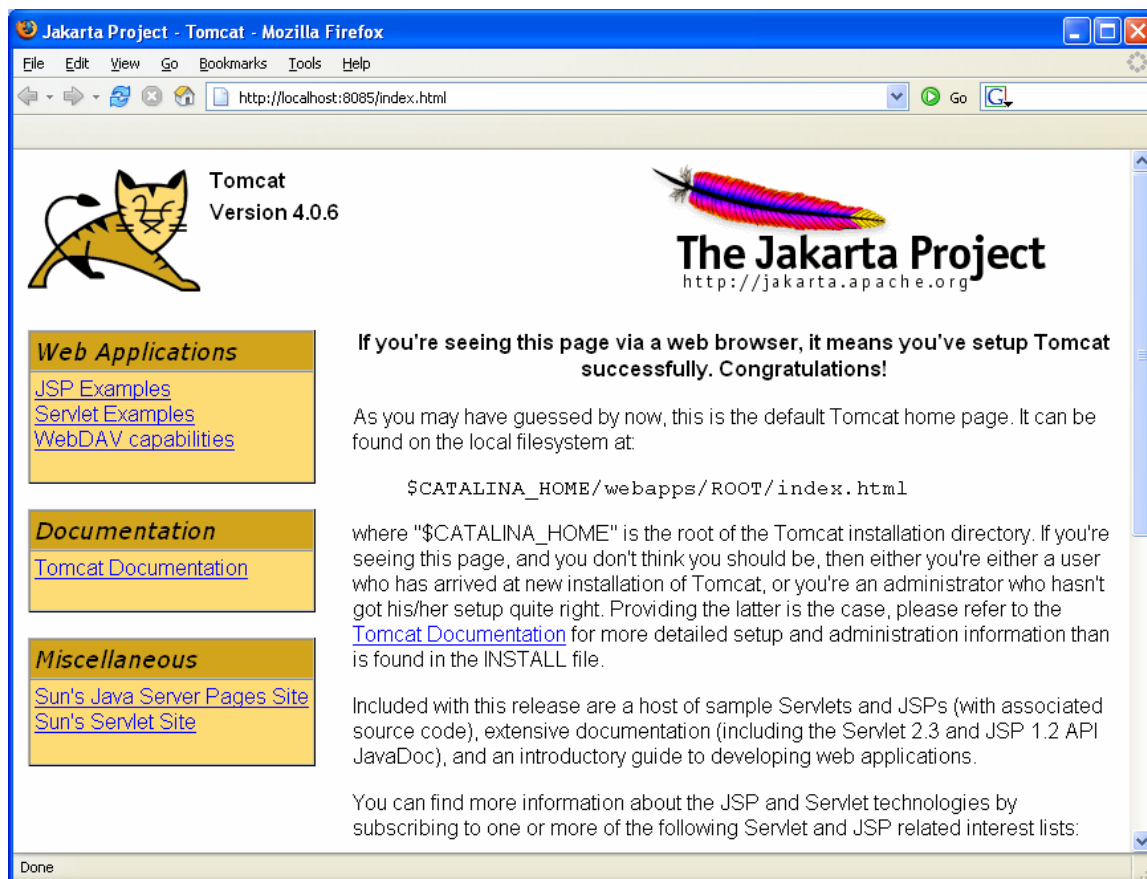
```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8085 -->
    <Connector
className="org.apache.catalina.connector.http.HttpConnector"
    port="8085" minProcessors="5" maxProcessors="75"
    enableLookups="true" redirectPort="8443"
    acceptCount="10" debug="0"
connectionTimeout="60000"/>
```

Apache Tomcat 4.0.6 đã được tự động thiết lập để có thể quản lý các file có định dạng của WAP (wml, wml script, wbmp, xhtml,...) Ta có thể thay đổi thiết lập trong file: Start Menu>All Programs>Apache Tomcat 4.0>Configuration>Edit Webapp Defaults

Thực thi Apache Tomcat 4.0.6

Chọn Start Menu>All Programs>Apache Tomcat 4.0>**Start Tomcat** để chạy Apache Tomcat 4.0.6 (**Stop Tomcat** để dừng)

Mở trình duyệt web (IE, Mozilla, Firefox,...) truy cập đến địa chỉ <http://localhost:8085>. Nếu server được thiết lập chính xác thì sẽ xuất hiện trang homepage của Tomcat:



Muốn triển khai các ứng dụng web lên Apache Tomcat 4.0.6 thì ta copy toàn bộ ứng dụng đặt vào thư mục `ROOT_DIRECTORY/ Apache Tomcat 4.0.6/webapps`. Trong đó `ROOT_DIRECTORY` là thư mục cài đặt Apache Tomcat 4.0.6. Ví dụ: `C:/Program Files`

3. Kết hợp nối hệ thống thông qua bộ giả lập trình duyệt wap của Nokia

Xem phần phụ lục C “Bộ toolkit của Nokia”

4. Đưa trang Web lên internet qua đường truyền ADSL (self-hosting)

a) Đặc điểm

Khi ta tiến hành kết nối Internet (ADSL, dial-up,...) thông qua một nhà cung cấp dịch vụ ta sẽ được cung cấp một địa chỉ IP động (Dynamic IP address - địa chỉ này có thể thay đổi trong quá trình sử dụng hoặc thay đổi trong những lần kết nối khác nhau tùy theo nhà cung cấp dịch vụ. Xin đừng nhầm lẫn với địa IP động – Mobile IP address).

Với kỹ thuật này ta có thể thiết lập để có thể truy cập từ xa vào một máy thông qua đường truyền Internet. Qua đó ta có thể cung cấp các dịch vụ (Web, FTP, Telnet,...) mà không cần thuê hosting của một nhà cung cấp nào cả.

Với kỹ thuật kết nối này hệ thống cục bộ có máy cung cấp dịch vụ vẫn được đảm bảo an ninh tốt.

b) Nguyên tắc

Đối với địa chỉ IP động, ta có thể đăng ký với một nhà cung cấp dịch vụ DNS động (Dynamic DNS – Tên miền động) để được cung cấp một tên miền và sẽ tự động cập nhật địa chỉ IP khi có sự thay đổi.

Có thể kể tên một số nhà cung cấp dịch vụ DNS động như: No-IP (<http://www.no-ip.com>), Dynamic DNS (<http://dyndns.org>) ...

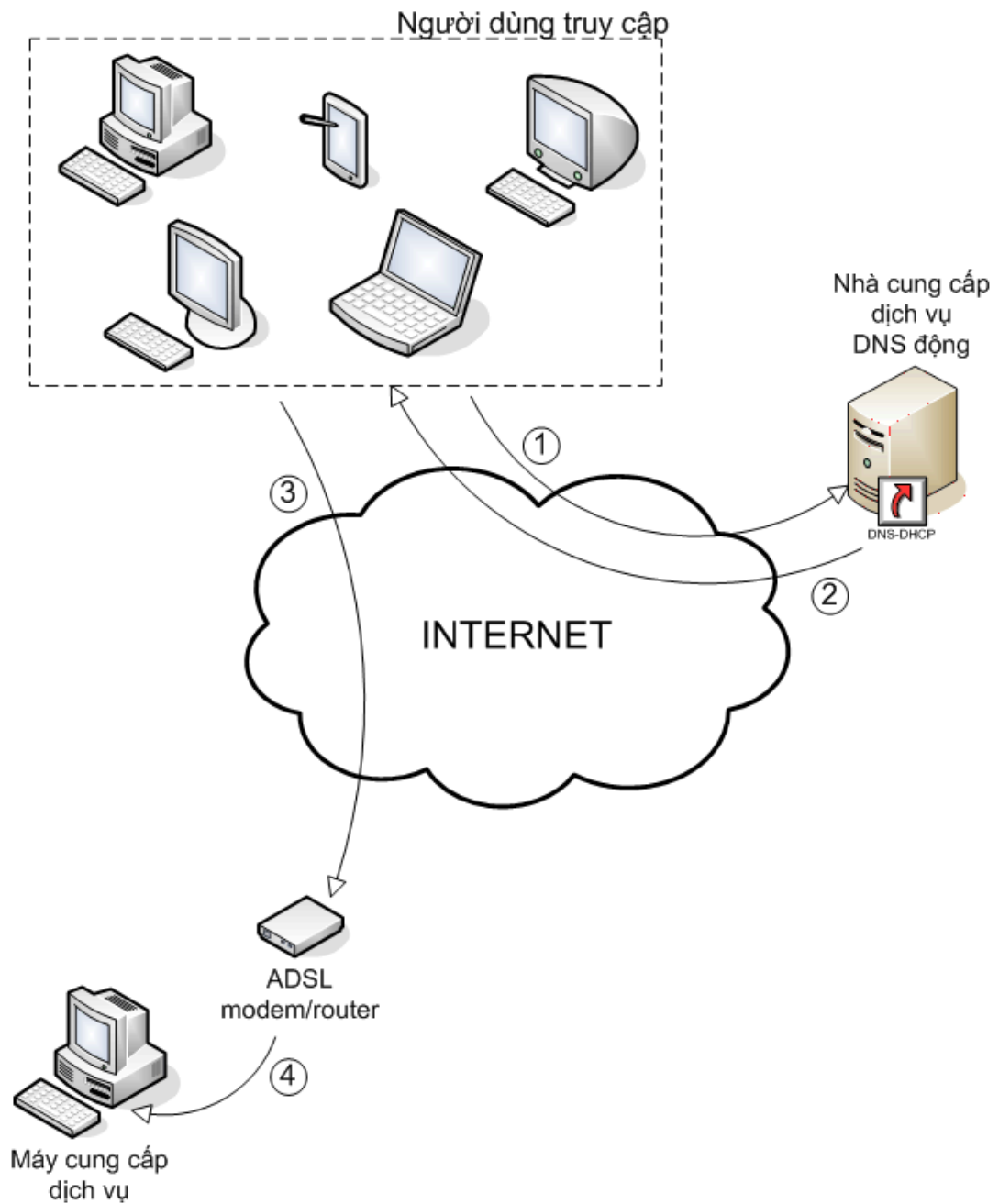
Cuối cùng cần config modem/router ADSL như sau

Unlock các dịch vụ cần sử dụng (HTTP, FTP, Telnet,...)

Thiết lập bảng NAT để router sẽ forward các kết nối đến máy cục bộ cung cấp dịch vụ thông qua cổng xác định

Khi đó ta đã có thể kết nối từ xa thông qua tên miền được cung cấp.

c) Mô hình hoạt động



Hình 8.2-1: Mô hình hoạt động quá trình cung cấp IP thông qua đường ADSL (self-hosting)

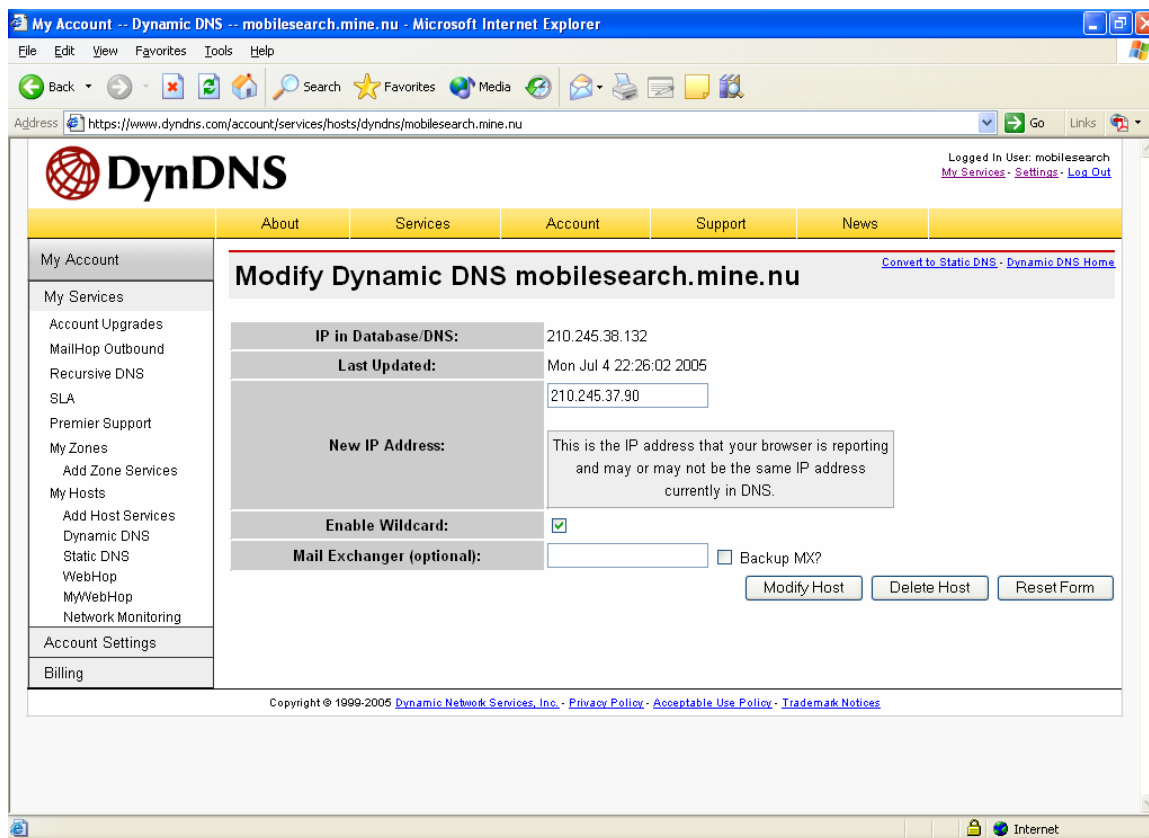
d) Trình tự kết nối từ bên ngoài

1. Máy kết nối thông qua tên miền (được cung cấp bởi dịch vụ DNS động) sẽ kết nối vào máy chủ dịch vụ DNS động
2. Máy chủ DNS động sẽ cung cấp IP hiện tại của máy local tương ứng với tên miền đã đăng ký
3. Máy kết nối sẽ truy cập vào ADSL modem/router thông qua địa chỉ IP được cung cấp
4. ADSL modem/router sẽ forward yêu cầu vào máy local cung cấp dịch vụ

e) Thiết lập

Hệ thống được triển khai thông qua đường truyền ADSL của FPT, với ADSL modem/router ZyXEL được hỗ trợ tự động bởi nhà cung cấp dịch vụ DNS động DynDNS (<http://dyndns.org>)

Trang web nhà cung cấp dịch vụ DynDNS:



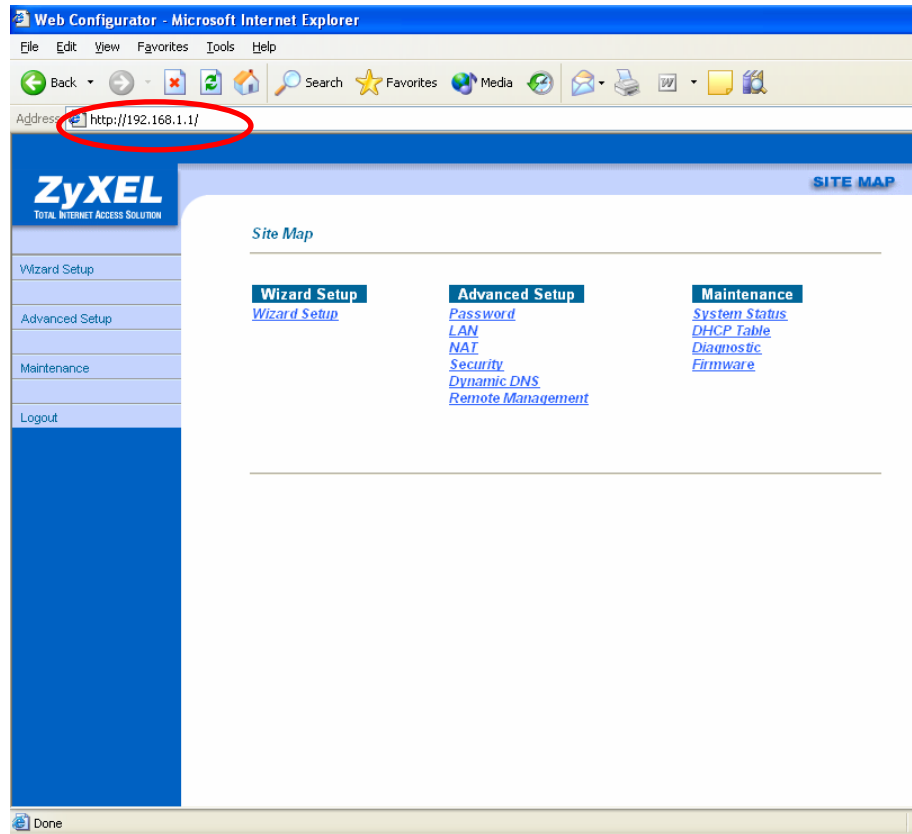
Hình 8.2-2: Trang web nhà cung cấp dịch vụ DynDNS

Cấu hình cho ADSL modem/router thông qua giao diện web.

ADSL modem/router có địa chỉ thường là 192.168.1.1 và cũng là default gateway của hệ thống cục bộ.

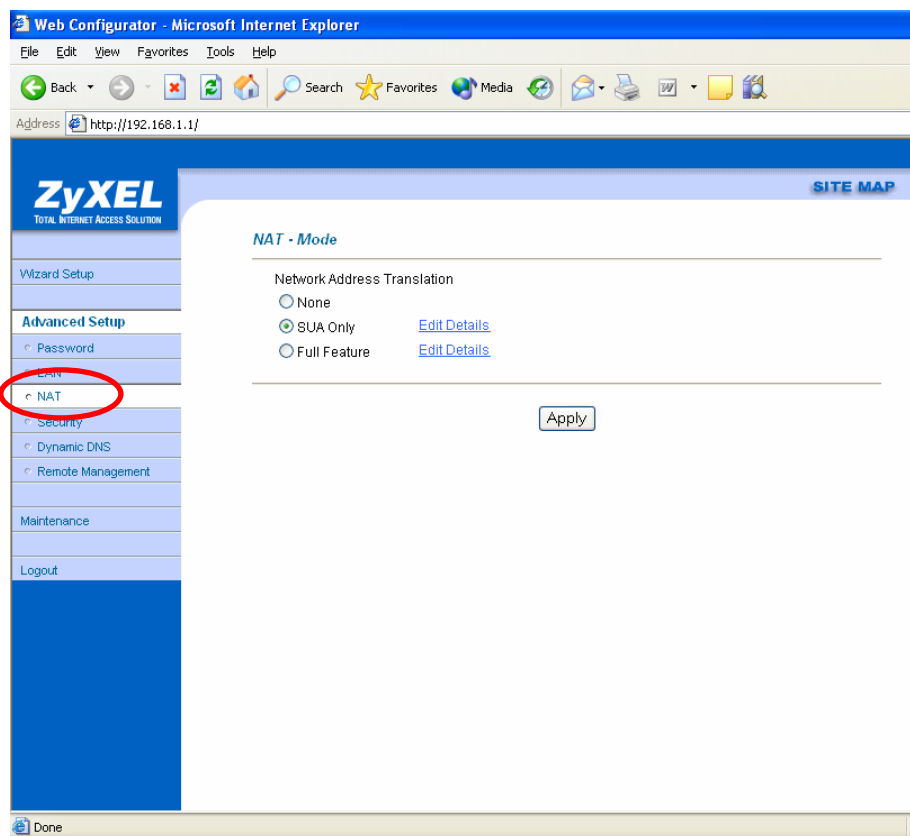
Kết nối vào ADSL modem/router – phải cung cấp tên truy cập và mật khẩu.

Giao diện sau khi kết nối thành công.



Hình 8.2-3: Giao diện kết nối thành công vào ADSL modem/router

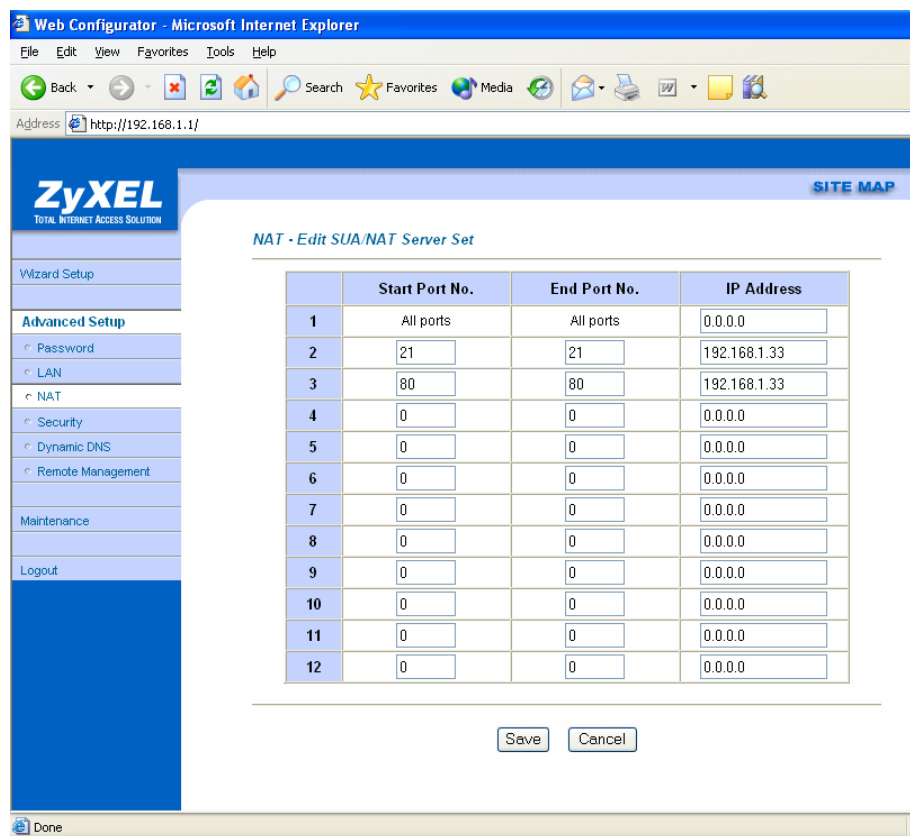
Chọn liên kết NAT để cấu hình bảng NAT của ADSL modem/router



Hình 8.2-4: Chọn chức năng cấu hình bảng NAT

Chọn tùy chọn “SUA Only” và click vào liên kết “Edit Details”

Xuất hiện bảng NAT của ADSL modem/router



Hình 8.2-5: Bảng NAT

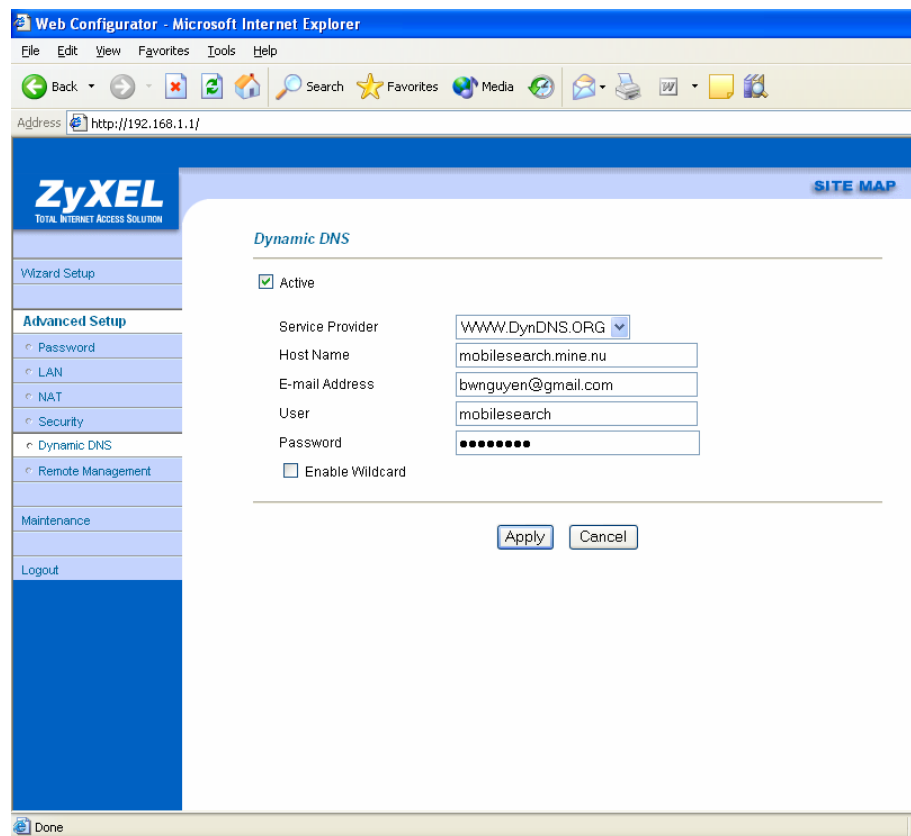
Start Port No. Cổng lắng nghe kết nối từ Internet (80 cho Web/HTTP, 21 cho FTP, ...)

End Port No. Cổng lắng nghe kết nối trên máy cục bộ cung cấp dịch vụ

IP Address Địa chỉ IP cục bộ của máy cung cấp dịch vụ

Chọn Save và sau đó chọn Apply để áp dụng bảng NAT mới cho ADSL modem/router

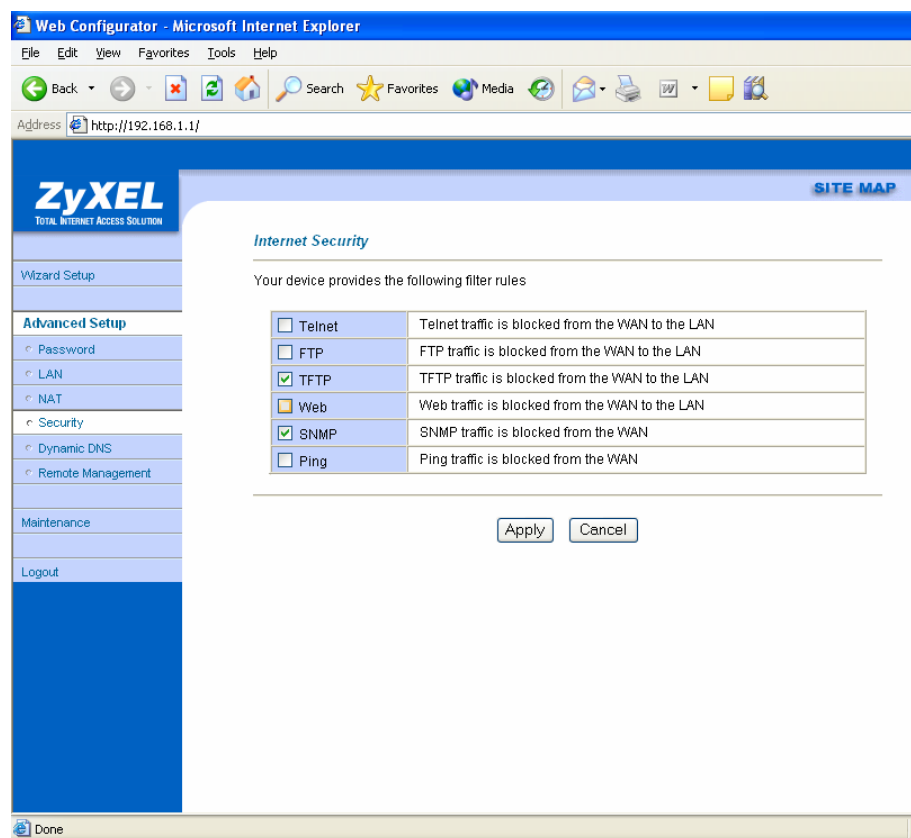
Đăng ký dịch vụ DNS động tự động với nhà cung cấp dịch vụ DynDNS



Hình 8.2-6: Đăng ký dịch vụ DNS động với nhà cung cấp dịch vụ

Chọn Apply để áp dụng thiết lập cho ADSL modem/router

Tuy nhiên vì lý do an ninh mặc định ADSL modem/router sẽ lock các cổng để tránh kết nối không cần thiết từ phía bên ngoài. Do đó ta cần unlock dịch vụ cần thiết để có thể kết nối từ Internet:

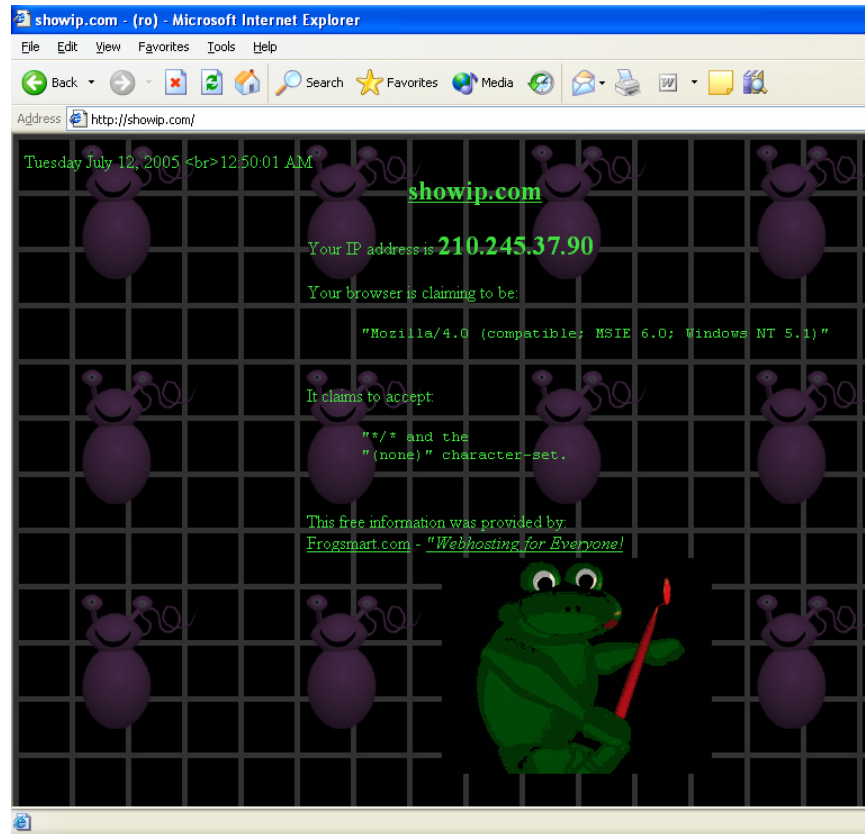


Hình 8.2-7: Thực hiện unlock các dịch vụ cần thiết

Đến đây ta đã hoàn thành các bước cấu hình để có thể truy cập từ xa vào hệ thống cục bộ thông qua đường truyền Internet ADSL.

Đối với các ADSL modem/router khác (không được hỗ trợ bởi DynDNS) ta có thể đăng ký dịch vụ DNS động với nhà cung cấp No-IP (<http://no-ip.com>) và tải về phần mềm được cung cấp miễn phí No-ip AutoUpdate. Phần mềm này sẽ tự động thông báo cho No-IP khi có sự thay đổi IP của hệ thống cục bộ. Các thiết lập trên bảng NAT của ADSL modem/router cũng tương tự như trên.

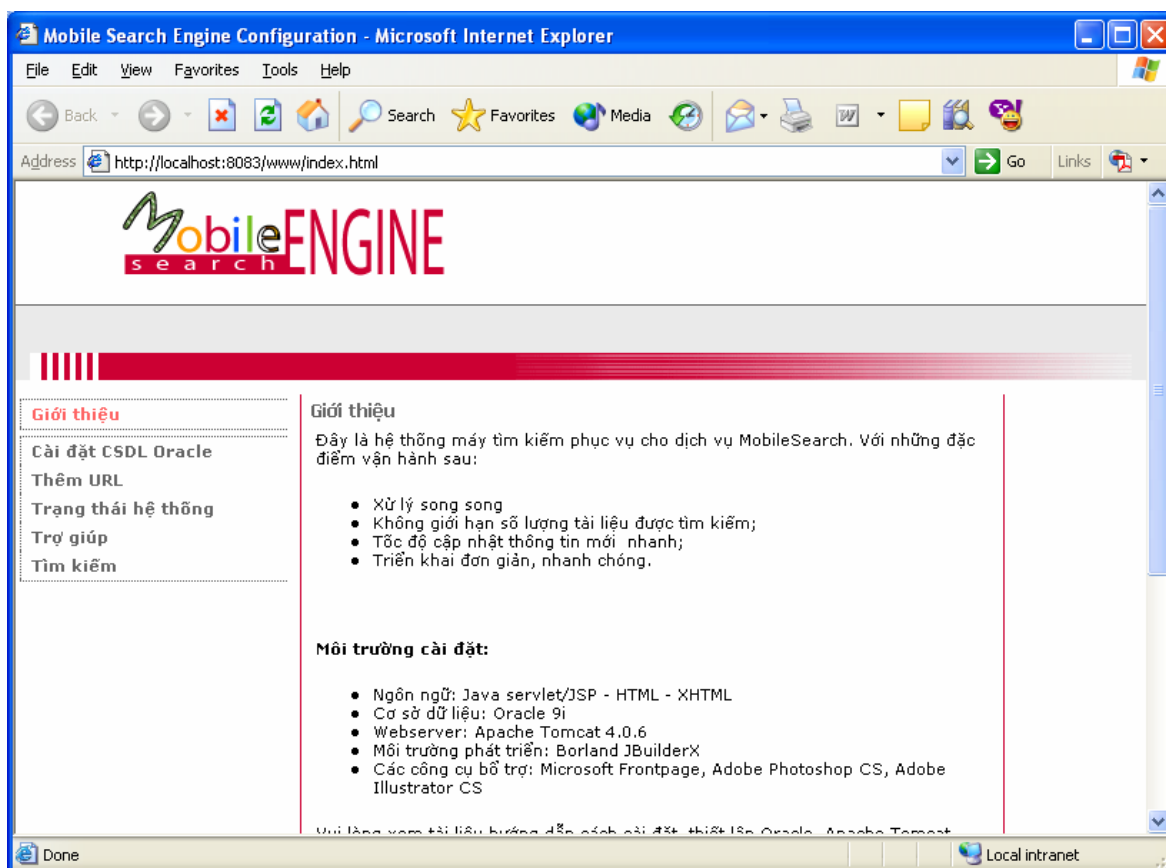
Ngoài ra ta có thể kiểm tra IP hiện hành của hệ thống local (nhìn từ phía Internet) bằng cách truy cập trang web <http://showip.com>



Hình 8.2-8: Trang showip.com

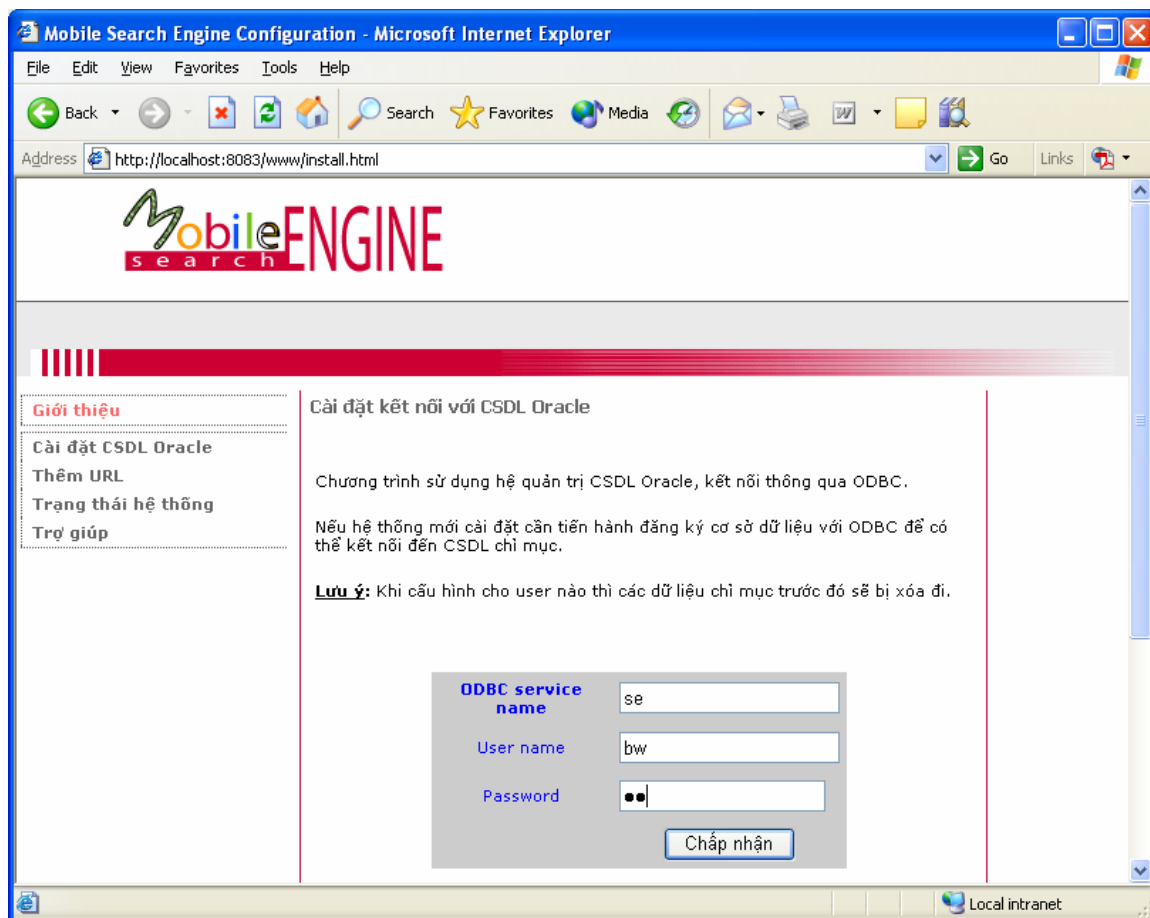
PHỤ LỤC B
QUẢN TRỊ HỆ THỐNG MOBILE SEARCH ENGINE

1. Trang chủ



a) Cài đặt thiết lập cơ sở dữ liệu Oracle cho hệ thống

Chọn “Cài đặt CSDL Oracle”



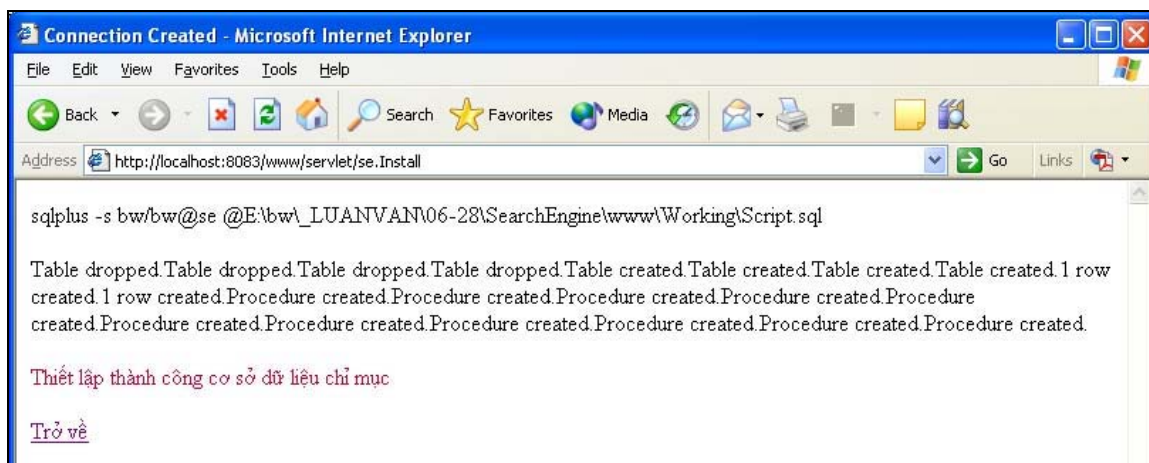
Nhập tên dịch vụ đã đăng kí với ODBC trước đó (Xem phụ lục A.1.b “Đăng ký cơ sở dữ liệu với ODBC”)

Cung cấp tên user và mật khẩu đăng nhập

Lưu ý: Khi cấu hình cơ sở dữ liệu trên một user, nếu hệ thống trước đó đã có dữ liệu thì tất cả các dữ liệu đó sẽ bị xóa hết.

Khi chọn chấp nhận hệ thống sẽ tự động thực thi file SCRIPT.SQL để tạo các bảng và các procedure

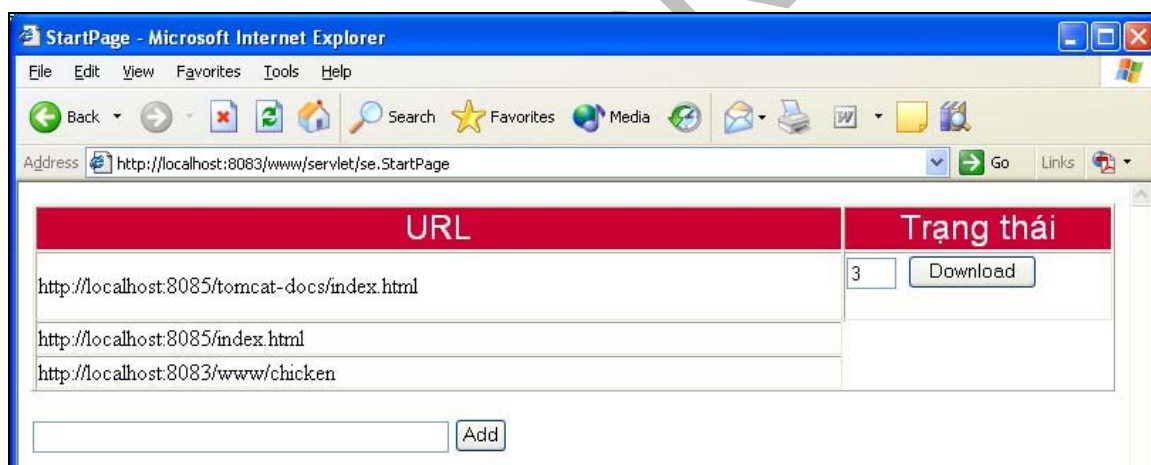
Nếu thành công sẽ hiện ra thông báo



Sau đó quay trở về trang chủ để tiến hành nhập địa chỉ URL các trang web cần lập chỉ mục.

b) Thêm URL và download các trang web

Chọn liên kết “Thêm URL”



Ở ô trạng thái nhập độ sâu của liên kết khi download

Chọn “Download” để tiến hành download

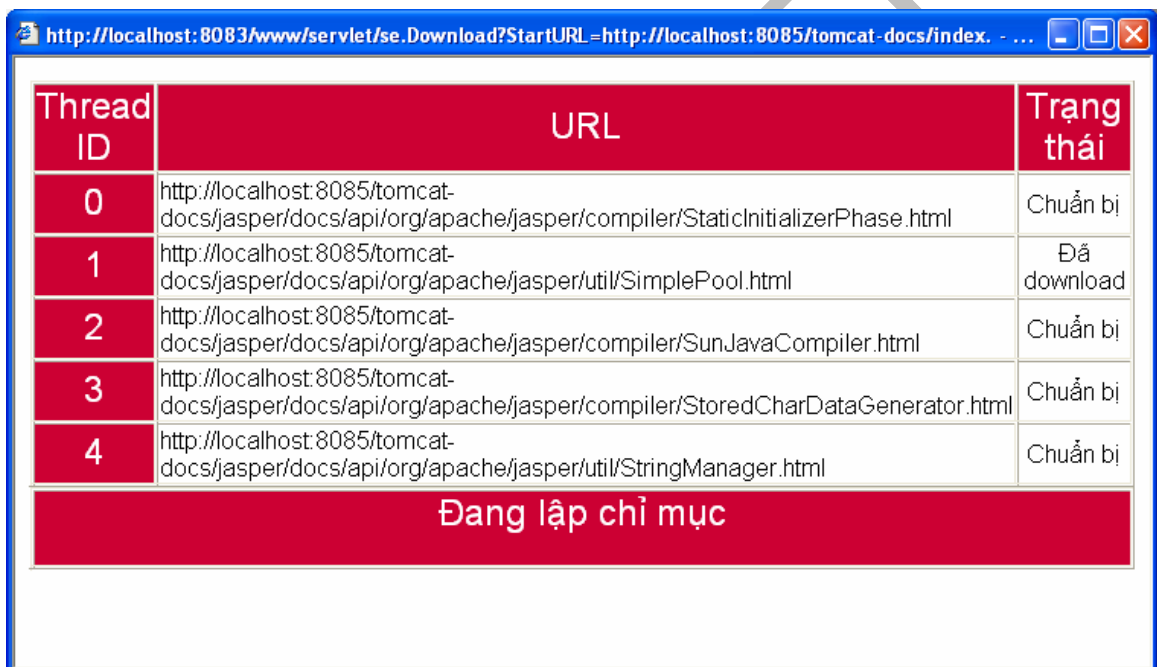
Xuất hiện màn hình yêu cầu xác nhận việc download



Chọn “Trở lại” để quay về màn hình nhập URL

Chọn “Download” để chấp nhận download các trang web từ địa chỉ URL nhập vào

Khi đó hệ thống sẽ kích hoạt các tiến trình con để download các trang web



Sau khi download hết các trang web, sẽ xuất hiện thông báo thời gian download, số trang đã download cùng với tổng dung lượng các trang, cuối cùng là số liên kết bị hỏng.

Trong khi đó hệ thống vẫn tiếp tục lập chỉ mục các trang web được download về

http://localhost:8083/www/servlet/se.Download?StartURL=http://localhost:8085/tomcat-docs/index. - ...

Quá trình download đã hoàn tất		
Tổng thời gian download	30 giây	
Số file đã download	691	14412077 Byte
Số liên kết hỏng	3	

Đang lập chỉ mục

Sau khi kết thúc quá trình lập chỉ mục ta tắt cửa sổ đi

http://localhost:8083/www/servlet/se.Download?StartURL=http://localhost:8085/tomcat-docs/index. - ...

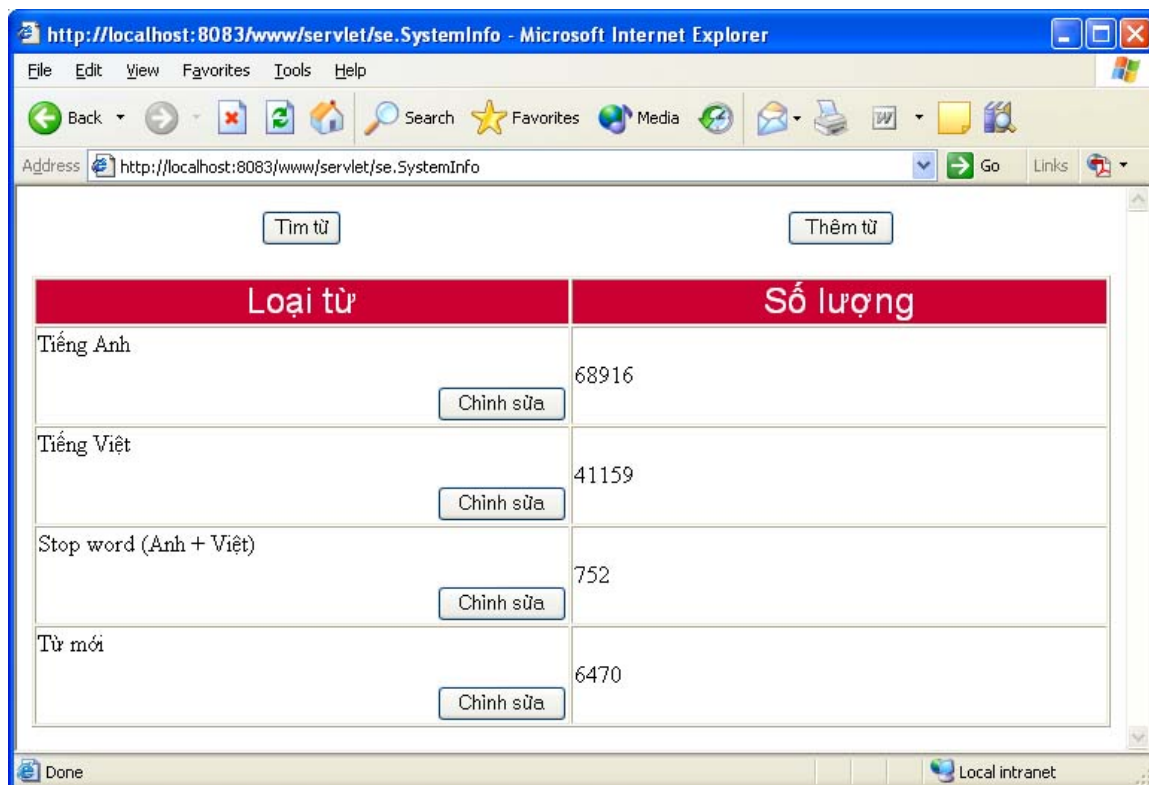
Quá trình download đã hoàn tất		
Tổng thời gian download	30 giây	
Số file đã download	691	14412077 Byte
Số liên kết hỏng	3	

Quá trình lập chỉ mục đã hoàn tất

Đóng cửa sổ

c) Kiểm tra thông tin tự diễn của hệ thống

Chọn liên kết “Trạng thái hệ thống”



Ta có thể xem danh sách các từ trong từ điển

étude	épée	émigré	élitist	élitism	élite	élan	éclat
éclair	é la mode	é la carte	zymotic	zymotechnics	zymosis	zymometer	zymology
zymase	zygote	zygospore	zygomorphous	zygomorphic	zygoma	zygomata	zulu
zucchini	zounds	zouave	zoroastrian	zoroastrianism	zoo	zootomy	zootomist
zootomic	zootherapy	zootechnic	zoot suit	zoospore	zoopsychology	zoopsychologist	zoopsychologic
zoophyte	zoophagous	zoophaga	zoophagan	zoonomy	zoom	zoomorphism	zoomorphic
zoology	zoologist	zoological	zoolite	zoography	zoographer	zoogeography	zoogeographical
zoochemistry	zoochemical	zoobiology	zoobiological	zonked	zoning	zone	zoneless
zonate	zonal	zombie	zoic	zodiac	zodiacal	zloty	zither
zircon	zirconium	zip	zipper	zipfastener	zip-fastener	zion	zionist

[Trang đầu](#)
[Trang trước](#)
[Trang kế](#)
[Trang cuối](#)

Tìm từ

Thêm từ

Từ mới :

Từ loại : Tiếng Việt ▾

Thao tác : Thêm từ ▾

Chỉnh sửa từ

Từ mới : zoographer

Từ loại : Tiếng Việt ▾

Thao tác : Thêm từ ▾

PHỤ LỤC C
BỘ TOOLKIT CỦA NOKIA

1. Nokia Mobile Internet Toolkit v4.1

a) Giới thiệu

Nokia Mobile Internet Toolkit [14] (NMIT) bao gồm một tập hợp các trình soạn thảo dùng để tạo nên nhiều loại nội dung khác nhau trên môi trường Internet di động. NMIT cho phép hiển thị các nội dung này trên nhiều bộ SDK của điện thoại.

Những bộ SDK của điện thoại được cài đặt riêng lẻ. Khi khởi động, NMIT sẽ tự động dò tìm, những bộ SDK điện thoại được hỗ trợ sẽ được thêm vào bảng danh sách trong phần SDK Control Panel của nó. Bảng danh sách này có thể được hiển thị phần nội dung chỉ cần nhấp chuột vào nút Show trên trình soạn thảo.

Nhiều trình soạn thảo NMIT được dùng để tạo ra những nội dung dựa trên XML được định nghĩa bởi Document Type Definition (DTDs). Những trình soạn thảo này thực hiện việc xác nhận nội dung để kiểm tra chúng với DTD, và còn cung cấp những tính năng cho việc chọn lựa các phần tử một cách dễ dàng và các chức năng cho việc chèn thêm dựa trên vị trí hiện tại của con trỏ. Thêm vào đó, NMIT còn cung cấp một trình quản lý DTD mà qua nó bạn có thể nhập thêm vào các DTD mới dùng cho các trình soạn thảo NMIT.

b) Các chức năng

Các chức năng chính của NMIT bao gồm:

Một tập các trình soạn thảo hỗ trợ cho việc tạo lập và sửa đổi nội dung Internet trên di động. Những tính năng này có thể được truy xuất bằng cách sử dụng lệnh File>New hoặc File>Open. Bảng sau đây mô tả một cách ngắn gọn về các trình soạn thảo này.

Browsing Editor

WML 1.3 Deck

Tạo lập một tài liệu WML. Hỗ trợ WML 1.3 DTD, tương thích với đặc tả WAP tháng 6/2000. Có hỗ trợ các tài liệu WML 1.1 và 1.2.

WML Script

Tạo lập nội dung WMLScript. WMLScript bắt nguồn từ ECMA Script và được dùng để thêm các luận lý logic vào một WML Deck, ví dụ như các tính toán.

WBMP Image

Tạo lập một hình ảnh dạng Wireless Bitmap (WBMP). Cũng giống như hầu hết các trình xử lý ảnh, trình soạn thảo WBMP cho phép tạo lập và chỉnh sửa các hình ảnh dạng WBMP, cũng như chuyển đổi những hình ảnh sẵn có từ định dạng GIF và JPEG sang WBMP.

XHTML-MP

Tạo lập một tài liệu XHTML dựa trên XHTML Mobile Profile DTD

XHTML-MP + CHTML

Tạo lập một tài liệu XHTML dựa trên XHTML Mobile Profile DTD với các phần tử cả thuộc tính cộng thêm từ Compact HTML.

CSS

Tạo lập một bảng mẫu Cascading Style Sheet (CSS). CSS chứa các kiểu mẫu định dạng sẽ được áp dụng cho các phần tử được chỉ ra trong một tài liệu XHTML.

Select DTD From List

Tạo lập một tài liệu dựa trên một hệ thống DTD đã được chọn lựa, đó chính là một DTD từ một danh sách kèm theo hay là do bạn tự tạo ra.

Push Content Editors

Service Indication (SI) Editor

Tạo một thông điệp Service Indication Push, thông điệp này được gửi đến người dùng thông báo rằng nội dung mới đã được sẵn sàng.

Service Loading (SL) Editor

Tạo một thông điệp Service Loading Editor, thông điệp này được gửi đến để bắt buộc một dịch vụ người dùng đang chạy trên thiết bị khách cần phải tải về nội dung mới (không thông báo cho người dùng)

Cache Operation (CO) Editor

Tạo một thông điệp Cache Operation Editor, thông điệp này được gửi đi nhằm làm mất hiệu lực nội dung nằm trong cache của dịch vụ người dùng (vì thế buộc phải nạp lại nội dung nếu người dùng yêu cầu nội dung đó lần tiếp theo)

Multipart Message Editor

Tạo một thông điệp đa phần, đây là một loại của thông điệp Push chứa nhiều hơn một phần, mỗi một phần được thực thi riêng lẻ bởi dịch vụ người dùng. Trình soạn thảo sẽ tập hợp và sắp xếp lại những phần đã có sẵn (các tập tin) vào trong một thông điệp đa phần .

Messaging Editors

MMS Wizard

Tạo tập tin Thông điệp Đa phương tiện (Multimedia Messaging) chứa một hay nhiều phần, mỗi phần bao gồm văn bản, hình ảnh, hoặc âm thanh.

MMS Message Editor

Tạo lập hoặc sửa đổi một thông điệp MMS. Các chức năng chính cho phép bạn thêm, xóa, hoặc sắp xếp lại các phần truyền thông; sửa đổi các tiêu đề MMS và tiêu đề của các phần riêng; sau đó đẩy thông điệp này cho bộ SDK được lựa chọn.

SMIL Editor

Tạo một tập tin SMIL (Synchronized Multimedia Integration Language), cho việc chỉ ra các tùy chọn trình diễn của một thông điệp MMS.

Deployment Editors

DRM Editors

Tạo một thông điệp DRM và các thứ tự của nó

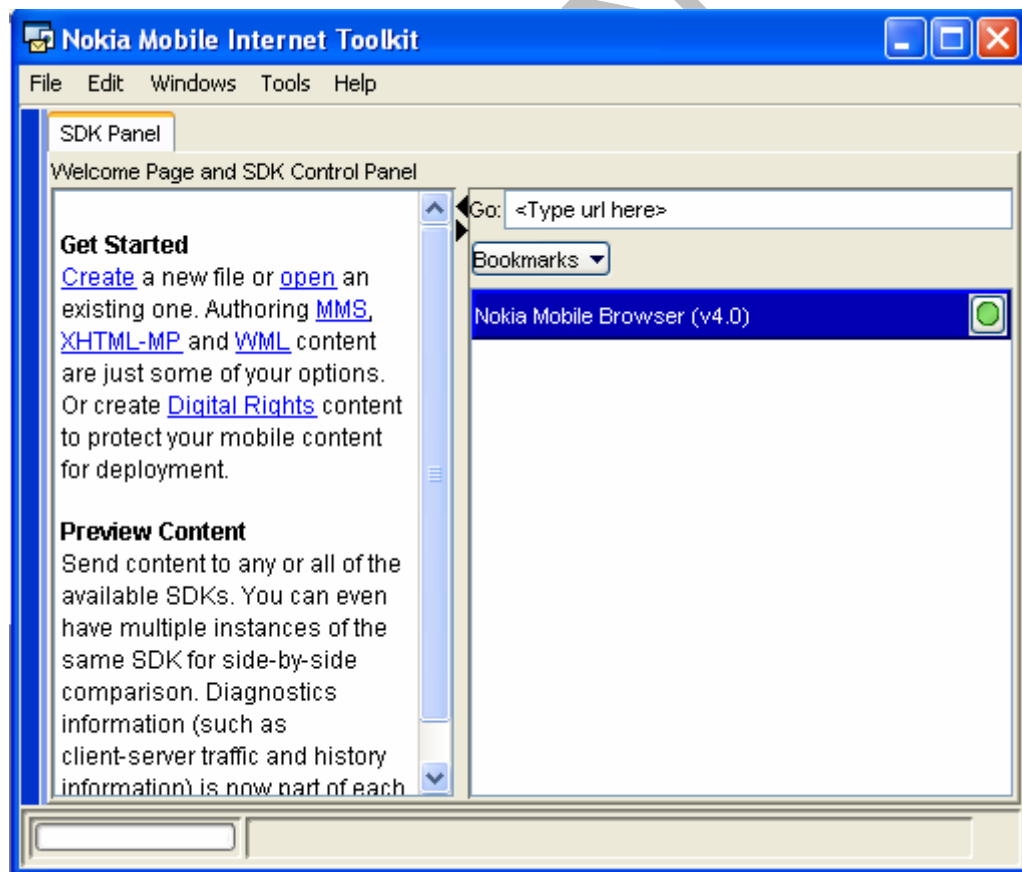
Rights Editor

Soạn thảo các thứ tự trong một trình soạn thảo XML.

Download Descriptor Editor

Tạo nên một Download Descriptor, dùng để mô tả nội dung để người sử dụng điện thoại có thể quyết định xem liệu nội dung này có thể được tải lên điện thoại hay không.

SDK Control Panel được dùng đến khi bạn muốn chọn một hay nhiều bộ SDK điện thoại được cài đặt sẵn để hiển thị nội dung từ một trình soạn thảo hay từ Internet. Panel là một thẻ nằm trên cửa sổ chính và được hiển thị như bên dưới đây.

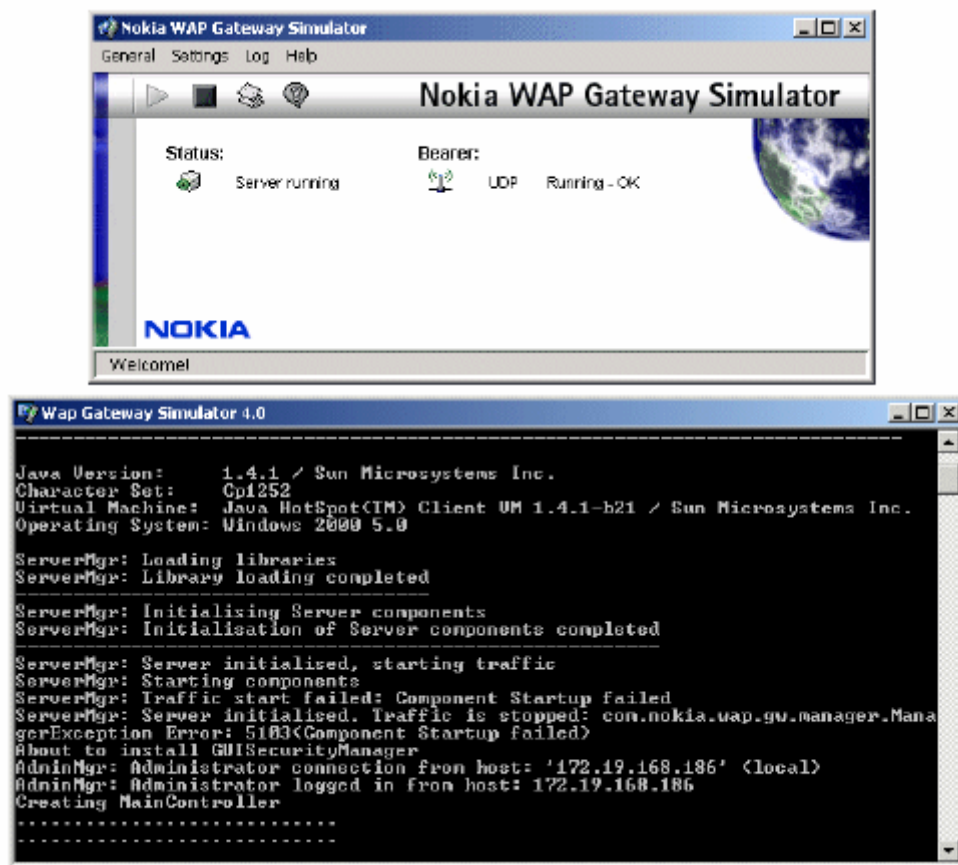


2. Nokia WAP Gateway Simulator

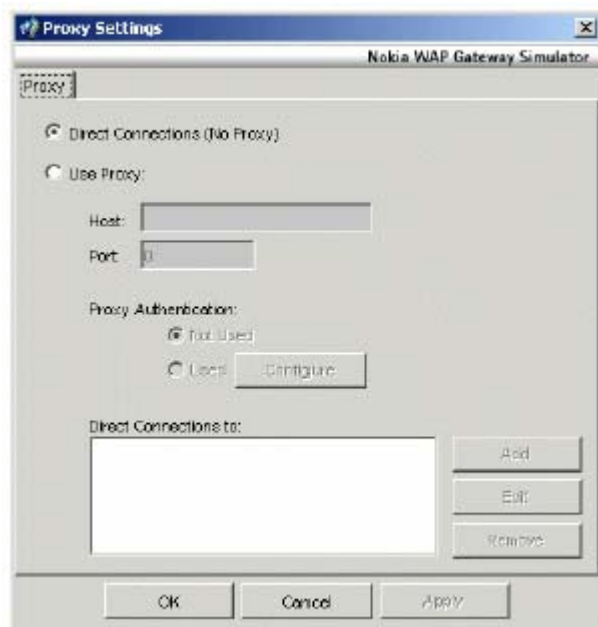
Nokia WAP Gateway Simulator [15], sau đây được đề cập đến dưới tên NWGS, là một WAP Gateway người dùng đơn dựa trên Nokia Activ Server đa người dùng, không được tích hợp chung với NMIT. Khi được cài đặt trên một máy tính, NWGS cho phép người dùng trên máy tính đó có thể truy cập vào mạng Internet trên di động thông qua các chương trình giao tiếp sử dụng giao thức WAP ví dụ như Nokia Mobile Browser Simulator 4.0 SDK.

NWGS bao gồm một trình giải mã các yêu cầu đến từ các dịch vụ WAP khách, chẳng hạn như các trình giả lập điện thoại di động (SDKs), để sau đó chúng có thể được gửi tiếp qua giao thức HTTP đến các server Internet. Nó cũng gồm một trình mã hóa được dùng để mã hóa các trả lời từ server (HTTP) trước khi gửi chúng về cho yêu cầu của các client.

Lúc bắt đầu chạy chương trình, NWGS hiển thị cả cửa sổ quản trị và ứng dụng chủ đang chạy trong cửa sổ Command Prompt, như hình bên dưới đây:



Tùy thuộc vào cấu hình mạng của mình, bạn có thể cần phải chỉ ra một HTTP proxy server. Chẳng hạn như nếu máy tính của bạn nằm bên trong một nhóm Intranet sử dụng HTTP proxy server như là gateway để ra Internet. Nếu như thế, chọn trên menu của NWGS tùy chọn **Setting>Proxy** và sau đó nhập vào tên host và cổng cho proxy trong dialog hiển thị như bên dưới đây:



Nếu bạn cần thêm thông tin về NWGS , tham khảo trong *Nokia WAP Gateway Simulator User's Guide*.

3. Nokia Browser Simulator

Tài liệu này sử dụng Nokia Mobile Browser 4.0 SDK (NMB) [16]



NMB là một công cụ được phát triển nhằm đến các nhà phát triển nội dung Internet trên di động, họ mong muốn xem trước phần nội dung của họ trông như thế nào trước khi nó được thử trên một điện thoại cầm tay thật.

Sử dụng NMB, các nhà phát triển nội dung có thể hiển thị bất kỳ nội dung Internet trên di động nào được phát triển dùng Nokia Mobile Internet Toolkit 4.0 (NMIT), cũng như nội dung tập tin tại cục bộ và nội dung thường trú trên các server Internet và truy cập vào thông qua nối kết WAP. Các nối kết WAP có thể được hình thành thông qua WAP Gateway server hoặc qua trình giả lập WAP Gateway của Nokia (NWGS).

NMB sử dụng phần mềm Nokia Mobile Browser, phần mềm này được Nokia phát triển dùng cho việc triển khai trên các điện thoại cầm tay thật. Tuy nhiên, NMB không được thiết kế tương ứng với chức năng của bất kỳ một thiết bị cầm tay riêng biệt nào mà nó chỉ mở ra một phạm vi mới cho các nhà phát triển Internet trên di động theo công nghệ hiện nay.

Nokia Mobile Browser có thể được sử dụng trong một môi trường độc lập để nạp nội dung cục bộ hay trên Internet di động. Nó có thể quản lý tất cả các dạng nội dung có thể được tạo ra trong NMIT ngoại trừ thông điệp đa truyền thông MMS.

Hệ thống menu:




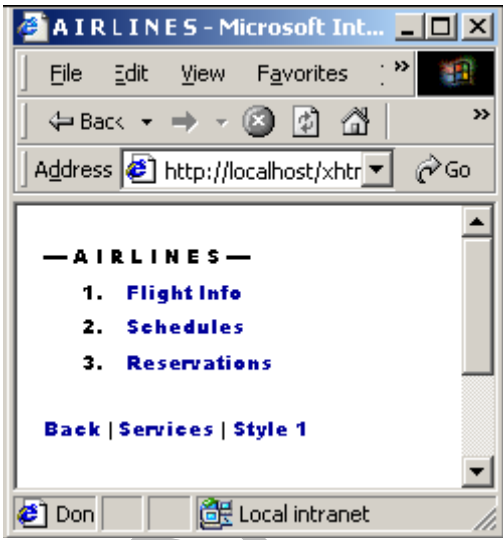
PHỤ LỤC D
BỘ WAP CSS

Sự đổi mới ấn tượng nhất của XHTML MP so với WAP 1.x và các ngôn ngữ đánh dấu dùng trên di động đó là việc hỗ trợ CSS.

CSS là một dạng báo cáo có tư liệu tốt trong các quyển sách hay trên Internet. Trong tài liệu này, sự khác nhau giữa WAP CSS và CSS1 đầy đủ được bàn đến, bên cạnh một vài gợi ý về cách sử dụng một cách hiệu quả trên các thiết bị di động. WAP CSS là một tập riêng của CSS1.

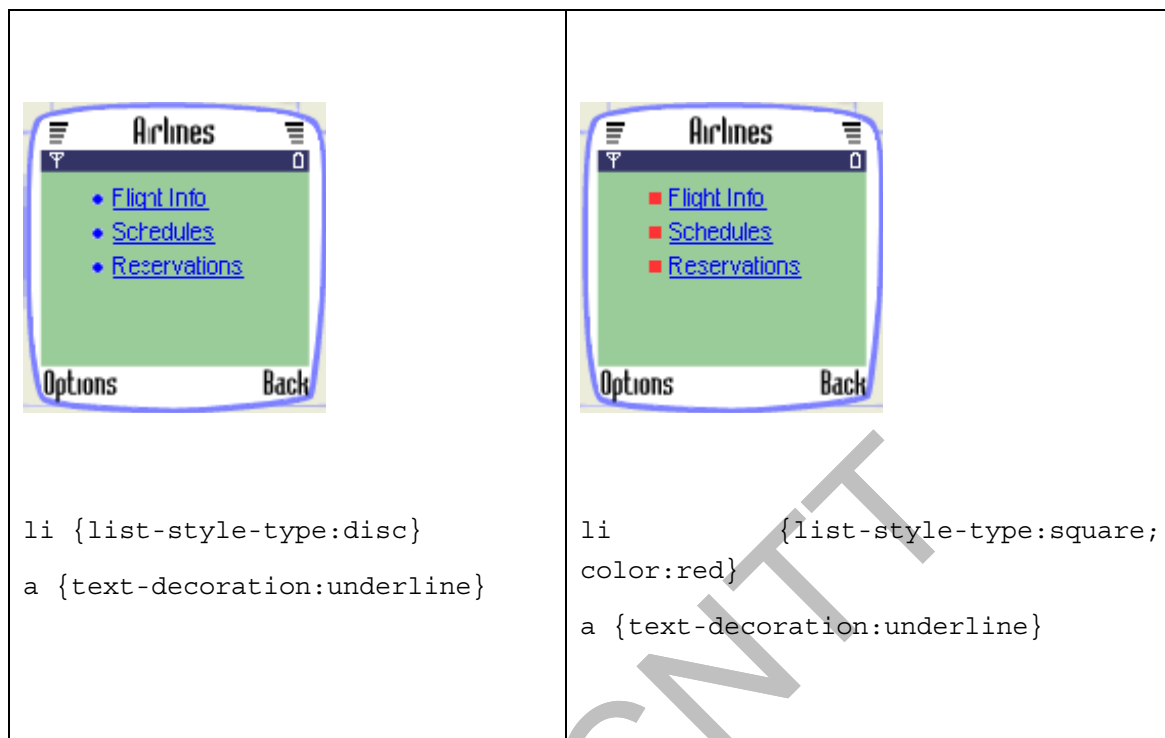
Khả năng của CSS là cung cấp điều khiển chính xác và dễ dàng về mặt nội dung có thể trình bày trên bất kỳ thiết bị di động nào. Mọi góc độ xuất hiện của tài liệu – vị trí, font chữ, các thuộc tính văn bản, đường biên, canh lề, dòng (luồng) – có thể được định nghĩa bằng cách dùng các mẫu. Nếu cần thay đổi bất kỳ một giá trị nào thì chỉ cần thực hiện tại một vị trí duy nhất. Thực tế, bằng việc sử dụng các mẫu chia sẻ bên ngoài, việc thay đổi bất kỳ một cách trình bày nào của toàn bộ tài liệu trên một trang chỉ cần thực hiện một lần.

Điều khiển cung cấp bởi CSS giúp chương trình dễ dàng được định dạng lại nội dung cho các thiết bị di động khác. Trong CSS, sự trình bày, kiểu mẫu, và việc định vị được dễ dàng định dạng lại đối với mỗi trình duyệt. Trong hình X, cùng một trang XHTML MP được hiển thị trên hai trình duyệt khác nhau, với mỗi trình duyệt sử dụng riêng bộ mẫu của mình. Cột 1 là nội dung hiển thị trên Nokia Mobile Browser 4.0, trong khi cột 2 lại là cùng một nội dung nhưng được hiển thị trên Internet Explorer 5.0.

Nokia Browser 4.0	Microsoft IE 5.0
	

Hình 8.2-1: Các kiểu định dạng khác nhau trên các trình duyệt khác nhau

Hình C-2 sau đây mô tả việc sử dụng các thuộc tính định dạng CSS để thay đổi thể hiện của các bullet trong một danh sách được bulleted. Các định nghĩa thuộc tính CSS được mô tả bên dưới mỗi trình bày. Các mẫu bullet được định nghĩa trong bản mẫu của CSS, không phải trong tài liệu XHTML MP. Sử dụng CSS, bạn có thể dùng các bullet trong các kiểu mẫu khác nhau, từ các số kiểu Roman và Latin, đến các hình vuông và hình tròn.



Hình 8.2-2: Thay đổi kiểu bullet sử dụng mẫu

1. Các vấn đề được kiểm soát bởi các dạng mẫu

Sự cách khoảng, thêm mới, và canh lề.

Canh hàng và dàn trang.

Font chữ và các thuộc tính văn bản.

Các đường biên.

Màu trên các thiết bị có khả năng hiển thị màu.

Sự chuyển đổi văn bản và nền trên các thiết bị không có khả năng hiển thị màu.

Mặt nạ định dạng nhập văn bản sử dụng `-wap-input-format`.

2. Áp dụng các kiểu định dạng

Mỗi trình duyệt sử dụng một bản mẫu mặc định của mình để chỉ rõ cách các phần tử XHTML sẽ được hiển thị như thế nào. Vì thế, ngay cả khi bạn không đưa ra một kiểu mẫu nào, hay chỉ là một vài kiểu, thì phần còn lại của tài liệu cũng được

hiển thị sử dụng các kiểu mẫu được đưa ra trong bảng mẫu mặc định. Các dạng mẫu được định nghĩa trong bản mẫu mặc định có ưu tiên thấp nhất trong thứ tự thác nước, điều này cho thấy rằng nếu bạn định nghĩa một dạng mẫu nào đó của riêng bạn thì nó sẽ được thực hiện trước các dạng mẫu mặc định.

a. Các bảng định dạng bên ngoài

Bên trong phần tử head của một phần tài liệu bạn có thể sử dụng một hay nhiều phần tử link để chỉ đến một hay nhiều URL, những địa chỉ này xác định các bản mẫu bên ngoài được dùng để định dạng cho tài liệu hiện hành. Ví dụ như, dòng lệnh sau đây mô tả cách sử dụng bảng mẫu mydoc.css, nó được đặt trong cùng đường dẫn với tài liệu hiện hành.

```
<link rel="stylesheet" href="mydoc.css" type="text/css"/>
```

b. Phần tử style trong đầu đề tài liệu

Phần tử <style> có thể sử dụng bên trong phần tử <head> của tài liệu nhằm để xác định một dạng mẫu áp dụng cho toàn bộ tài liệu. Trong ví dụ tiếp sau đây phần tử style cho biết là màu xanh được dùng làm màu chữ cho tất cả các đoạn văn bản (được đánh dấu bởi các phần tử <p>).

```
<style> p {color: blue} </style>
```

c. Phần tử style trong thân tài liệu

Phần tử <style> có thể được dùng bất cứ ở đâu trong giới hạn của hai phần tử <body> và </body> của một tài liệu. Các định nghĩa mẫu theo cách này sẽ được thực hiện mà không quan tâm đến các định nghĩa được đưa ra trước đó (bên trong một bảng mẫu ngoài hay do một phần tử <style> được chỉ ra trong <head> của tài liệu) trong cùng phần tử. Đó là do một dạng mẫu trực tiếp được thực hiện sau cùng trong thứ tự thác nước.

d. Luật thác nước cho các phần tử mẫu

Vị trí bên trong tài liệu mà các phần tử kiểu mẫu được chỉ ra (hoặc là phần tử <style> hoặc là <link> cần được chú ý: Các phần tử kiểu mẫu được đưa ra trước

tiên sẽ bị đè lên bởi chính các phần tử mẫu được định nghĩa sau đó trong thứ tự của tài liệu. Bất cứ một thiết lập thuộc tính mẫu nào áp dụng cho một phần tử XHTML riêng biệt sẽ có quyền cao hơn tất cả các thiết lập trước cho thuộc tính đó trên phần tử đó.

Ví dụ giả sử rằng một tài liệu XHTML MP chứa một nối kết đến một bảng mẫu ngoài như sau:

```
<link rel="stylesheet" href="your-style-sheet-url.css"
                                     type="text/css"/>
```

và giả định rằng bảng mẫu được tham chiếu đến chứa định nghĩa sau:

```
p {font-family: arial; color: black}
```

Bây giờ, nếu như một phần tử `<style>` như sau được đưa ra trong tài liệu trên một dòng tiếp sau dòng có chứa phần tử `<link>`:

```
<style> p {color: blue} </style>
```

khi đó các phần tử `<p>` trong tài liệu sẽ sử dụng font chữ Arial, nhưng màu chữ sẽ là màu xanh.

Tuy nhiên, nếu như thứ tự bị đảo lại:

```
<style> p {color: blue} </style>
<link rel="stylesheet" href=" style-sheet-url.css"
type="text/css"/>
```

Khi đó màu của chữ sẽ là màu đen.

e. Sử dụng các thuộc tính của XHTML

style

Thuộc tính style cho phép đặt kiểu mẫu cho một phần tử đơn lẻ tại một điểm trong tài liệu tại nơi phần tử được đưa ra. Tham chiếu đến như là một kiểu mẫu bên trong (“inline style”), đặc trưng này là một phần mở rộng trong XHTML MP. Nó cho phép áp dụng các kiểu mẫu cho các phần tử riêng lẻ, do đó được ưu tiên hơn bất

kỳ kiểu mẫu nào khác được định nghĩa trong các phần tử `<style>` trước đó hay các bảng mẫu ngoài:

```
<p style="color:red">red</p>
```

Một hạn chế của việc chỉ ra các kiểu mẫu bên trong là nó làm cho việc tìm kiếm và thay đổi tất cả sự cố của mẫu khó khăn. Một hạn chế khác là việc sử dụng chúng có thể làm tăng kích thước của tài liệu nếu như được sử dụng nhiều lần để thay đổi các kiểu mẫu của cùng một phần tử tại những nơi khác nhau bên trong tài liệu.

Sẽ tốt hơn nếu định nghĩa một lớp chứa các kiểu mẫu mà ta muốn áp dụng cho nhiều phần tử, chẳng hạn như sau:

```
.class1 {padding-bottom: 4px}
```

Lúc này ta có thể áp dụng kiểu mẫu định dạng một cách dễ dàng chỉ cần định nghĩa định dạng tại một vị trí (trong bảng mẫu). Ví dụ, để áp dụng kiểu mẫu này cho một phần đoạn riêng biệt nào đó, bạn có thể sử dụng thuộc tính class bên trong phần tử đoạn và chỉ ra giá trị của thuộc tính này chẳng hạn là “class1” như được mô tả bên dưới:

```
<p class="class1">This paragraph will have  
bottom padding. </p>
```

id

XHTML cùng với WAP CSS có thể cho phép bạn giữ các định nghĩa mẫu của mình tại cùng một điểm (trong một bảng mẫu hay trong một đoạn mẫu bên trong phần tử `<head>`), ngay cả khi bạn muốn áp dụng một mẫu cho một phần tử đơn nhưng không ảnh hưởng đến các phần tử khác cùng một loại.

Ví dụ như bạn có thể định nghĩa một mẫu cho một phần tử `<h1>` riêng lẻ trong một đoạn mẫu bên trong phần tử `<head>` như sau:

```
<head>
<style>
#myid {border-width: 1; border: solid; text-align:
center}
</style>
</head>
```

và sau đó áp dụng mẫu này cho một phần tử `<h1>` riêng như sau:

```
<h1 id="myid"> This heading has a solid order.</h1>
```

div và span

Ngoài việc áp dụng các kiểu mẫu cho một nội dung được đưa ra bằng một hay nhiều phần tử như đã được bàn đến ở trên, bạn cũng có thể sử dụng các mẫu, được định nghĩa trong bảng mẫu hay bên trong phần tử `<head>`, cho một chuỗi các phần tử sử dụng phần tử `div` hay một chuỗi các ký tự bên trong sử dụng phần tử `span`. Ví dụ một bảng mẫu chức các giá trị sau:

```
<style type="text/css">
div.test-case {text-align: center; color: blue}
span.test-id {color: red}
</style>
```

Bên trong phần thân tài liệu, bạn có thể áp dụng kiểu mẫu được đưa ra bởi phần tử `div` cho một chuỗi các đoạn văn và kiểu mẫu được đưa ra bởi phần tử `span` cho một chuỗi các ký tự như sau:

```
<div class="test-case">
<p>This paragraph is blue.</p>
<p>Here are <span class="test-id">Test51 and
Test52</span> specs.</p>
</div>
```

Trong đoạn code ở trên, phần nội dung của cả hai đoạn văn được canh giữa và sử dụng font chữ màu xanh, ngoại trừ chuỗi ký tự “Test51 and Test52”, được sử dụng font màu đỏ.

3. Những điều cần tránh

Tránh ẩn đường bao nội dung XHTML. Hay nói cách khác, không sử dụng giá trị nowrap của white-space property. Lốp bao được nhấn mạnh sử dụng bởi vì nó cho phép nhiều nội dung được nhìn thấy ngay mà không cần phải dùng đến thanh cuộn. Lốp bao được kích hoạt tự động trong Nokia Mobile Browser v3.0.

Tránh việc xóa các đường gạch dưới từ các phân tử <a>. Hay nói cách khác, không sử dụng mẫu text-decoration:none trong các liên kết. Người sử dụng sẽ không thể phân biệt được liệu một liên kết không được gạch dưới có phải là một liên kết không hay đơn giản chỉ là một đoạn văn bản bình thường. Hơn nữa, một nối kết được gạch dưới đã khá quen thuộc với người dùng và việc bỏ đi phần gạch dưới sẽ làm cho tiền định vị của dịch vụ trở nên mơ hồ và chậm lại mà thôi.

Và ngược lại, tránh sử dụng text-decoration:underline property trong dạng văn bản bình thường. Việc gạch dưới phần văn bản sẽ làm cho nhiều người sử dụng hiểu nhầm vì nó trông giống như một là liên kết. Thay vào đó, có thể làm nổi bật văn bản bằng cách khác, ví dụ như làm cho nó đậm lên chẳng hạn. Nói chung là các dạng văn bản gây phiền phức cần được tránh nhằm giúp tài liệu dễ đọc và tập trung hơn vào nội dung của tài liệu.

Tránh làm nổi bật văn bản quá nhiều thông qua việc sử dụng font-style và font-variant, đặc biệt là trên cùng một trang XHTML. Quá nhiều và quá khác nhau sẽ làm cho nội dung tài liệu mất đi sự tập trung. Cố gắng áp dụng phần nhấn mạnh trên chỉ một vài từ trong mỗi trang XHTML.

Tránh sử dụng quá nhiều loại font chữ (font-family) và kích thước (font-weight, font-size) khác nhau trong cùng một trang XHTML. Cố gắng giới hạn việc sử dụng các kích thước và các loại font chữ trong mỗi trang XHTML là từ một đến hai mà thôi. Đồng thời cũng tránh định nghĩa từ mà chỉ sử dụng các ký tự in hoa vì

nó thường được hiểu như là những tiếng reo hò, và chúng lại khó đọc hơn là các ký tự thường.

Không nên xóa các đường biên trên các bảng được dùng để mô tả thông tin theo cách ghi thành bảng.

Ở những nơi có thể, định nghĩa các kiểu mẫu sử dụng các luật thác nước thay cho việc dùng các thuộc tính class hay id bên trong các phần tử. Ví dụ như trong bảng mẫu WAP CSS của bạn sử dụng các thuộc tính như:

```
p {color:red}
```

hơn là dùng thuộc tính class như sau:

```
.red {color:red}
```

Nhưng loại trừ khi cần phải đưa ra chuỗi class="red" bên trong mỗi phần tử <p> trong một tài liệu.

PHỤ LỤC E
CÁC NGUYÊN TẮC THIẾT KẾ

1. Trước khi thiết kế một wapsite

Khi thiết kế một dịch vụ nhắm vào cả thiết bị di động và máy tính để bàn thì phải bắt đầu bằng bảng thiết kế cho giao diện người dùng trên thiết bị di động. Việc mở rộng một dịch vụ hướng di động cho môi trường máy tính để bàn thường sẽ dễ dàng hơn là làm theo cách ngược lại.

Khi bắt đầu bằng một Website dành cho máy tính để bàn, người ta khuyến cáo rằng nên chia dịch vụ thành những phần nhỏ. Ta chỉ chọn những phần là trung tâm của hệ thống dịch vụ để dành cho phiên bản trên môi trường di động. Cần phải chú trọng vào các phần trung tâm này khi thiết kế dịch vụ trên di động.

Cần phải nghiên cứu đặc tính vật lý của các trình duyệt di động khác nhau trên thị trường để có thể làm một bản thiết kế hiệu quả sử dụng XHTML và CSS. Cần phải nắm rõ kích thước tối đa của trang nội dung, bảng định dạng (style sheet) và các hình ảnh đồ họa; lượng không gian màn hình cho phép để hiển thị nội dung; và lượng không gian màn hình dành cho các đối tượng khác như: văn bản soft-key, các biểu tượng, các tiêu đề v.v..

Ví dụ như các điện thoại di động của Nokia có một đường tiêu đề chứa các biểu tượng và có thể chứa tiêu đề của trang XHTML. Nó cũng có một vùng nội dung nơi dùng để hiển thị nội dung của trang XHTML và một vùng chứa một hay nhiều phím mềm (soft key) tùy thuộc vào từng model điện thoại khác nhau

2. Các nguyên tắc chung cho một thiết kế tốt

Việc đưa ra XHTML MP và CSS tạo nên một số lượng đa dạng các giao diện người dùng mới. XHTML có nhiều phần tử/thẻ hơn WML, và cách hiển thị các phần tử có thể được hiệu chỉnh theo nhiều cách với CSS. XHTML MP cung cấp nhiều khả năng hơn cho các nhà cung cấp dịch vụ giúp cho dịch vụ của họ hấp dẫn, lôi cuốn hơn đồng thời thêm vào đó là sự phức tạp do đó tạo nên những thách thức về tính tiện lợi.

3. Cần chú ý đến mô hình liên kết

Ta cần cung cấp một giao diện người dùng dễ dàng sử dụng được xây dựng cùng với một mô hình định vị nhất quán và dễ dàng nắm bắt. Điều này quan trọng hơn là việc cố gắng sử dụng tất cả các khả năng hiển thị lõi cuốn của XHTML.

Nhu cầu và sự kỳ vọng của những người dùng di động khác rất nhiều so với người dùng trên máy tính để bàn. Những người dùng di động thường mong muốn truy cập một cách dễ dàng và nhanh chóng đến những thông tin họ cần. Do đó, các nội dung trên di động cần phải ngắn gọn, chính xác và nhanh chóng.

Tránh việc tạo nên các nội dung thừa, không cần thiết hoặc các trang splash (splash screen) rối rắm; tuy nhiên, ta có thể hiển thị một logo nhỏ hoặc những phần nhấn mạnh thương hiệu tạo nên sự gần gũi với khách hàng. Trong nhiều trường hợp nên hiển thị nội dung chính yếu ngay khi người dùng có yêu cầu xem nó.

Việc nhập dữ liệu cũng là một việc làm gây khó khăn và mất thời gian trên phần lớn các thiết bị di động (các thiết bị không được cung cấp bàn phím chuẩn QWERTY), do đó cần xây dựng wapsite với yêu cầu nhập nội dung tối thiểu, nhất là nội dung văn bản. Để làm được điều đó ta có thể cân nhắc khả năng giao tiếp với người dùng để chọn hình thức nhập khác như danh sách chọn (listbox, combo box), các nút chọn (check box, radio button) thay vì sử dụng ô nhập văn bản thuần túy.

Khi thực sự cần phải nhập thông tin, ta có thể sử dụng thuộc tính `-wap-input-format` và đặt input mask, ví dụ như `*N` để nhập số. Điều này giúp cho người dùng tránh được thao tác chuyển đổi chế độ nhập.

Thuộc tính `-wap-input-format` của CSS xác định một input mask cho dữ liệu mà người dùng nhập vào, do đó hạn chế việc chuyển đổi chế độ nhập giữa text và số. Tuy nhiên trên những phiên bản trình duyệt XHTML không phải của Nokia trước đây chưa hỗ trợ thuộc tính này mà chỉ hỗ trợ thuộc tính cũ `format=""`, do đó cần phải đặt cả hai thuộc tính là `-wap-input-format` và `format` với cùng một chuỗi định dạng (ví dụ `*N` cho dữ liệu số). Xem thêm bảng đặc tả WAP June 2000

[WML] hoặc WAP Overview [WAPOver] để rõ hơn chi tiết cú pháp của WAP input mask.

Nhiều người dùng di động phải trả theo thời gian truy cập do đó nếu họ không lấy được thông tin họ cần trong một thời gian ngắn thì họ sẽ ngưng dùng dịch vụ.

4. Thiết kế hệ thống phân cấp trong liên kết

Mô hình định vị là cách mà một người dùng duyệt qua các trang XHTML của một dịch vụ, tương tác thông qua các liên kết, các trình đơn và dữ liệu nhập. Để xây dựng một mô hình định vị cho wapsite, cần phải bảo đảm các yêu cầu sau:

- Mô hình định vị phải thống nhất trên toàn thể hệ thống
- Đối với các dịch vụ XHTML, tránh thêm vào các liên kết dẫn về trang vừa mới truy cập bởi vì bộ trình duyệt sẽ xử lý việc này một cách tự động thông qua một phím back được tích hợp sẵn.
- Tránh xây dựng một hệ thống quá sâu. Người dùng sẽ gặp khó khăn trong việc duy trì một cái nhìn tổng thể của một dịch vụ chứa 4 hoặc 5 lớp.
- Đưa vào một hướng liên kết trở về trang khởi đầu hoặc các nhánh chính của hệ thống để người dùng dễ dàng quay trở lại điểm bắt đầu. Hệ thống định vị càng sâu thì càng cần phải có một liên kết trở về trang khởi đầu.

5. Nguyên tắc thiết kế cho màn hình nhỏ

Các dự báo cho rằng số lượng thiết bị di động sẽ nhanh chóng vượt qua số lượng máy tính để bàn trên toàn thế giới, tạo nên một cơ hội kinh doanh khổng lồ cho các ứng dụng thân thiện người dùng được thiết kế cho những thiết bị có màn hình nhỏ này. Một điều chắc chắn rằng sẽ có nhiều thách thức cho việc hiển thị trên thiết bị nhỏ nhưng khó có thể tạo nên các ứng dụng lôi cuốn trên các thiết bị di động như trên máy tính để bàn. Ta nên tuân theo các yêu cầu sau khi thiết kế các nội dung dành để hiển thị trên các thiết bị có kích thước màn hình nhỏ:

- Cần đảm bảo là có thông tin được hiển thị khi người dùng truy cập vào trang wap
- Sử dụng mục <title> trong phần <head> để cung cấp một tiêu đề ngắn cho mỗi trang wap. Thông thường tiêu đề không nên dài quá 14 ký tự trừ khi ta có ý định hiển thị trang wap trên một thiết bị di động xác định.
- Sử dụng định dạng thống nhất cho tất cả các trang XHTML trong một dịch vụ. Tính thống nhất đó sẽ giúp nâng cao khả năng nắm bắt, đặc biệt là những người dùng sử dụng dịch vụ thường xuyên.
- Hạn chế việc cuộn trang theo chiều ngang. Thêm vào đó người dùng có thể trở nên mất định hướng về vị trí của họ trên toàn thể trang. Nếu có thể thì nên thiết kế nội dung không rộng hơn hoặc dài hơn màn hình hiển thị của thiết bị.
- Sử dụng thuộc tính canh lề (left, right, center) cho các thành phần để tăng tính rõ ràng, nhưng tránh sử dụng nhiều hơn hai hoặc ba kiểu trên cùng một trang đơn bởi vì khi đó sẽ làm cản trở khả năng nắm bắt cấu trúc tổ chức của trang.
- Sử dụng khoảng trắng đặc biệt là dọc theo các hình ảnh cao, hẹp. Ta có thể làm như thế bằng cách sử dụng thuộc tính align trong phần tử . Ví dụ: . Thêm vào đó trong file CSS ta có thể làm như thế (thậm chí tốt hơn nữa) một trong các cách đó là đặt thuộc tính float cho phần tử . Khi đó các hình ảnh với thuộc tính này cho phép hiển thị văn bản dọc theo nó do đó sẽ sử dụng hết phần hiển thị của thiết bị.
- Tránh lạm dụng các thuộc tính làm nổi bật văn bản như in đậm (bold), in nghiêng (italic) và gạch dưới (underline) vì khi đó chúng sẽ làm giảm sự rõ ràng của trang wap.

- Tránh sử dụng các từ dài, phức tạp trong khi có thể dùng các từ ngắn gọn, và súc tích hơn.
- Tránh sử dụng quá nhiều màu khác nhau trên cùng một trang. Mặc dù màu sắc làm cho dịch vụ trở nên lôi cuốn hơn nhưng quá nhiều màu có thể gây phản tác dụng. Cố gắng sử dụng màu sắc thống nhất, ví dụ có thể sử dụng cùng một màu cho một phần tử XHTML trên toàn bộ hệ thống.
- Tránh đề cập đến tên màu sắc, ví dụ như: “nhấn vào liên kết màu đỏ để...” bởi vì trên những thiết bị di động không có màn hình màu thì các nội dung có màu sẽ trở thành trắng đen.

6. Đảm bảo các tài liệu phải có kích thước nhỏ

Bởi vì bộ nhớ của các thiết bị di động có giới hạn nên phải giữ cho tài liệu có kích thước càng nhỏ càng tốt. Tuy nhiên, do XHTML MP không hỗ trợ chế độ nhiều card trong một tài liệu như WML nên việc chia nhỏ nội dung thành nhiều trang riêng biệt sẽ làm cho việc load các trang chậm hơn. Để giải quyết vấn đề này ta nên tập hợp tất cả các nội dung tóm tắt lên một trang và sử dụng các liên kết để giúp di chuyển đến các phần tương ứng.

Một số tiếp cận hữu ích để làm cho kích thước của trang nhỏ:

- Không đưa vào trang các lời chú thích dài. Mặc dù đây là một thói quen tốt trong lập trình nhưng nó không phù hợp cho các tài liệu dành riêng cho các thiết bị di động.
- Sử dụng kí tự tab thay vì khoảng trắng để lùi đầu dòng, hoặc thậm chí không lùi đầu dòng nếu có thể.
- Sử dụng tên ngắn cho tên các file, tên các lớp CSS và các ID trong CSS.
- Định nghĩa các kiểu định dạng bằng qui tắc cascading thay vì các thuộc tính lớp và ID trong các phần tử. Ví dụ, trong bảng định dạng WAP CSS sử dụng thuộc tính `p(color:red)` thay vì sử dụng thuộc tính lớp như

sau `.red {color:red}`. Điều này sẽ hạn chế việc phải xác định chuỗi `class="red"` trong mỗi phần tử `<p>` trong tài liệu.

7. Tạo các ứng dụng trên điện thoại di động

Khi quyết định thông tin nào được đưa vào các ứng dụng khác nhau trên một thiết bị di động cần cân nhắc từng tình huống mà thiết bị sẽ được sử dụng. Nội dung của dịch vụ phải thỏa mãn được nhu cầu của nhóm người dùng đích và phải được tối ưu hóa cho những tác vụ thông thường. Bởi vì tính di động của thiết bị người dùng có thể sử dụng nó, chủ yếu khi không có máy tính để bàn để truy cập Internet, để lấy thông tin một cách nhanh chóng. Ví dụ như truy cập nhanh để lấy lịch chuyến bay, mẫu tin ngắn và thông tin thời tiết. Hiếm khi những người dùng đó sử dụng điện thoại di động của họ để lướt web một thời gian dài.

8. Đảm bảo các tác vụ tiến hành trôi chảy và sử dụng hợp lý các hình ảnh

Các trang wap đầy màu sắc sẽ trong hấp dẫn hơn nhưng sẽ kém phần hấp dẫn nếu các hình ảnh làm cho dịch vụ bị chậm lại. Theo các nghiên cứu trong việc sử dụng, người dùng ít nhiệt tình hơn đối với các dịch vụ mà hình ảnh minh họa làm chậm trễ các tác vụ của họ. Đặc biệt, những hình ảnh lớn không được đánh giá cao khi người dùng đang được liên kết đến trang cần tìm. Những hình ảnh có chứa đựng thông tin sẽ được đánh giá cao nhưng trong nhiều trường hợp người dùng sẽ không cho hiển thị hình ảnh để tiết kiệm thời gian và tiền bạc cũng như sẽ chuyển đến trang kế tiếp mà không chờ đến khi hình ảnh được tải về đầy đủ. Điều đó rất quan trọng khi cho phép người dùng chuyển đến các trang khác thậm chí trước khi tất cả hình ảnh được tải về.

Các bảng (table) lớn cũng gây ra vấn đề tương tự, đó là người dùng có thể sẽ bị mắc kẹt tại một trang cho đến khi nó được tải về hết hoặc không thể tìm được cách để thực hiện tiếp trước khi trang wap được tải về đầy đủ. Do màn hình hiển thị của các điện thoại di động có kích thước khác nhau nên cần phải đảm bảo các bảng

dữ liệu có thể đọc được thậm chí trên những màn hình hiển thị nhỏ nhất; thường thì chúng sẽ bị ép lại cho vừa với màn hình.

9. Đảm bảo cấu trúc wapsite dễ dùng đối với người mới sử dụng

Đối với các dịch vụ trên di động thường thì một cấu trúc không quá sâu sẽ dễ nắm bắt hơn cho người dùng. Các liên kết và các trang wap nên cung cấp các tên có tính mô tả giúp cho người dùng tìm được thông tin mà họ cần.

Thật khó để có thể đưa ra số lượng liên kết hợp lý trên một trang danh sách liên kết. Nếu các liên kết rõ ràng với nhau và dễ dàng để duyệt qua (mỗi liên kết trên một dòng, theo thứ tự chữ cái hoặc theo thứ tự logic khi đó người dùng không phải đọc hết tất cả các liên kết), khi cung cấp khoảng 30 liên kết trên một trang đơn sẽ tốt hơn là 5 liên kết trên 6 trang khác nhau. Nếu có khoảng vài chục liên kết thì nên cung cấp tùy chọn để sắp xếp các liên kết trước khi hiển thị chúng lên. Một liên kết nằm vừa trên một dòng giúp cho việc chọn lựa dễ dàng hơn và trang wap trông sẽ tốt hơn.

Không có phần tử <do> trong WAP 2.0, thay vào đó chúng được thay thế bằng các phím truy cập. Tuy nhiên hầu hết người dùng dường như không quan tâm tới các phím truy cập và cũng không thể tìm thấy chúng. Để giúp người dùng hiểu được khái niệm đó, cần đảm bảo rằng các phím truy cập hiển thị trên màn hình và dưới dạng giống như các phím điện thoại.

Nếu không thể thực hiện, thì nên cung cấp một chức năng tìm kiếm. Những người dùng có kinh nghiệm sẽ đánh giá cao nó.

10. Cung cấp vừa đủ thông tin trên một trang

Những trang tương tác nên ngắn gọn. Ta nên bắt đầu một wapsite bằng một trang giới thiệu ngắn, trang này chỉ hiển thị lời chào và logo của dịch vụ. Điều này sẽ tốt hơn khi người dùng đi đến trực tiếp trang wap dịch vụ.

Đối với XHTML, nội dung được tải xuống dưới dạng các trang, không phải là các thẻ như trong WML. Có nghĩa là việc cung cấp đầy đủ thông tin trên một

trang đơn sẽ quan trọng hơn cho các tác vụ của người dùng. Việc đi tới và lui giữa các trang có thể mất nhiều thời gian hơn trong XHTML bởi vì các trang được tải về một cách riêng biệt. Đối với trường hợp các trang không thể lưu (cache) tại thiết bị - ví dụ như các nội dung phải trả tiền hoặc có chứa các thông tin cá nhân, thì việc đó càng quan trọng hơn.

Phần hiển thị trên cùng của trang là phần quan trọng nhất. Tất cả các liên kết thường dùng, phần tìm kiếm, phần đăng nhập và phần thông tin chủ yếu sẽ được đặt ở đó. Người dùng có thể di chuyển đến các trang khác mà không phải chờ phần còn lại của trang được tải về đầy đủ cũng như không phải cuộn trang.

Tránh sử dụng phần trên cùng của trang cho bảng quảng cáo hoặc những hình ảnh mang tính minh họa. Tốt hơn là nên đặt quảng cáo ở bên trái hoặc bên phải trang.

Việc cuộn trang lên xuống rất khó khăn nên các trang tương tác có chứa các form dữ liệu không nên quá dài. Người dùng sẽ không biết được họ đã điền đầy đủ các thông tin hay chưa trên một form quá dài. Người dùng có thể mất khả năng điều khiển nếu form nhập dài hơn hai trang màn hình.

Tại trang đích mà người dùng đang hướng đến phải chứa đầy đủ các thông tin mà họ cần. Ví dụ như, nếu trang đích chứa một câu chuyện hoặc một hướng dẫn thì toàn bộ nội dung nên được chứa trên cùng một trang.

Thực tế là thông tin được tải về dưới dạng các trang riêng biệt là thay đổi lớn nhất ảnh hưởng đến sự định hướng và cấu trúc giữa WML và XHTML.

11. Phản ánh được hành động của người dùng

Nên cung cấp sự phản ánh cho các hành động của người dùng cũng như các tình huống lỗi. Ví dụ, sau khi người dùng click vào một liên kết thì trang mới hiển thị phải có tiêu đề mang thông tin giống như liên kết đó. Việc tối thiểu các bước trong định hướng cũng tạo nên cảm giác không an toàn cho người dùng, ví dụ có thể cần đến các trang xác nhận (confirmation page) cho các hành động của người

dùng, mặc dù điều đó cần thêm một hành động click nữa. Nếu không có các trang xác nhận người dùng có thể cảm thấy cần phải kiểm tra xem hành động có tác dụng không, điều này làm tăng số lượng tương tác của họ. Người dùng nên được thấy rằng họ đang điều khiển hệ thống.

Nếu có vấn đề xảy ra người dùng phải được hướng dẫn bước phả làm kết tiếp. Có thể tránh được các lỗi nhập liệu nếu người dùng được cung cấp các định dạng dữ liệu mẫu.

12. Hạn chế số lượng và kích thước của màn hình

Số lượng và kích thước các hình ảnh trong XHTML cần phải được cân nhắc cẩn thận. Mỗi hình ảnh trên một trang tạo nên một luồng truyền tải riêng, điều đó sẽ làm chậm lại thời gian hiển thị của toàn bộ trang. Do đó số lượng các luồng tải hình ảnh phải được hạn chế tối đa. Đồng thời mỗi khi một hình ảnh được tải đến thiết bị di động thì toàn bộ trang sẽ được sắp xếp lại điều này mất nhiều thời gian mà tài nguyên xử lý của thiết bị. Do đó, một trang nội dung với chỉ một vài hình ảnh có thể sẽ được tải về nhanh hơn một trang với nhiều hình ảnh nhỏ. Nếu có thể nên sử dụng cùng các hình ảnh cho nhiều trang khác nhau trong cùng một dịch vụ khi đó một hình ảnh sẽ được tải về chỉ một lần và lưu vào bộ lưu trữ (cache). Ví dụ, nếu các hình ảnh được dùng cho các nút (bullet) thì chúng nên được dùng trên toàn bộ hệ thống.

Kết nối theo giao thức TCP/IP có thể làm cho tốc độ tải các trang sẽ khác nhau mặc dù lượng dữ liệu thì như nhau. Ví dụ, một trang có chứa 4 hình ảnh với dung lượng 2KB mỗi hình sẽ nhanh hơn là tải về một trang chứa 8 ảnh với dung lượng 1KB mỗi ảnh.

Nếu có sử dụng WAP gateway thì nó nên được đặt gần với trạm GSSN (Gateway GPRS Support Node). Việc mất dữ liệu sẽ làm tăng độ trễ do cơ chế truyền lại dữ liệu bị mất của giao thức HTTP. Độ trễ giữa WAP gateway và máy chủ chứa nội dung cần được hạn chế tối đa.

13. Thiết lập các thuộc tính chiều cao và chiều rộng màn hình

Các nhà phát triển nội dung nên chỉ rõ chiều cao và rộng của các hình ảnh trong các thẻ đánh dấu để trình duyệt có thể dành ra phần không gian trống cho nó. Nếu tham số chiều rộng và chiều cao được dùng trong các thẻ `` thì trình duyệt XHTML có thể dành riêng khoảng trống cho các hình ảnh trước khi chúng được tải về đầy đủ. Khi đó trang wap có thể được hiển thị trước khi tải hình ảnh về và nó sẽ được hiển thị khi đã tải về đầy đủ. Điều này sẽ không làm thay đổi thời gian tải và thời gian xử lý của 1 trang XHTML nhưng về căn bản nó giúp người dùng không phải chờ cho đến khi tải xong hình ảnh mới có thể đọc được nội dung của trang wap.

Ví dụ:

```

```

14. Sử dụng bảng một cách cẩn thận

Trình duyệt hỗ trợ việc sử dụng các bảng đơn và các bảng lồng vào nhau trong các trang XHTML. Các nhà phát triển nên cẩn thận khi xác định chiều rộng của ô dữ liệu đặc biệt đối với các bảng lồng vào nhau.

Khi sử dụng các bảng lồng nhau, ta nên tránh chỉ định chiều rộng của bảng cấp cha theo phần trăm khi chiều rộng của các bảng con được định kích thước rõ ràng. Bởi vì các thiết bị có kích cỡ màn hình hiển thị khác nhau nên tỷ lệ phần trăm không đảm bảo sẽ hiển thị cùng một số lượng pixel trên những thiết bị khác nhau. Do đó các nhà phát triển khuyến cáo nên sử dụng kích thước chính xác (theo pixel) cho chiều dài của cả bảng cấp cha và các bảng con lồng bên trong để đảm bảo nội dung sẽ được hiển thị chính xác. Cần phải cẩn thận để đảm bảo tổng chiều rộng của bảng bằng với tổng chiều rộng của các cột riêng biệt cộng với các đường viền và khoảng cách giữa các ô. Nói chung khi sử dụng các bảng với nhiều cấp lồng nhau sẽ làm tăng tính phức tạp cho trang wap và cần phải có nhiều thời gian xử lý để có

thể hiển thị nội dung lên màn hình. Để đảm bảo thời gian hiển thị ở mức chấp nhận được ta nên tránh sử dụng các bảng lồng nhau nhiều cấp, phức tạp.

Đồng thời các đường bao của bảng không nên quá dày bởi vì khi đó ta sẽ mất nhiều pixel để hiển thị đường bao làm cho phần hiển thị nội dung trở nên quá nhỏ.

15. Cần cân nhắc các tùy chọn

Ta có thể chỉ định kiểu định dạng theo nhiều cách khác nhau: trong một bảng định dạng riêng biệt, trong một phần tử định dạng trong phần <head> của tài liệu hoặc bằng cách sử dụng thuộc tính định dạng trong một phần tử xác định. Mặc dù theo nguyên tắc việc sử dụng các bảng định dạng riêng bên ngoài để tách biệt phần định dạng khỏi các thẻ đánh dấu là một cách làm tốt, nhưng cũng có những bất lợi cần phải cân nhắc.

Việc hiển thị một trang XHTML mà phần định dạng được chứa trong phần code của XHTML sẽ nhanh hơn nhưng việc sử dụng các bảng định dạng riêng bên ngoài cung cấp một cách thức tiện lợi để có thể thay đổi định dạng trên toàn thể dịch vụ. Một bảng định dạng cần phải được sử dụng cho toàn bộ hệ thống để tránh việc tải về thiết bị di động nhiều bảng định dạng. Bảng định dạng sẽ được tải về một lần duy nhất và được lưu trong bộ nhớ lưu trữ (cache).

16. Loại bỏ các khoảng trắng và các ghi chú trong phần code

Cần phải đảm bảo rằng không có khoảng trắng thừa trong phần code. Mặc dù khoảng trắng không được hiển thị lên màn hình nhưng nó vẫn được trình duyệt xử lý.

Số lượng ghi chú trong phần code XHTML nên được hạn chế tối đa để phần code được ngắn gọn.

17. Sử dụng các chỉ dẫn trong phần tiêu đề HTTP trong việc lưu trang

Bộ trình duyệt đặt các trang XHTML trong bộ nhớ lưu trữ; tuy nhiên, có thể các trang sẽ không được lưu trữ mặc định. Tiêu đề yêu cầu lưu trữ rõ ràng nên được gửi cùng với tài liệu để đảm bảo các trang được lưu trữ tại máy truy cập nếu có thể.

Thêm vào đó một qui định về thời gian hết hạn của các trang nội dung nên được thiết lập để đảm bảo các nội dung được lưu trữ trong một khoảng thời gian hợp lý.

Việc đặt các chỉ dẫn lưu trữ trong các thẻ meta (ví dụ dùng HTTP-EQUIV) sẽ không được hỗ trợ, nhưng việc lưu trữ có thể được điều khiển bằng các tiêu đề của giao thức HTTP. Chỉ dẫn “Cache-control:no-cache” có thể được thiết lập bởi máy chủ quản lý các trang nội dung để cho biết các trang không thể được lưu trữ tại máy truy cập.

Việc lưu trữ được thực hiện với thuật toán “least recently used”, nghĩa là các mục được sử dụng ít nhất sẽ bị xóa trước. Ta nên sử dụng cùng các hình và file CSS trong tất cả các trang XHTML để đảm bảo rằng nó luôn được lưu trữ và không phải tải về mỗi lần sử dụng chúng.

18. Sử dụng mã Unicode cho các nội dung XHTML

Trình duyệt XHTML hỗ trợ các bảng mã ASCII và Unicode 2.0. Do đó các nội dung XHTML thuộc nhóm các ngôn ngữ khác Latin nên sử dụng bảng mã Unicode để đảm bảo tính tương tác. Đối với các ngôn ngữ thuộc bộ mẫu tự Latin, có thể sử dụng bảng mã ASCII. Một vài gateway và proxy có thể chuyển đổi các bộ ký tự địa phương sang Unicode nhưng không phải tất cả đều có thể. Do đó cách duy nhất để đảm bảo các thiết bị truy cập nhận được các nội dung bằng mã Unicode ta nên dùng bảng Unicode cho các nội dung đó. Có thể tham khảo thêm các thông tin chi tiết về Unicode và các ngôn ngữ không thuộc hệ Latin ở các tài liệu:

CJKV Information Processing, Lunde, Ken. 1st edition. O'Reilly & Associates (December 1998)

Unicode: A Primer, Graham, Tony. John Wiley & Sons (March 2000)

19. Sử dụng chính xác các kiểu MIME và mã XHTML

Kiểu MIME được đề cập cho các nội dung XHTML MP, được đặc tả bởi OMA, là “application/vnd.wap.xhtml+xml”. Ta nên sử dụng kiểu này để cung cấp các tài liệu XHTML MP đến người dùng cuối. Thêm vào đó cũng có thể dùng kiểu

“application/xhtml+xml”. Trong một số trình duyệt thuộc Series 60, cần phải dùng kiểu “application/vnd.wap.xhtml+xml” để đảm bảo các nội dung XHTML được hiển thị đúng.

Ta được khuyến cáo nên sử dụng phần đuôi/mở rộng của file là *.xhtml cho tất cả các nội dung XHTML MP. Mã XHTML nên được kiểm tra hợp lệ để tránh các vấn đề về khả năng giao tiếp và để tăng hiệu suất của dịch vụ. Các nội dung XHTML có thể được kiểm tra hợp lệ bằng bộ kiểm tra của W3C tại địa chỉ <http://validator.w3.org>. Và khi tạo các nội dung XHTML động thì phần mã được phát sinh nên phải căn cứ vào tài liệu DTD XHTML MP 1.0.

20. Các tiêu đề chỉ dẫn và các nhãn phần tử

Tiêu đề của trang mô tả nội dung của trang đang được hiển thị. Việc sử dụng các tiêu đề được khuyến cáo trong WML và bắt buộc đối với các nội dung XHTML. Các tiêu đề giúp người dùng các thẻ định hướng trong ứng dụng bởi vì chúng nhắc người dùng đang ở vị trí nào trong ứng dụng. Việc bắt đầu tiêu đề với tên của dịch vụ là một ý tưởng tốt và giữ cho tiêu đề ngắn gọn. Đối tượng được chọn nên xác định tiêu đề của trang tiếp theo. Ví dụ, tiêu đề “Bookmarks” cho người dùng biết nội dung đang hiển thị là một danh sách các bookmark trong ứng dụng và đối tượng được chọn trước đó là Bookmarks.

Nên sử dụng các dạng font cân đối cho tiêu đề và nếu tiêu đề quá dài nó sẽ tự động bị cắt đi. Các tiêu đề bị cắt bớt được dùng thay vì dùng các từ viết tắt để tránh làm cho người dùng bối rối.

Mặc dù được khuyến cáo nên sử dụng các từ ngắn cho phần nhãn của các đối tượng nhưng các nhóm từ viết tắt không thông dụng trong cộng đồng người dùng nên hạn chế sử dụng. Các nhãn cùng loại nên được dùng cho các đối tượng có cùng chức năng đặc biệt là các nhãn có chức năng như: Delete, Remove, Erase, Clear và Destroy.

21. Thực hiện kiểm tra khả năng sử dụng của hệ thống

Ta nên thực hiện các thử nghiệm kiểm tra khả năng sử dụng của các ứng dụng mới. Các thử nghiệm đó nên được thực hiện càng sớm càng tốt trong quá trình phát triển. Bất cứ các thay đổi cần thiết nào đạt được từ các thử nghiệm sau đó có thể được đưa vào bảng kế hoạch thời gian phát triển. Những người thử nghiệm là những đại diện cho người dùng trong tương lai. Ít nhất các thử nghiệm nên được thực hiện ở qui mô nhỏ nếu không đủ khả năng thực hiện thử nghiệm đầy đủ.

KHOA CNTT

PHỤ LỤC F
DANH SÁCH CÁC THUẬT NGỮ

ACL - Access Control Lists
ADSL - Asymmetric Digital Subscriber Line
CDMA - Code Division Multiple Access
CHTML - Ngôn ngữ đánh dấu dùng cho iMode
CSS - Cascading Style Sheets
DNS - Domain Name Server
DTD - Document Type Definition
FTP - File Transfer Protocol
GSSN - Gateway GPRS Support Node
HDML - Handheld Device Markup Language
HTML - Hyper Text Markup Language
HTTP - Hypertext Transfer Protocol
iMode - Dịch vụ internet di động của NTTDoCoMo nhằm mục đích tạo ra các trang web sử dụng một dạng ngôn ngữ thuộc họ HTML (cHTML) hiển thị trên các trình duyệt Web di động
IP - Internet Protocol
ISP - Internet Service Provider
LDAP - Lightweight Directory Access Protocol
MAC - Media Access Control
MIME - Multipurpose Internet Mail Extension
NAT - Network Address Translation
PDA - Personal Digital Assistant
PPP - Point-to-Point Protocol
RAS - Remote Access Server
SGML - Standard Generalize Markup Language
SMS - ShortMessage
SSL - Secure Sockets Layer
TCP - Transmission Control Protocol
TCP/IP - Transmission Control Protocol/Internet Protocol

TLS - Transport Layer Security
UDP - User Datagram Protocol
URL - Uniform Resource Locator
W3C - World Wide Web Consortium
WAE - Wireless Application Environment
WBMP - Wireless Bitmap
WAP - Wireless Application Protocol
WDP - Wireless Datagram Protocol
WML - Wireless Markup Language
WML Script - Wireless Markup Language Script
WPS - Wireless Protocol Stack
WSP - Wireless Session Protocol
WTA - Wireless Telephony Application
wTCP/IP - Wireless Profile TCP/IP
WTLS - Wireless Transport Security Layer
WTP - Wireless Transaction Protocol
XHTML - Extensible Hyper Text Markup Language
XHTML MP - Extensible Hyper Text Markup Language Mobile Profile
XML - Extensible Markup Language
XSLT - Extensible Stylesheet Language Transformation