สำนักหอสมุดกลาง พระจอมเกล้าฉกระบัง

แว็พบราวเซอร์ภาษาไทย

Thai WAP Browser

นายคมสันต์ ทิวงศ์
นายบุญร่วม ศิริพันธ์

2541
A152 ส
2543

5

เลขหมู่.....
เลขทะเบียน.....42835
วัน, เดือน, ปี.....10 มิ.ย. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

แว็พบราวเซอร์ภาษาไทย

Thai WAP Browser

โดย

นายคมสันต์ ทิวงศ์

นายบุญร่วม ศิริพันธ์

อาจารย์ที่ปรึกษา

ดร.วิศิษฏ์ หิรัญกิตติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

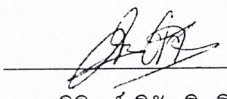
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แว็พบราวเซอร์ภาษาไทย

Thai WAP Browser

ผู้จัดทำ

1. นาย คมสันต์ ทิวงศ์ รหัสประจำตัว 41013522
2. นาย บุญร่วม ศิริพันธ์ รหัสประจำตัว 41013537



อาจารย์ที่ปรึกษา

(ดร. วิศิษฏ์ หิรัญกิตติ)

แว็พบราวเซอร์ภาษาไทย

นายคมสันต์ ทิวงศ์ รหัส 41013522
นายบุญร่วม ศิริพันธ์ รหัส 41013537
ดร. วิศิษฏ์ หิรัญกิตติ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

แว็พบราวเซอร์ภาษาไทยได้ถูกสร้างขึ้นมาโดยมีฟังก์ชันการทำงานหลักคือ สามารถเข้าไปชมแว็พไซต์ต่างๆ ที่ต้องการ ให้ผู้ใช้สามารถเขียนโปรแกรมภาษา WML และสามารถดูผลการทำงานโดยจำลองการทำงานของโทรศัพท์มือถือซึ่งสามารถแสดงผลเป็นภาษาไทยในขณะที่แว็พบราวเซอร์อื่นๆ ไม่รองรับการแสดงผลภาษาไทย

ปริญญานิพนธ์นี้ได้สร้างขึ้นมาด้วยวิธีการใช้หลักการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML และสร้างขึ้นมาจากการเขียนโปรแกรมภาษาจาวาโดยใช้เทคนิคการแปลงไฟล์เอกสาร WML ให้เป็นไฟล์เอกสาร HTML ในการแสดงผลบนคอมโพเนนต์ของภาษาจาวาที่สนับสนุนการแสดงผลภาษา HTML เพื่อให้การแสดงผลมีความถูกต้องตามหลักไวยากรณ์ภาษา WML มากที่สุด

Thai WAP Browser

Komsant Tivong

Boonruam Siripun

Dr. Visit Hirankitti Advisor

ABSTRACT

Thai WAP browser in this project is built to support Thai mobile phone users. Main functions of this browser are able to browse WAP pages, provide a tool for WAP developers to write WML codes, and emulate the browser as that in the real mobile phone, but this one is on PC. This browser is able to support Thai language whilst other WAP browsers available are not.

This browser is designed and developed by object-oriented software development method using UML approach and JAVA. The technique in this project to handle the displaying of WML codes is by means of conversion of WML to HTML content and then using JAVA component to interpret and display the converted HTML. Our browser can handle most of the WML tags.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้สำเร็จได้ก็คือ ดร.วิศิษฏ์ หิรัญกิตติ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบคุณเป็นอย่างมาก และขอขอบคุณเพื่อนๆ ทุกคนในห้อง ICL ที่ให้กำลังใจในยามที่ท้อแท้

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ และเอาใจใส่ข้าพเจ้ามาตลอด ในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

คมสันต์ ทิวงศ์

บุญร่วม ศิริพันธ์

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 ความรู้เบื้องต้นเกี่ยวกับแว็พ	3
2.1.1 ความเป็นมาของแว็พ	3
2.1.2 ข้อกำหนดมาตรฐานแว็พ	4
2.1.3 การทำงานและโครงสร้างของแว็พ	7
2.1.4 การกำหนด MIME Type เพื่อให้เซิร์ฟเวอร์รองรับแว็พได้	10
2.1.5 การเขียนโปรแกรมภาษา WML	11
2.2 ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมด้วยภาษาจาวา	15
2.2.1 คอมไพเลอร์ภาษาจาวาเวอร์ชัน 1.2.2 ที่สนับสนุนภาษาไทย	15
2.2.2 API ต่างๆ ในภาษาจาวาที่รองรับการเขียนโปรแกรมบนเครือข่าย	16
2.3 ความรู้เบื้องต้นเกี่ยวกับการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML	16
2.3.1 แผนภาพยูสเคส (Use Case)	16
2.3.2 แผนภาพแสดงลำดับขั้นตอนการทำงาน	18
2.3.3 แผนภาพแสดงการทำงานร่วมกันของวัตถุ	18
2.3.4 แผนภาพคลาส	19
2.3.5 แผนภาพสถานะ	20
บทที่ 3 การสร้างและการออกแบบ	21
3.1 คอนเท็กซ์ไดอะแกรมของแว็พบราวเซอร์ภาษาไทย	21
3.2 ยูสเคสไดอะแกรมของแว็พบราวเซอร์ภาษาไทย	22
3.3 ลำดับขั้นตอนการทำงานของแว็พบราวเซอร์ภาษาไทย	22

สารบัญ (ต่อ)

หน้าที่

3.3.1 ลำดับขั้นตอนการทำงานของการเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการ 22

3.3.2 ลำดับขั้นตอนการทำงานของการเขียนโปรแกรมภาษา WML บนแว็พบราวเซอร์ภาษาไทย 23

3.3.3 ลำดับขั้นตอนการทำงานของการปรับแต่งแว็พบราวเซอร์ภาษาไทย 24

3.4 การทำงานร่วมกันระหว่างวัตถุของแว็พบราวเซอร์ภาษาไทย 24

3.4.1 การทำงานร่วมกันระหว่างวัตถุของการเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการ 24

3.4.2 การทำงานร่วมกันระหว่างวัตถุของการเขียนโปรแกรมของผู้ใช้บนแว็พบราวเซอร์ภาษาไทย 25

3.4.3 การทำงานร่วมกันระหว่างวัตถุของการปรับแต่งแว็พบราวเซอร์ภาษาไทย 25

3.5 คลาสต่างๆ ของแว็พบราวเซอร์ภาษาไทย 26

3.6 สถานะของแว็พบราวเซอร์ภาษาไทย 26

3.7 คลาสต่างๆ ที่สร้างขึ้นในโปรแกรมแว็พบราวเซอร์ภาษาไทย 28

3.7.1 ส่วนแสดงผลไฟล์เอกสาร WML 28

3.7.2 ส่วนแสดงผลบนโทรศัพท์มือถือจำลอง 29

3.7.3 ส่วนปรับแต่งแว็พบราวเซอร์ภาษาไทย 30

3.7.4 ส่วนรับ URL จากผู้ใช้ 30

3.7.5 ส่วนเก็บแท็กต่างๆ ของภาษา WML 30

3.7.6 ส่วนรับการคอมไพล์จากผู้ใช้ 31

3.7.7 ส่วน WAP Parser 31

3.8 การสร้างแว็พบราวเซอร์ภาษาไทย 33

บทที่ 4 ผลการทดสอบและปัญหาที่เกิดขึ้น 37

4.1 ผลการทดสอบจากการที่ผู้ใช้เข้าไปชมแว็พไซต์ตาม URL ที่ต้องการ 37

4.1.1ผลการทดสอบในส่วนแสดงผลไฟล์เอกสาร WML 37

4.1.2 ผลการทดสอบการแสดงผลบนหน้าจอโทรศัพท์มือถือจำลอง 38

4.2 ผลการทดสอบในการเขียนโปรแกรมของผู้ใช้ 39

4.2.1 ผลการทดสอบการเขียนโปรแกรมภาษา WML 39

4.2.2 ผลการทดสอบในโทรศัพท์มือถือจำลอง 40

สารบัญ (ต่อ)

	หน้าที่
4.3 ปัญหาที่เกิดขึ้นในการแสดงตัวอักษรภาษาไทยแบบต่างๆ ในแว็พบราวเซอร์ภาษาไทย	40
4.4 ปัญหาที่เกิดขึ้นในการแสดงตัวอักษรภาษาอังกฤษในแบบต่างๆ ในแว็พบราวเซอร์ภาษาไทย	41
บทที่ 5 บทวิเคราะห์และสรุป	42
5.1 สรุปและวิเคราะห์ผลการทดสอบ	42
5.2 แนวทางการพัฒนา	43
ภาคผนวก ก.	44
ภาคผนวก ข.	47
บรรณานุกรม	85

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 การกำหนด MIME Type ให้กับเซิร์ฟเวอร์ 11

ตารางที่ 2-2 แสดงแท็กต่างๆ ของภาษา WML 15

ตารางที่ 3-1 แท็กของภาษา WML ที่ถูกแปลงเป็น HTML 33

สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 แสดงแบบจำลองสำหรับการเขียนโปรแกรม 4
รูปที่ 2-2 แสดงโครงสร้างจริงของเครือข่ายสื่อสารแว็พ 5
รูปที่ 2-3 สถาปัตยกรรมของแว็พ 8
รูปที่ 2-4 แสดงการใช้งานโปรแกรมแว็พบราวเซอร์และเครื่องโทรศัพท์มือถือ
ในการชมแว็พไซต์ 9
รูปที่ 2-5 โครงสร้างของแว็พเกตเวย์ 10
รูปที่ 2-6 เด็คและการ์ดของภาษา WML 12
รูปที่ 2-7 แสดงผลลัพธ์ของไฟล์เอกสาร WML 13
รูปที่ 2-8 แผนภาพยูสเคส 17
รูปที่ 2-9 แผนภาพแสดงขั้นตอนการทำงาน 18
รูปที่ 2-10 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุ 19
รูปที่ 2-11 แผนภาพคลาสของเส้นและจุด 20
รูปที่ 2-12 แผนภาพแสดงสถานะของคีย์บอร์ด 20
รูปที่ 3-1 แผนภาพคอนเท็กซ์ของแว็พบราวเซอร์ภาษาไทย 21
รูปที่ 3-2 แสดงยูสเคสของแว็พบราวเซอร์ภาษาไทย 22
รูปที่ 3-3 แผนภาพลำดับขั้นตอนการทำงานของการเข้าชมแว็พไซต์ต่างๆ
ตาม URL ที่ต้องการ 23
รูปที่ 3-4 แผนภาพแสดงลำดับขั้นตอนการทำงานของการเขียนโปรแกรมภาษา WML
บนแว็พบราวเซอร์ภาษาไทย 23
รูปที่ 3-5 แผนภาพแสดงลำดับขั้นตอนการทำงานของการปรับแต่งแว็พบราวเซอร์
ภาษาไทย 24
รูปที่ 3-6 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการเข้าชมแว็พไซต์ต่างๆ
ตาม URL ที่ต้องการ 24
รูปที่ 3-7 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการเขียนโปรแกรม
บนแว็พบราวเซอร์ภาษาไทย 25
รูปที่ 3-8 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการปรับแต่ง
แว็พบราวเซอร์ภาษาไทย 25
รูปที่ 3-9 แผนภาพคลาสของแว็พบราวเซอร์ภาษาไทย 26
รูปที่ 3-10 แผนภาพสถานะของแว็พบราวเซอร์ภาษาไทย 27

สารบัญรูปภาพ (ต่อ)

หน้าที่

รูปที่ 3-11 ความสัมพันธ์ระหว่าง JTextArea และ JScrollPane 28
รูปที่ 3-12 ส่วนแสดงผลไฟล์เอกสาร WML ที่ได้สร้างขึ้น 28
รูปที่ 3-13 แสดงความสัมพันธ์ระหว่าง JEditorPane และ JScrollPane 29
รูปที่ 3-14 ส่วนแสดงผลบนโทรศัพท์มือถือจำลอง 29
รูปที่ 3-15 ผลจากคลาส JFontChooser ถูกเรียกใช้งาน 30
รูปที่ 3-16 ส่วนการรับ URL 30
รูปที่ 3-17 แสดงความสัมพันธ์ระหว่าง JButton และ JToolBar 31
รูปที่ 3-18 แสดงปุ่มของแท็กต่างๆ บนแว็พบราวเซอร์ภาษาไทย 31
รูปที่ 3-19 แสดงส่วนรับการคอมไพล์จากผู้ใช้ 31
รูปที่ 3-20 โครงสร้างการแปลง (Conversion) ของ WAP Parser 32
รูปที่ 3-21 แสดงผลการรัน JavaWebServer 2.0 อย่างถูกต้อง 33
รูปที่ 3-22 โปรแกรม Jbuilder 3.0 35
รูปที่ 3-23 แสดงการเซตพรอพอตี้ (Propoties) ของโปรแกรม Jbuilder 3.0 35
รูปที่ 3-24 แสดงการเซตคอมไพเลอร์โปรแกรม Jbuilder 3.0 ให้เป็น JDK1.2.2 ภาษาไทย 36
รูปที่ 4-1 ผลลัพธ์หลังจากที่ผู้ใช้ใส่ URL ตามที่ต้องการ 37
รูปที่ 4-2 รูปขยายการแสดงผลไฟล์เอกสาร WML ที่ได้จาก URL 38
รูปที่ 4-3 ผลลัพธ์จากหน้าจอแสดงผลของไฟล์ WML ที่ได้จาก URL 38
รูปที่ 4-4 แสดงไฟล์ WML ที่ผู้ใช้เขียนโปรแกรมขึ้นมา 39
รูปที่ 4-5 ผลลัพธ์ที่ได้จากการคอมไพล์โปรแกรมที่ผู้ใช้ทดลองเขียนขึ้นมา 40
รูปที่ 4-6 ปัญหาที่เกิดขึ้นในการแสดงผลภาษาไทยจากการกำหนดรูปแบบตัวอักษร
แบบต่างๆ 40
รูปที่ 4-7 ปัญหาที่เกิดขึ้นในการแสดงผลภาษาอังกฤษจากการกำหนดรูปแบบตัวอักษร
แบบต่างๆ 41

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันความเจริญก้าวหน้าทางด้านการสื่อสารโทรคมนาคมทำให้การเชื่อมต่อโลกของโทรศัพท์เคลื่อนที่เข้ากับเครือข่ายอินเตอร์เน็ตสามารถทำได้โดยการร่วมมือกันของกลุ่ม WAP Forum และได้ตั้งมาตรฐานการสื่อสารสากลที่เรียกว่า WAP (Wireless Application Protocol) ซึ่งทำให้อุปกรณ์สื่อสารไร้สายต่างๆ สามารถเชื่อมโยงเข้ากับเครือข่ายอินเตอร์เน็ตได้โดยมีการกำหนดสภาวะแวดล้อมของการประยุกต์ใช้งานที่เรียกว่า WAE (Wireless Application Environment) โดยมีวัตถุประสงค์เพื่อให้ผู้ให้บริการบนระบบเครือข่าย ผู้ผลิตอุปกรณ์สื่อสารไร้สาย และบริษัทซึ่งทำหน้าที่ให้บริการข่าวสารข้อมูลได้ใช้เป็นมาตรฐานเดียวกันโดยมีการกำหนดภาษา WML เป็นภาษาที่ใช้ในการแสดงข้อมูลบนอุปกรณ์สื่อสารแบบไร้สาย

สำหรับนักพัฒนาเว็พในการให้บริการข้อมูลต่างๆ กับผู้ใช้บริการนั้นซึ่งจะเขียนเว็พไซต์ให้กับผู้ใช้บริการด้วยภาษา WML ทำให้อุปกรณ์สื่อสารไร้สายสามารถรับข้อมูลข่าวสารต่างๆ บนเครือข่ายอินเตอร์เน็ตได้โดยนักพัฒนาเว็พจะใช้เครื่องมือในการพัฒนาโดยปัจจุบันมีหลายบริษัทเช่น ของบริษัทโนเกีย (Nokia) เป็นต้น โดยการใช้เครื่องคอมพิวเตอร์จำลองการทำงานของโปรแกรมที่เขียนขึ้นและสามารถรับข้อมูลที่มาจากเว็พไซต์ต่างๆ มาแสดงผลได้ซึ่งจำลองการทำงานของอุปกรณ์สื่อสารไร้สายออกมา เรียกว่าเว็พบราวเซอร์ (WAP browser) ซึ่งนักพัฒนาจะใช้เครื่องมือเหล่านี้ในการพัฒนาโปรแกรมในการให้บริการข้อมูลต่างๆกับผู้ใช้ซึ่งการใช้เว็พบราวเซอร์นั้นจะเป็นประโยชน์ทั้งนักพัฒนาเว็พ และบุคคลที่ไม่มีอุปกรณ์สื่อสารไร้สายที่สนับสนุนการทำงานของเว็พและบุคคลที่ไม่มีอุปกรณ์สื่อสารไร้สายได้มีโอกาสได้รับข้อมูลข่าวสารต่างๆในส่วนของเว็พไซต์

โดยเว็พบราวเซอร์ต่างๆที่นักพัฒนาส่วนใหญ่ใช้ในการพัฒนาเว็พไซต์ในปัจจุบันถูกเขียนขึ้นมาโดยการสนับสนุนภาษาสากลอย่างเช่น ภาษาอังกฤษ เป็นต้น ซึ่งทำให้เว็พไซต์ต่างๆ ในปัจจุบันเป็นภาษาสากลไปด้วย ซึ่งทำให้นักพัฒนาเว็พซึ่งต้องการให้บริการข้อมูลต่างๆกับผู้ใช้บริการที่เป็นคนไทยทำได้ยากเนื่องจากเว็พบราวเซอร์เหล่านี้ไม่สนับสนุนอักษรภาษาไทย

ด้วยเหตุผลดังกล่าวปริญญานิพนธ์เล่มนี้จึงได้เกิดขึ้นเพื่อเป็นการรองรับเทคโนโลยีเว็พในการสนับสนุนการพัฒนาเว็พไซต์ต่างๆที่เป็นเว็พไซต์ภาษาไทยที่กำลังจะเกิดขึ้นอันเนื่องมากจากความนิยมที่กำลังเกิดขึ้นในปัจจุบันเพื่อสนับสนุนการบริการกับผู้ใช้บริการที่คนเป็นไทยซึ่งคาดว่าเทคโนโลยีเว็พจะเติบโตขึ้นในระยะเวลาอันใกล้นี้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาโครงสร้างทางเทคโนโลยีและการทำงานของแว็พ
2. เพื่อศึกษาโครงสร้างภาษาและไวยากรณ์ของภาษา WML
3. เพื่อสร้างแว็พบราวเซอร์ภาษาไทย ที่สามารถแสดงผลภาษาไทยได้
4. เพื่อเป็นแนวทางในการพัฒนาแว็พบราวเซอร์ภาษาไทยต่อไปได้

1.3 ขอบเขตของปริญญานิพนธ์

1. ในส่วนบราวเซอร์สามารถแสดงผลได้ถูกต้องตามหลักการของไวยากรณ์ภาษา WML
2. ในส่วนของอิดิตเตอร์สามารถทำให้เขียนโปรแกรมภาษา WML และทำการแสดงผลลัพธ์จากโปรแกรมที่ผู้ใช้เขียนขึ้นมาบนบราวเซอร์ได้
3. ในส่วนบราวเซอร์นั้นสามารถแสดงผลข้อมูลที่เป็นภาษาไทยได้
4. ในส่วนอิดิตเตอร์สามารถรองรับการเขียนโปรแกรมภาษา WML ที่มีข้อมูลเป็นภาษาไทยได้

1.4 วิธีการดำเนินงาน

1. ศึกษาโพรโตคอลของแว็พเพื่อให้เข้าใจการทำงานและโครงสร้างของแว็พโดยการหาข้อมูลจาก WAP Forum
2. ศึกษาวิธีการติดตั้งเซิร์ฟเวอร์เพื่อให้สามารถรองรับภาษา WML ได้
3. สำรวจหาเครื่องมือต่างๆที่มีเพื่อใช้ในการเขียนและพัฒนาภาษา WML
4. ศึกษาไวยากรณ์ภาษา WML และทดลองเขียนโปรแกรมขึ้นโดยใช้บราวเซอร์ที่มีอยู่แล้ว และทำการตรวจสอบผลลัพธ์ในส่วนการแสดงผล
6. รวบรวมข้อมูลเพื่อนำมาออกแบบโดยใช้ภาษา UML ในการวิเคราะห์
7. ทำการเขียนโปรแกรมตามที่ได้ออกแบบไว้โดยการใช้ภาษาจาวาในการเขียนโปรแกรม
8. ทดสอบการทำงานของโปรแกรม เพื่อให้ครอบคลุมการแสดงผลที่ถูกต้อง
9. ปรับปรุงการทำงานของโปรแกรมให้สมบูรณ์ยิ่งขึ้น
10. เตรียมรายงาน เอกสาร คำอธิบายโปรแกรมและปริญญานิพนธ์

บทที่ 2
หลักการและทฤษฎี

2.1 ความรู้เบื้องต้นเกี่ยวกับแว็พ

2.1.1 ความเป็นมาของแว็พ

แว็พ (WAP: Wireless Application Protocol) เป็นตัวกลางสำคัญในการเชื่อมต่อโลกของโทรศัพท์เคลื่อนที่เข้ากับเครือข่ายอินเตอร์เน็ต ซึ่งในบางกรณีก็สามารถประยุกต์ใช้กับเครือข่ายอินทราเน็ตภายในองค์กร จุดประสงค์เพื่อขยายขอบเขตในการให้บริการเสริมให้กับผู้ใช้บริการโทรศัพท์เคลื่อนที่โดยไม่คำนึงถึงประเภทของเครือข่ายบริการ และเครื่องลูกข่ายแต่อย่างใด ผู้ใช้บริการสามารถดึงข้อมูลจากเครือข่ายอินเตอร์เน็ตได้จากโทรศัพท์เคลื่อนที่ของตน โดยตัวเครื่องอาจมีขนาดเท่าฝ่ามือแต่มีขีดความสามารถในการทำงานเทียบเท่ากับเครื่องคอมพิวเตอร์ส่วนบุคคล

แว็พเป็นมาตรฐานสื่อสารสากลซึ่งเป็นมาตรฐานที่สร้างขึ้นจากความนิยม หรือที่เรียกว่า de-facto Standard ซึ่งมิได้ถูกออกแบบกำหนดและควบคุมโดยบริษัทใดบริษัทหนึ่งแต่เป็นผลมาจากการร่วมกันวางข้อกำหนดระหว่างบริษัทอีริคสัน โนเกีย โมโตโรลา และบริษัทอันไวร์ แพลเน็ต ซึ่งร่วมกันก่อตั้งองค์กรที่มีชื่อเรียกว่า WAP Forum ขึ้นในปี พ.ศ. 2540 จุดมุ่งหมายในเบื้องต้นก็คือการวางข้อกำหนดทางอุตสาหกรรมสำหรับสนับสนุนการพัฒนาบริการพิเศษผ่านเครือข่ายสื่อสารไร้สาย ข้อกำหนดแว็พเป็นการระบุถึงกลุ่มโพรโตคอลหรือข้อกำหนดทางการสื่อสารที่มีใช้งานในระดับชั้นต่างๆ ตามแบบจำลอง OSI ซึ่งช่วยทำให้ผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ ผู้ผลิตอุปกรณ์สื่อสาร และบริษัททำหน้าที่วิจัยและพัฒนาบริการพิเศษสามารถร่วมมือกันสร้างบริการพิเศษไม่จำกัดรูปแบบ ผ่านเครือข่ายสื่อสารไร้สายในปัจจุบันมีองค์การต่างๆ ซึ่งเป็นตัวแทนทั้งจากผู้ให้บริการเครือข่าย ผู้ผลิตอุปกรณ์สื่อสาร ผู้ให้สัมปทาน บริษัทผู้ผลิตซอฟต์แวร์ ผู้ขายสื่อข้อมูลต่างๆ มากกว่า 1,000 รายเข้าร่วมเป็นสมาชิกของ WAP Forum เพื่อผลักดันมาตรฐานแว็พให้มีการประยุกต์ใช้งานจริงในเครือข่ายโทรศัพท์เคลื่อนที่ทั่วโลก ซึ่งสามารถเข้าชมเว็บไซต์ของ WAP Forum ได้ที่ http://www.wapforum.org

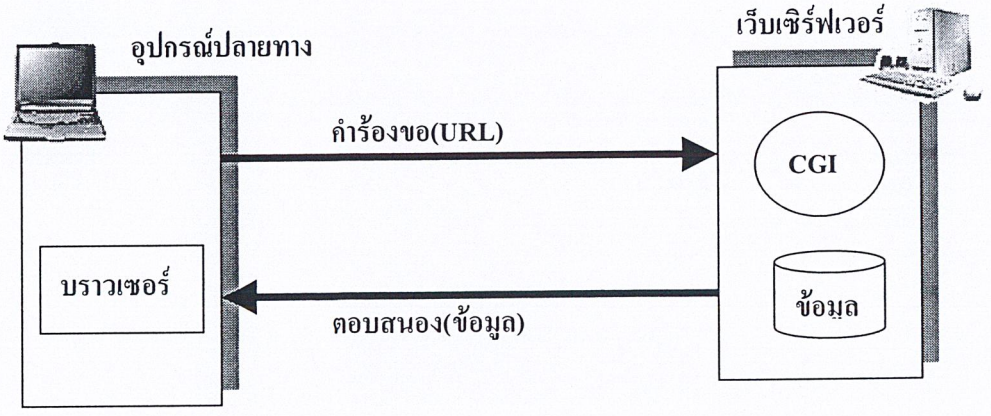
นอกจากนี้ข้อกำหนดแว็พยังมีการกล่าวถึงสภาวะแวดล้อมของการประยุกต์ใช้งาน (Wireless Application Environment) หรือ WAE โดยมีจุดประสงค์เพื่อให้บรรดาผู้ให้บริการระบบเครือข่าย ผู้ผลิตอุปกรณ์สื่อสาร และบริษัทซึ่งทำหน้าที่ให้บริการข่าวสารข้อมูล ได้มีโอกาสในการสร้างบริการพิเศษให้กับผู้ใช้บริการของตน โดยสามารถสร้างบริการเสริมพิเศษที่มีความแตกต่างจากคู่แข่งของตน ข้อกำหนด WAE มีการระบุถึงเรื่องของไมโครเบราเซอร์ (Microbrowser) การเขียนสคริปต์ อิเล็กทรอนิกส์เมล์ บริการแลกเปลี่ยนข่าวสารระหว่างเว็บไซต์กับเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ (World Wide Web-to-mobile-handset messaging) และบริการรับส่งเทเลแฟกซ์ผ่านโทรศัพท์เคลื่อนที่ ซึ่งทั้งนี้จะเห็นว่าการเชื่อมต่อระหว่างเครือข่ายทั้งสองนี้กระทำผ่านอุปกรณ์ที่มีชื่อว่า WAP Gateway เมื่อเปรียบเทียบกับแนวคิดของเทคโนโลยีอินเตอร์เน็ตตามปกติแล้ว เครื่องลูกข่ายโทรศัพท์เคลื่อนที่จะทำหน้าที่เป็นเว็บไมโครเบราเซอร์

ในขณะที่ข่าวสารและโปรแกรมประยุกต์ใช้งานต่างๆ ก็ยังคงอยู่บนเว็บเซิร์ฟเวอร์ในเครือข่ายอินเตอร์เน็ตดังเช่นเดิม

2.1.2 ข้อกำหนดมาตรฐานแว็พ

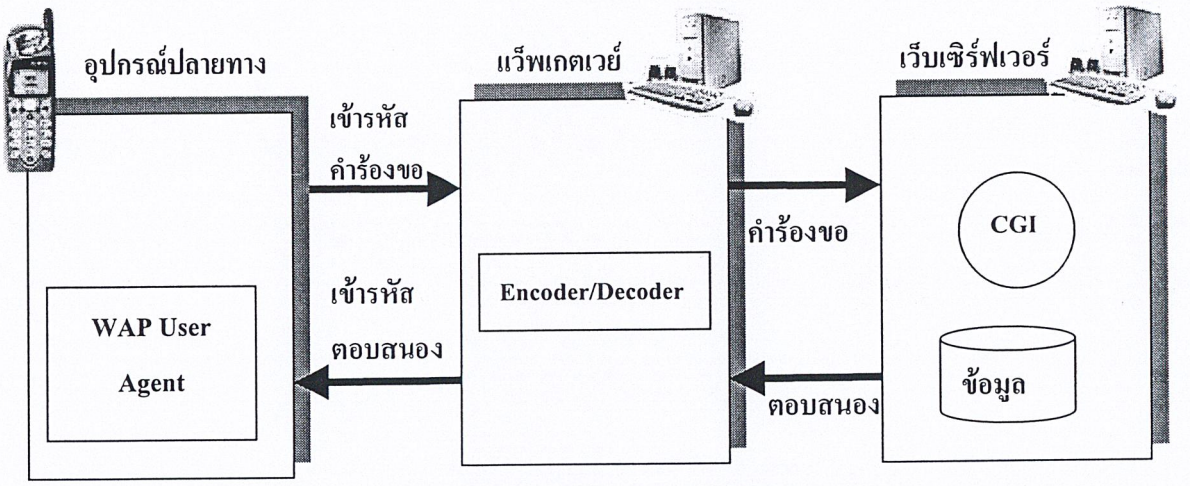
ความสำเร็จของข้อกำหนดมาตรฐานแว็พมีผลต่อเนื่องมาจากการที่มาตรฐานดังกล่าวได้รับการจัดวางให้เป็นข้อกำหนดแบบเปิดมาตั้งแต่แรก ผนวกกับการเลือกใช้โพรโตคอลและโครงสร้างของสถาปัตยกรรมระบบสัญญาณที่เป็นมาตรฐานสากลอยู่ก่อนแล้ว ทั้งหมดนี้ก็เพื่อให้สามารถรองรับการใช้อุปกรณ์สื่อสารไร้สายในการเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต นอกจากนั้นข้อกำหนดมาตรฐานแว็พยังเปิดทางออกให้กับปัญหาต่างๆ ที่ยังไม่ได้รับการแก้ไขจากองค์การสื่อสารอื่นๆ ไม่ว่าจะเป็น W3C, ETSI, TIA หรือ IETF และด้วยการที่แว็พมีฐานะเป็นตัวเร่งให้เกิดความสนใจในการพัฒนาและกำหนดมาตรฐานการสื่อสารไร้สายอื่นๆ ส่วนประกอบที่สำคัญของมาตรฐานแว็พประกอบไปด้วยสิ่งต่างๆ ต่อไปนี้

2.1.2.1 ข้อกำหนดของแบบจำลองสำหรับใช้ในการพัฒนาโปรแกรมแบบแว็พ ดังแสดงในรูปที่ 2-1 ซึ่งแสดงตามรายละเอียดทางเทคนิคของแบบจำลอง มาตรฐานการกำหนดแบบจำลองดังกล่าวก่อให้เกิดประโยชน์ต่อกลุ่มนักพัฒนาโปรแกรมเป็นอันมาก ไม่ว่าจะเป็นเรื่องของแบบจำลองที่ไม่แตกต่างไปจากมาตรฐานการเขียนโปรแกรมสำหรับใช้งานกับเครือข่ายอินเตอร์เน็ตที่เคยมีอยู่ก่อนแล้ว การออกแบบโครงสร้างสำหรับเขียนโปรแกรมที่ได้รับการยอมรับ และความเป็นไปได้ในการจัดหาอุปกรณ์หรือเครื่องมือในการเขียนโปรแกรมในตระกูล XML นอกจากนี้ทาง WAP Forum ยังได้กำหนดความเหมาะสมและเปิดทางเลือกเพิ่มเติมในตัวของแบบจำลอง เพื่อให้สามารถนำไปใช้งานได้อย่างมีประสิทธิภาพกับสภาพการสื่อสารไร้สายผ่านเครือข่ายแต่ละประเภท ไม่ว่าจะเป็นเครือข่ายวิทยุติดตามตัวแบบ 2 ทิศทาง มาตรฐานการสื่อสารและการเขียนโปรแกรมที่ใช้งานอยู่ก่อนแล้วจะถูกนำมาปรับปรุงหรือนำมาใช้งานโดยตรงกับข้อกำหนดแว็พ อยู่ตลอดเวลา



รูปที่ 2-1 แสดงแบบจำลองแว็พสำหรับการเขียนโปรแกรม

ข้อกำหนดเทคโนโลยีแว็พมีความคล้ายคลึงกับเทคโนโลยีเบราเซอร์บนเครือข่ายอินเตอร์เน็ต ซึ่งความแตกต่างอันเกิดจากคุณลักษณะทางกายภาพ และอัตราเร็วในการรับส่งข้อมูลของเครือข่ายโทรศัพท์เคลื่อนที่เท่านั้น หากจะมองโครงสร้างของการเชื่อมต่อตามเทคโนโลยีแว็พให้เป็นภาพของการสืบค้นข้อมูลในเครือข่ายอินเตอร์เน็ตก็จะเป็นไปตามรูปที่ 2-1 ซึ่งผู้ใช้บริการอินเตอร์เน็ตคือผู้ใช้อุปกรณ์ปลายทางทำการส่งคำสั่งขอดาวน์โหลดข้อมูลจากเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ซึ่งอยู่ที่ใดที่หนึ่งในโลก สิ่งที่ผู้ใช้บริการร้องขอจะได้รับการตอบรับจากเครื่องคอมพิวเตอร์เซิร์ฟเวอร์โดยมีการส่งเนื้อหา (Content) ตอบกลับมายังผู้บริการในโลกของการสื่อสารอินเตอร์เน็ตเครื่องคอมพิวเตอร์ของผู้ใช้บริการจะมีการติดตั้งโปรแกรมเบราเซอร์ ในขณะที่เครื่องคอมพิวเตอร์เซิร์ฟเวอร์มีสคริปต์ CGI ทำงานอยู่ เมื่อมองกลับมาถึงโครงสร้างการเชื่อมต่อที่แท้จริงของเทคโนโลยีแว็พตามรูปที่ 2-2 แล้วจะพบว่ามีอุปกรณ์เกตเวย์แว็พคั่นกลางอยู่ระหว่างอุปกรณ์ปลายทางซึ่งก็คือเครื่องลูกข่ายโทรศัพท์เคลื่อนที่กับเว็บเซิร์ฟเวอร์ คำสั่งและเนื้อหาที่รับส่งระหว่างอุปกรณ์ปลายทางกับเกตเวย์แว็พจะอยู่ในรูปแบบคำสั่งที่ถูกแปลงรูปหรืออาจเรียกได้ว่าเข้ารหัสไว้ โดยเกตเวย์แว็พจะเป็นตัวกลางในการแปลงข้อมูลดังกล่าวกลับไปเป็นข้อมูลโต้ตอบมาตรฐานดังรูปที่ 2-2



รูปที่ 2-2 แสดงโครงสร้างจริงของเครือข่ายสื่อสารแว็พ

สิ่งที่เห็นได้ชัดเจนก็คือเครื่องลูกข่ายโทรศัพท์เคลื่อนที่สามารถดึงจากเว็บไซต์ทั่วไปที่มีอยู่ในเครือข่ายอินเตอร์เน็ตได้ โดยไม่จำเป็นต้องมีการแก้สคริปต์หรือเนื้อความใดๆ บนเว็บไซต์เหล่านั้น ทั้งนี้เกตเวย์แว็พจะรับหน้าที่ในการแปลงรูปแบบข้อมูลให้เหมาะสมกับตัวเครื่องลูกข่ายโทรศัพท์เคลื่อนที่เอง อย่างไรก็ตามผู้ออกแบบเว็บไซต์หรือแม้กระทั่งผู้ให้บริการระบบเครือข่ายโทรศัพท์เคลื่อนที่ก็สามารถออกแบบสร้างเว็บไซต์ที่มีเนื้อหาเฉพาะสำหรับส่งไปให้กับเครื่องลูกข่ายได้โดยตรง ทั้งนี้เกิดจากการเขียนเว็บไซต์ด้วยภาษา WML และการเขียน WMLScript

2.1.2.2 ภาษาและรูปแบบที่ใช้ในการเขียนโปรแกรมบนมาตรฐานแว็พนั้นมีรากฐานมาจากมาตรฐานภาษา XML ทั้งนี้ภาษา WML (Wireless Markup Lanuage) ที่ใช้ในมาตรฐานแว็พนี้ได้รับการสร้างขึ้นเพื่อสนับสนุนกิจกรรมทางการสื่อสารซึ่งต้องอาศัยการประมวลผลที่สูงมากบนอุปกรณ์สื่อสารที่มีข้อ

จำกัดหลายประการที่สำคัญก็คือภาษา WML และ WMLScript ตามมาตรฐานแว็พจะไม่มีการยึดติดกับรูปแบบการป้อนข้อมูลเข้าจากคีย์บอร์ดตามมาตรฐาน QWERTY ซึ่งเป็นรูปแบบมาตรฐานทั่วไปตามเครื่องคอมพิวเตอร์แต่สามารถทำงานได้กับการป้อนเข้าข้อมูลในหลายรูปแบบ ซึ่งเป็นสิ่งจำเป็นอันเนื่องมาจากข้อกำหนดทางกายภาพของอุปกรณ์สื่อสารไร้สายเหล่านี้ยิ่งไปกว่านั้นภาษาดังกล่าวยังได้รับการออกแบบมาให้สามารถทำงานกับอุปกรณ์แสดงผลที่มีขนาดเล็ก

สิ่งที่ต่างออกไปจากรูปแบบเอกสาร HTML ที่ใช้กับเครือข่ายอินเตอร์เน็ตก็คือ เอกสารที่ถูกสร้างขึ้นตามมาตรฐาน WML ก็คือเอกสารแต่ละชุด ซึ่งก็เปรียบเสมือนเว็บไซต์แต่ละแห่งจะมีการจัดการข้อมูลภายในโดยแบ่งออกเป็นหน้าเช่นเดียวกับที่ปรากฏในเครือข่ายอินเตอร์เน็ต แต่ละหน้าจะถูกเรียกเป็นการ์ด(Card) โดยการ์ดแต่ละแผ่นจะมีคุณสมบัติพิเศษที่ช่วยให้ผู้ใช้งานสามารถสั่งย้อนถอยไปสู่ข้อมูลในหน้าที่ผ่านมาหรือเดินหน้าไปดูข้อมูลที่อยู่หน้าถัดไป การจัดสรรข้อมูลในลักษณะดังกล่าวทำให้โครงสร้างของแฟ้มข้อมูลเอกสาร WML มีขนาดเล็กกว่า HTML มากส่งผลให้มีความเหมาะสมที่จะถูกรับส่งผ่านเครือข่ายสื่อสารไร้สายอันมีข้อจำกัดในเรื่องของแบนด์วิดธ์ได้ดีในส่วนของอุปกรณ์เกตเวย์แว็พทำหน้าที่เป็นจุดเชื่อมต่อระหว่างเครือข่ายอินเตอร์เน็ตกับเครือข่ายไร้สายนั้น จะมีการส่งข้อมูล WML ที่ได้รับจากเครือข่ายไร้สายไปสู่เครือข่ายอินเตอร์เน็ต โดยข้อมูลเหล่านั้นจะถูกแปลงรูปแบบให้เป็นไปตามมาตรฐาน HTML 1.1 ซึ่งหมายความว่าบรรดาผู้พัฒนาเว็บไซต์ในเครือข่ายอินเตอร์เน็ตสามารถใช้โครงสร้างรวมถึงเครื่องมือในการพัฒนาที่มีอยู่เดิมของตนในการทำธุรกิจผ่านอุปกรณ์สื่อสารเคลื่อนที่ได้

2.1.2.3 ข้อกำหนดทางเทคนิคของโปรแกรมบราวเซอร์ขนาดเล็ก (Micro Browser) ที่ฝังอยู่ในเครื่องลูกข่ายอุปกรณ์สื่อสารไร้สาย อันได้แก่โทรศัพท์เคลื่อนที่แบบแว็พที่อำนวยความสะดวกให้กับผู้ใช้งานในการเข้าสู่โลกอินเตอร์เน็ตผ่านอุปกรณ์สื่อสารของตนนั้น มีรูปแบบการทำงานที่สอดคล้องกับเว็บบราวเซอร์ (WEB browser) ที่มีใช้งานอยู่ทั่วไปบนเครือข่ายอินเตอร์เน็ตหน้าที่การทำงานที่สำคัญของบราวเซอร์ขนาดเล็กก็คือการจัดการแปลงเนื้อหาที่ปรากฏอยู่ในเอกสาร WML หรือ WMLScript ให้ไปแสดงผลบนหน้าจออุปกรณ์สื่อสารไร้สายเพื่อให้ผู้ใช้งานรับทราบ เบราเซอร์เหล่านี้ได้รับการออกแบบมาเป็นพิเศษเพื่อให้สามารถทำงานได้กับอุปกรณ์สื่อสารไร้สายได้อย่างมีประสิทธิภาพ ทั้งในเรื่องของอัตราการประมวลผลที่ไม่สูงนักอันเนื่องมาจากเนื้อหาของเอกสาร WML ที่มีความกระชับ และความรวดเร็วในการรับส่งข้อมูลที่ปรากฏต่อสายตาผู้ใช้งานนอกจากนั้นความคล่องตัวในการทำงานก็คือกุญแจสำคัญสำหรับเบราเซอร์เหล่านี้ ตัวอย่างเช่น ผู้ออกแบบเอกสาร WML สามารถใส่รูปภาพที่มีขนาดไม่ใหญ่มากลงในเอกสาร WML เพื่อแสดงเครื่องหมายการค้าหรือรูปภาพบางอย่างเพื่อให้ไปแสดงผลบนหน้าจอของอุปกรณ์สื่อสารได้นอกเหนือจากตัวอักษรทั่วไป

2.1.2.4 โครงสร้างของแบบจำลองโพรโตคอลที่ได้รับการออกแบบมาด้วยจุดประสงค์ที่ต้องการทำให้ข้อมูลที่ถูกรับส่งผ่านเครือข่ายสื่อสารไร้สายมีขนาดกะทัดรัดที่สุดนั่นหมายถึงการที่ข้อมูลควบคุมซึ่งเป็นของโพรโตคอลในระดับต่างๆ ที่มีขนาดเล็กทำให้ไม่ไปแย่งแบนด์วิดธ์ในการรับส่งข้อมูลของตัวเนื้อหาหลัก นอกจากนั้นการออกแบบโครงสร้างโพรโตคอลตามมาตรฐานแว็พยังเปิดช่องทางให้กับนักพัฒนาสามารถสร้างโปรแกรมการประยุกต์ใช้งานหรือบริการพิเศษที่สำคัญโดยอาศัยบริการที่มีบนมาตรฐาน

แว็พได้อย่างเต็มที่ และที่สำคัญหากพิจารณาในส่วนล่างของรูปที่ 2-3 จะเห็นว่ามาตราฐานแว็พสามารถทำงานได้กับมาตรฐานการสื่อสารไร้สายหลากหลายประเภท

2.1.2.5 การพัฒนาความสามารถในด้าน WTA (Wireless Telephony Application) ซึ่งจะช่วยทำให้ผู้พัฒนาโปรแกรมสามารถควบคุมการทำงานของโทรศัพท์เคลื่อนที่ได้จากเว็บไซต์ โดยการทำงานหลายๆ ประการของโทรศัพท์เคลื่อนที่จะถูกกำหนดให้อยู่ในรูปของชุดคำสั่งของการเขียนโปรแกรม ฟังก์ชั่นโดยทั่วไปที่สามารถควบคุมได้ก็คือ การโทรออกการจัดการสมุดบันทึกเลขหมายโทรศัพท์และข้อมูลในส่วนของ Short Message ซึ่งผู้พัฒนาเว็บไซต์ที่ให้บริการจัดสรรข้อมูลส่วนบุคคล (Organizer) โดยมีมาตรการในการรักษาความปลอดภัยของข้อมูลอย่างดี ผู้ใช้งานแต่ละรายสามารถใช้เครื่องคอมพิวเตอร์ตามสำนักงานเข้าไปจัดการปรับปรุงข้อมูลเว็บไซต์ แล้วจึงสั่งให้มีการดาวน์โหลดข้อมูล ใหม่ลงสู่เครื่องลูกข่ายโทรศัพท์เคลื่อนที่ของตนผ่านทางแว็พอีกตัวอย่างก็คือการควบคุมการรับสายเข้า โดยเมื่อมีสายเรียกเข้าสู่ผู้ใช้บริการรายหนึ่งก็จะมีเอกสาร WML ส่งไปเพื่อแจ้งให้ผู้ใช้บริการเลือกว่าจะรับสายหรือโอนสายไปยังเลขหมายต่างๆ ตามต้องการ ทั้งนี้สามารถกำหนดได้อีกด้วยว่าจะให้มีการจัดการเลขหมายที่เรียกเข้าได้อย่างไร แนวทางในการดำเนินธุรกิจจริงคงจะมีมากกว่านี้ ขึ้นอยู่กับความคิดของผู้ประกอบการแต่ละราย

2.1.3 การทำงานและโครงสร้างของแว็พ

2.1.3.1 สถาปัตยกรรมของแว็พ (WAP Architecture)

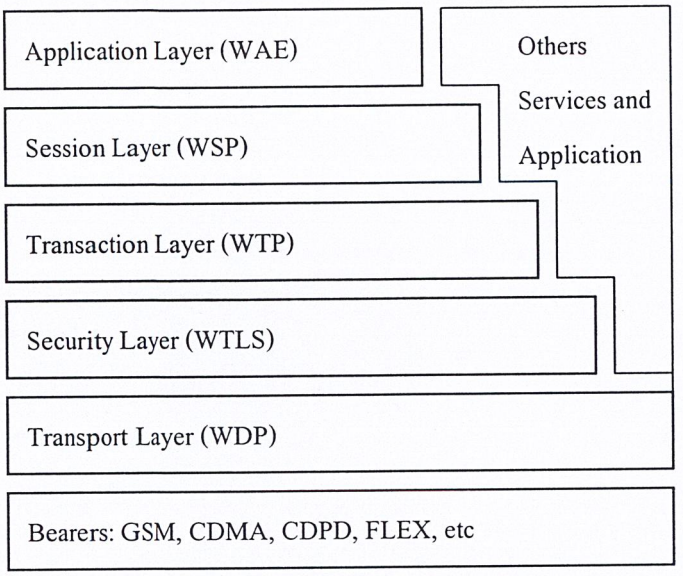
แว็พมีสถาปัตยกรรมแบ่งเป็นเลเยอร์ที่เหมือนกับโมเดลของระบบเน็ตเวิร์คที่เรียกว่า OSI โมเดล ดังรูปที่ 2-3 ซึ่งในแต่ละเลเยอร์ประกอบด้วยดังนี้

2.1.3.1.1 เลเยอร์ Wireless Application Environment (WAE) เป็นส่วนที่มีการทำงานเหมือนกัน HTML และ Java Script ก็คือการกำหนดภาษา WML และ WMLScript ที่ทำหน้าที่แสดงข้อมูล และข่าวสารที่จำเป็นให้กับผู้ใช้ผ่านทางแว็พบราวเซอร์

2.1.3.1.2 เลเยอร์ Wireless Session Protocol (WSP) และเลเยอร์ Wireless Transaction Protocol (WTP) เป็นส่วนประกอบที่มีการทำงานคล้ายกับโพรโตคอล HTTP ก็คือส่วนที่ทำหน้าที่เชื่อมต่อการสื่อสารระหว่างเครื่องโทรศัพท์มือถือ และเซิร์ฟเวอร์หรือเกตเวย์ในส่วนนี้จะเกี่ยวข้องกับเทคโนโลยีการสื่อสารแบบไร้สายโดยตรง เช่นความเร็วของข้อมูลที่สามารถส่งได้ หรือเวลาที่ใช้ในการต่อเข้ากับเซิร์ฟเวอร์

2.1.3.1.3 เลเยอร์ Wireless Transport Layer Security (WTLS) เป็นเลเยอร์ที่สร้างมาจากโพรโตคอลมาตรฐาน Transport Layer Security (TLS) ซึ่งรู้จักกันดีในชื่อของ Secure Sockets Layer (SSL) ซึ่งจะทำหน้าที่ในการรักษาความปลอดภัยของข้อมูล และป้องกันภัยจากผู้บุกรุกชนิดต่างๆ (attacker) โดยมีบริการของ การตรวจสอบความถูกต้องของข้อมูล (Data Integrity), การป้องกันข้อมูลที่เป็นความลับ (Confidential), การพิสูจน์ตน(Authentication) และการป้องกันการมาใช้บริการโดยไม่ได้รับอนุญาต(Denial of Service) มาเป็นเครื่องมือ

2.1.3.1.4 เลเยอร์ Datagram หรือ Transport Layer (WDP) มีการทำงานเหมือนโพรโตคอล TCP หรือ UDP ที่ทำหน้าที่ในการส่งข้อมูลระหว่างเลเยอร์ชั้นบนกับเครื่องเซิร์ฟเวอร์หรือเครื่องโทรศัพท์มือถือ (client)ผ่านระบบเครือข่ายไร้สายชนิดต่างๆ



รูปที่ 2-3 สถาปัตยกรรมของแว็พ

2.1.3.2 หลักการทำงานของแว็พ

แว็พมีการทำงานในลักษณะเดียวกันกับเว็บคือ เป็นแบบไคลเอนต์และเซิร์ฟเวอร์ (Client-Server) แต่แว็พนั้นจะมีเครื่องมือพิเศษเข้ามาทำงานเป็นตัวกลางเชื่อมต่อระหว่างโพรโตคอลแว็พ และ HTTP ซึ่งก็คือ แว็พเกตเวย์

การทำงานของแว็พบนโทรศัพท์มือถือสามารถอธิบายเป็นขั้นตอนย่อยๆ ได้ดังนี้

2.1.3.2.1 เมื่อใช้ต้องการเปิดแว็พไซต์ที่อยู่ในอินเตอร์เน็ตขึ้นมาชมในโทรศัพท์มือถือผู้ใช้จำเป็นต้องใส่ค่า URL(Uniform Resource Locator) ที่ต้องการลงไป เช่น http://www.ce.kmitl.ac.th

2.1.3.2.2 เครื่องโทรศัพท์มือถือจะทำหน้าที่แปลงข้อมูลของ URL พร้อมทั้งบีบอัดไฟล์ให้เป็นเลขฐานสอง หรือไบนารี่ไฟล์เช่น http://www.ce.kmitl.ac.th 0011010111ซึ่งจะช่วยในการประหยัดแบนด์วิธ

2.1.3.2.3 เครื่องโทรศัพท์มือถือส่งไบนารี่ไฟล์นั้นไปยังแว็พเกตเวย์ และเกตเวย์เปลี่ยนคำร้องขอข้อมูล(Request) ของ WSP เป็น HTTP

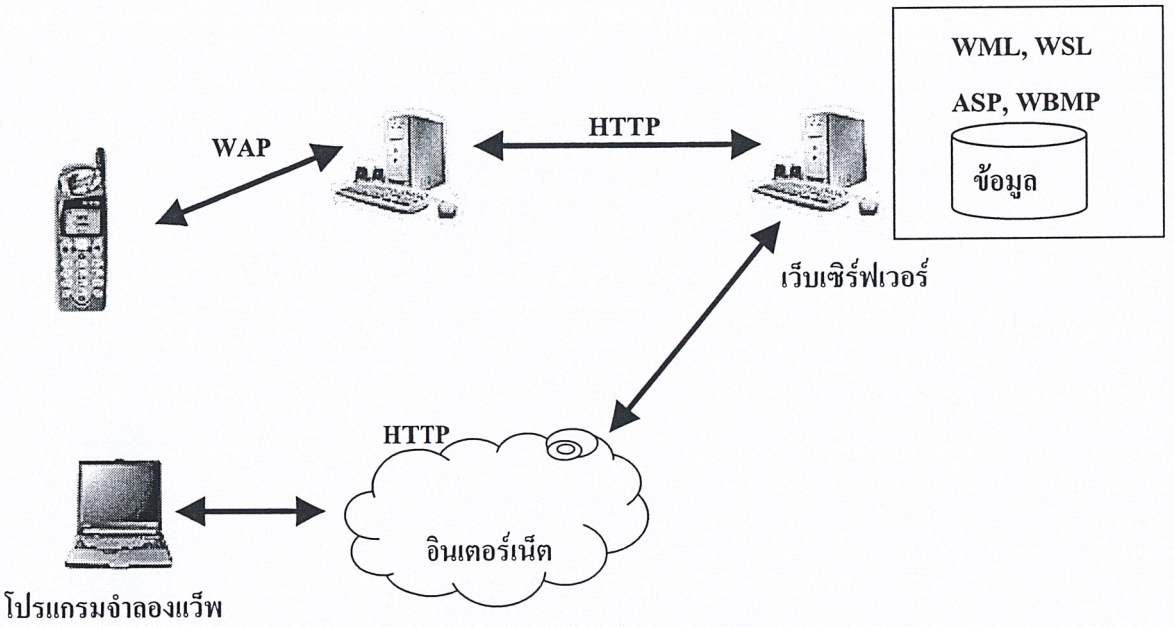
2.1.3.2.4 จากนั้นเกตเวย์จะคงจะส่ง HTTP Request ไปในเครือข่ายอินเตอร์เน็ต

2.1.3.2.5 ทางด้านฝั่งผู้ให้บริการแว็พก็จะมีเซิร์ฟเวอร์ที่มีแว็พเขียนด้วย WML เมื่อเซิร์ฟเวอร์ของ www.ce.kmitl.ac.th ได้รับคำร้องขอ หรือ HTTP Request จากเกตเวย์ ก็ทำการส่งไฟล์ WML กลับไป

2.1.3.2.6 เมื่อเกตเวย์ได้รับไฟล์ WML กลับมาจากเซิร์ฟเวอร์ ก็จะทำการเปลี่ยนแปลงข้อมูลเป็นไบนารี่ไฟล์ เพื่อเตรียมส่งข้อมูลกลับไปหาเครื่องโทรศัพท์มือถือนั้นโดยใช้โพรโตคอลแว็พ

2.1.3.2.7 เมื่อเครื่องโทรศัพท์มือถือรับไบนารี่ไฟล์นั้นแล้วก็ทำการแปลงข้อมูลเป็นภาษา WML ที่แว็พ บราวเซอร์ในเครื่องสามารถอ่านได้

แท้จริงแล้ว นอกจากจะสามารถเข้าชม แว็พไซต์โดยใช้โทรศัพท์มือถือแล้วยังสามารถใช้โปรแกรมจำลอง หรือที่เรียกกันว่า Emulator มาท่องโลกของแว็พก็ได้เพราะมีโปรแกรมแว็พบราวเซอร์อยู่ในโปรแกรมแล้ว หรือแม้แต่จะเข้าไปชมจากในเว็บไซต์ที่มีโปรแกรมเหล่านี้อยู่ก็ได้ ดังแสดงการใช้งานโปรแกรมแว็พบราวเซอร์และเครื่องโทรศัพท์มือถือในการชมแว็พไซต์ดังรูปที่ 2-4



รูปที่ 2-4 แสดงการใช้งานโปรแกรมแว็พบราวเซอร์และเครื่องโทรศัพท์มือถือในการชมแว็พไซต์

2.1.3.3 โครงสร้างของแว็พเกตเวย์

โครงสร้างของเกตเวย์แว็พในรูปที่ 2-5 ประกอบด้วยส่วนต่างๆ ดัง.นี้

2.1.3.3.1 WDP (WAP Datagram Protocol) เป็นระดับชั้นขนส่งข้อมูลซึ่งทำหน้าที่ส่งและรับข้อมูลข่าวสาร จากเครือข่ายต่างๆ ที่เกตเวย์ทำการเชื่อมต่ออยู่ด้วยไม่ว่าข้อมูลนั้นจะอยู่ในรูปของ SMS,USSD,CSD,CDPD,IS-136 และ GPRS

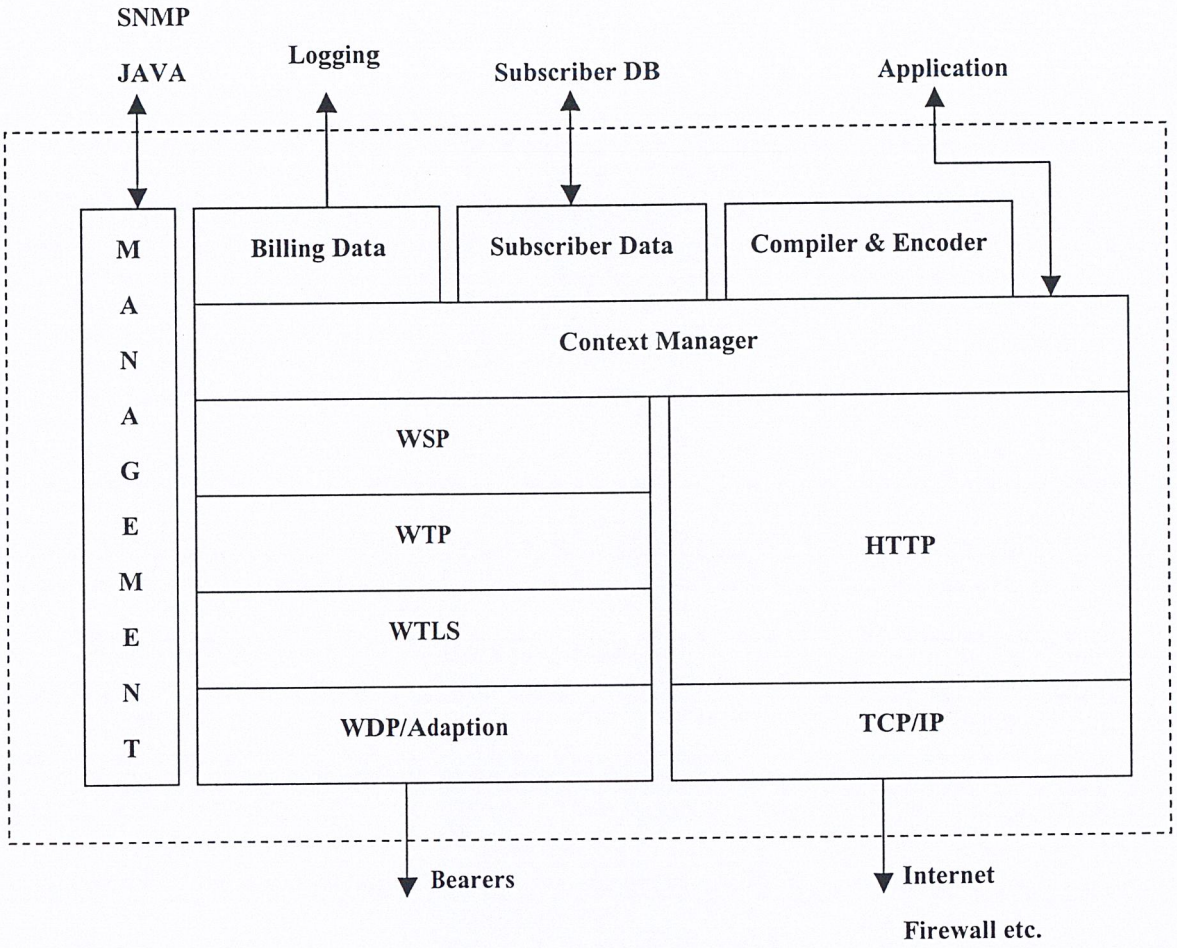
2.1.3.3.2 WTLS (Wireless Transport Layer Security) เป็นระดับชั้นที่ทำหน้าที่รักษาความปลอดภัยของข้อมูล ซึ่งผู้ให้บริการจะกำหนดให้มีหรือไม่ก็ได้ระดับชั้นดังกล่าวมีฟังก์ชันที่ทำหน้าที่ในการเข้ารหัสข้อมูลซึ่งเหมาะสำหรับการใช้งานกับบริการบางประเภทเช่น E-commerce

2.1.3.3.3 WTP (WAP Transaction Protocol) เป็นระดับชั้นที่ทำหน้าที่สนับสนุนการสร้างกระบวนการการส่งผ่านข้อมูล พร้อมทั้งช่วยเพิ่มระดับความน่าเชื่อถือให้กับการเชื่อมต่อแบบ Datagram ที่กระทำโดยระดับชั้น WDP

2.1.3.3.4 WSP (WAP Session Protocol) เป็นระดับชั้นที่ให้การสนับสนุนเพิ่มประสิทธิภาพในการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ต่างๆ

2.1.3.3.5 HTTP Interface ช่วยให้เครื่องลูกข่ายโทรศัพท์เคลื่อนที่สามารถดึงข้อมูลในรูปแบบของแว็พจากเครือข่ายอินเตอร์เน็ต

ข้อมูลในรูปแบบของแว็พไม่ว่าจะอยู่ในรูปของ WML หรือ WMLScript จะถูกแปลงให้อยู่ในรูปข้อมูลไบนารีชนิดพิเศษสำหรับส่งผ่านการเชื่อมต่อทางอากาศจากเครือข่ายโทรศัพท์เคลื่อนที่ไปยังเครื่องลูกข่าย จากนั้นไมโครเบราเซอร์ในเครื่องลูกข่ายจะทำการอ่านข้อมูลดังกล่าวแล้วแสดงผล



รูปที่ 2-5 โครงสร้างของแว็พเกตเวย์

2.1.4 การกำหนด MIME Type เพื่อให้เซิร์ฟเวอร์สามารถรองรับแว็พได้

เมื่อได้สร้างแว็พเพจเสร็จแล้ว ต้องทำการให้ข้อมูลที่สร้างขึ้นสามารถทำให้โทรศัพท์มือถือและแว็พบราวเซอร์สามารถอ่านข้อมูลเหล่านี้ได้โดยจำเป็นต้องทำการกำหนด MIME Type ของเซิร์ฟเวอร์ให้สามารถสนับสนุนการทำงานของแว็พได้โดยต้องทำการกำหนด MIME Type เป็นดังตารางที่ 2-1 เพื่อให้เซิร์ฟเวอร์นั้นสามารถรองรับการบริการจากโทรศัพท์มือถือและแว็พบราวเซอร์ได้

โดยที่ข้อมูลที่ถูกกำหนด MIME Type จะถูกแบ่งออกเป็น 3 ส่วนคือ

2.1.4.1 ประเภทเอกสาร จะเป็นข้อมูลในรูปแบบอักษร ได้แก่ เอกสาร WML ซึ่งจะมีนามสกุลเป็น *.wmlและ เอกสาร WML สคริปต์ จะมีนามสกุลเป็น *.wmls แต่อาจพบว่ามีการใช้นามสกุลเป็น *.ws หรือ *.wmlscript ด้วย

2.1.4.2 ประเภทรูปภาพซึ่งเป็นไฟล์รูปภาพที่ภาษา WML รองรับ ได้แก่ ไฟล์รูปภาพแบบ WBMP ซึ่งมีนามสกุลเป็น *.wbmp

2.1.4.3 ประเภทแอปพลิเคชั่น ใช้กับข้อมูลที่ถูกแปลงจากเอกสาร WML หรือ WMLScript ให้เป็นรหัส ไบนารีหรือไบต์โค้ดแล้วไฟล์เหล่านี้จะมีนามสกุลเป็น *.wmlc และ *.wmlsc

ชนิดของเอกสาร	MIME Type	นามสกุลของไฟล์
เอกสาร WML	text/vnd.wap.wml	.wml
ไฟล์รูปภาพ WBMP	image/vnd.wap.wbmp	.wbmp
เอกสาร WML ที่ได้คอมไพล์แล้ว	Application/vnd.wap.wmlc	.wmlc
WML สคริปต์	text/vnd.wap.wmlscript	.wmls
WML สคริปต์ที่ได้คอมไพล์แล้ว	Application/vnd.wap.wmlscriptc	.wmlsc

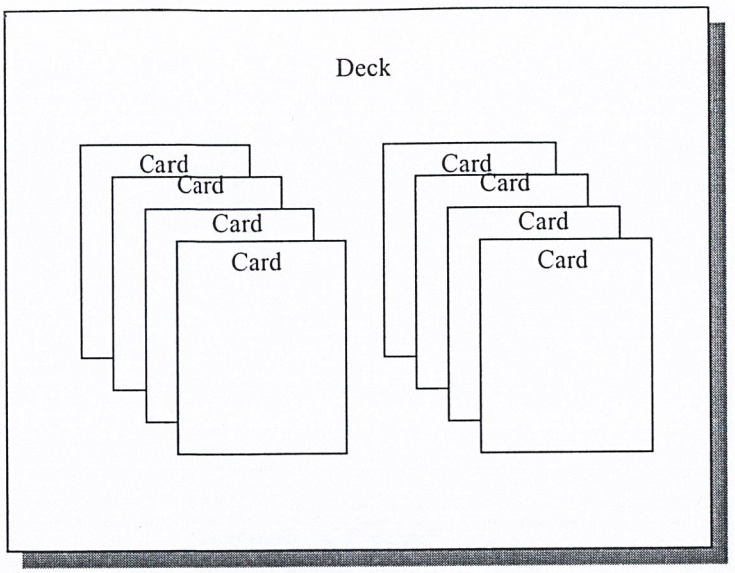
ตารางที่ 2-1 การกำหนด MIME Typeให้กับเซิร์ฟเวอร์

2.1.5 การเขียนโปรแกรมภาษา WML

2.1.5.1 โครงสร้างของเอกสาร WML

โครงสร้างของเอกสาร WML นั้นถูกออกแบบมาให้มีการทำงานเหมือนกับไพ่ โดยที่ไพ่แต่ละใบหรือการ์ด (Card) เปรียบได้กับแว็พเพจหนึ่งหน้า และเมื่อรวมการ์ดหรือแว็พเพจหลายๆ หน้าเข้าด้วยกันเป็นสำรับ หรือที่เรียกว่าเด็ค (Deck) ก็จะได้เป็นแว็พไซต์ ซึ่งการที่ WML ถูกออกแบบให้มีลักษณะเหมือนกับไพ่หลายๆ ใบในสำรับหนึ่ง ก็เพราะว่าอุปกรณ์แว็พจะได้ไม่เสียเวลาในการส่งคำร้องขอไปยังเซิร์ฟเวอร์บ่อยๆ ซึ่งเป็นการช่วยลดเวลาและแบนด์วิธลงได้

ดังนั้น โครงสร้างที่ใหญ่ที่สุดของเอกสาร WML ก็คือเด็คซึ่งก็คือเนื้อหาทั้งหมดที่อยู่ในเอกสาร WML ในแต่ละเด็คจะประกอบไปด้วยกลุ่มเอกสารย่อยๆ ซึ่งก็คือการ์ด โดยในแต่ละการ์ดจะเป็นที่เก็บข้อมูลต่างๆ ที่ต้องการให้แสดงที่หน้าจอรวมไปถึงคำสั่งต่างๆ ที่ต้องการให้เอกสาร WML ตอบสนองต่อความต้องการของผู้ใช้ดังแสดงในรูปที่ 2-6

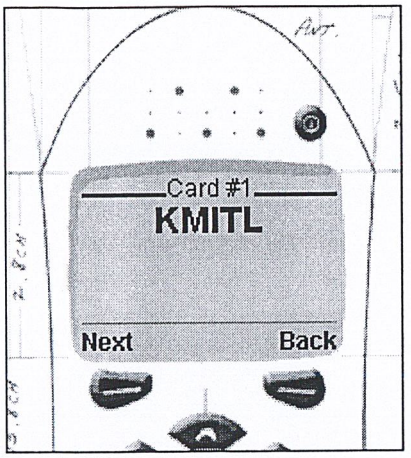


รูปที่ 2-6 เด็ค และการ์ดของภาษา WML

โดยเอกสาร WMLนั้นอย่างน้อยต้องประกอบไปด้วยส่วนต่างๆ ดังตัวอย่างที่ได้แสดงให้เห็นดังนี้

บรรทัดที่	เอกสาร WML
1	<?xml version=”1.0”>
2	<!DOCTYPE wml PUBLIC “-//WAPFORUM//DTD WML 1.1//EN”
3	“http://www.wapforum.org/DTD/wml_1.1.xml”>
4	<wml>
5	<card>
6	<!—สามารถเขียนข้อมูลลงไปได้-- >
7	KMITL
8	</card>
9	</wml>

ซึ่งผลลัพธ์ที่ปรากฏอยู่บนโทรศัพท์มือถือจะเป็นดังรูปที่ 2-7



รูปที่ 2-7 แสดงผลลัพธ์ของไฟล์เอกสาร WML

โดยมีความหมายในแต่ละบรรทัดดังนี้

บรรทัดที่	ความหมาย
1-3	เป็นส่วนที่เรียกว่า เฮดเดอร์ (Header) ซึ่งเป็นส่วนที่บอกบราวเซอร์ว่าเอกสารนี้เป็นเอกสาร WML เวอร์ชั่น 1.0 ในเอกสาร WML ทุกเอกสารจะต้องมีเฮดเดอร์นี้ มิฉะนั้นจะทำให้การแสดงข้อมูลไม่ถูกต้อง
4 และ 9	เป็นจุดเริ่มต้นและจุดสิ้นสุดของเด็ค (<wml>…</wml>) จะเห็นว่าลักษณะของแท็กจะคล้ายกับ HTML
5 และ 8	เป็นจุดเริ่มต้นและจุดสิ้นสุดของการ์ด (<card>..</card>)
6	เป็นส่วนหมายเหตุ <- - …- -> ข้อความที่อยู่ในแท็กหมายเหตุจะไม่ถูกแสดงผลในบราวเซอร์
7	เป็นส่วนข้อมูลที่ออกสู่บราวเซอร์ (..) ข้อความต่างๆ ที่อยู่ภายในแท็กนี้จะถูกนำมาแสดงผลที่บราวเซอร์

2.1.5.2 แท็กต่างๆ ในภาษา WML

แท็กต่างๆ ในภาษา WML นั้นประกอบด้วยแท็กต่างๆ มากมายดังตารางต่อไปนี้

ชื่อแท็ก	คำอธิบายแท็ก
a	ใช้เชื่อมโยงระหว่างการ์ดหรือหน้าแว็พเพจ ซึ่งจะต่างกับ do ตรงที่จะไม่มีการแสดงในเมนูให้เห็น
access	ใช้ในการควบคุมการเข้าถึงข้อมูลของผู้เข้าชมหรือให้เข้าชมเฉพาะบางกลุ่มก็ได้
anchor	ใช้เชื่อมโยงระหว่างการ์ดหรือหน้าแว็พเพจ ซึ่งจะต่างกับ do ตรงที่จะไม่มีการแสดงในเมนูให้เห็น
b	ทำให้ตัวอักษรหนาขึ้น
big	ทำให้ตัวอักษรใหญ่ขึ้น
br	ใช้สำหรับกำหนดให้ขึ้นบรรทัดใหม่สำหรับข้อความที่อยู่ต่อจากแท็กนี้
card	กำหนดจุดเริ่มต้นและจุดสิ้นสุดของการ์ด ในเด็คเด็คหนึ่งสามารถมีการ์ดได้หลายๆ การ์ด
do	ใช้เชื่อมโยงจากการ์ดที่แสดงอยู่ไปยังการ์ดอื่นหรือ URI อื่นๆ ได้โดยการรับจากเชื่อมโยงโดยใช้เมนู และปุ่มต่างๆ ที่แสดงบนหน้าจอ
em	ทำให้ตัวอักษรเด่นขึ้นซึ่งจะเห็นเป็นตัวเอียง
fieldset	ช่วยในการทำให้แว็พบราวเซอร์แสดงส่วนรับข้อมูลต่างๆ ได้ดีขึ้น
go	ใช้ในการกำหนดให้แว็พบราวเซอร์เปลี่ยนหน้าของแว็พเพจไปตามตำแหน่งที่กำหนดในแอตทริบิวต์
head	เป็นส่วนที่ใช้ในการเก็บข้อมูลและรายละเอียดที่มีความสัมพันธ์กับเอกสาร WML
i	ทำให้ตัวอักษรเป็นตัวเอียง
img	ใช้ในการใส่รูปลงไปในแว็พเพจ
input	ใช้รับข้อมูลที่เป็นตัวอักษร
meta	ใช้ในการกำหนดคำอธิบายเพื่อให้โปรแกรมสืบค้น (Search Engine) ได้ค้นหาแว็พเพจได้
noop	เป็นแท็กที่ไม่ทำอะไรเลย และใช้ในการยกเลิกการทำงานของ template ได้
onevent	ใช้ในการกำหนดเหตุการณ์ต่างๆ เช่น ontimer เป็นต้น
optgroup	ทำหน้าที่เป็นตัวจัดกลุ่มให้กับตัวเลือกซึ่งทำงานร่วมกับ select และ option
option	ใช้สำหรับทำหน้าที่เป็นตัวแสดงตัวเลือกในลักษณะเมนู โดยทำงานร่วมกับ select
p	ใช้สำหรับแสดงข้อมูลออกสู่จอภาพ โดยมีการกำหนดการจัดวางตัวอักษรเช่นชิดซ้าย ชิดขวา และตรงกลางและการขึ้นบรรทัดใหม่เองถ้าข้อความยาวกว่าหน้าจอ
postfield	ใช้ในการส่งค่าต่างๆ ที่รับจากส่วนรับข้อมูลส่งไปยัง CGI
prev	ใช้สำหรับสร้างลิงค์กลับไปที่หน้าเก่า เพื่อให้สามารถย้อนกลับไปยังหน้าที่ผ่านมาแล้วได้

ชื่อแท็ก	คำอธิบายแท็ก
refresh	ใช้ในการโหลดหน้าแว็พเพจขึ้นมาแสดงใหม่หรือเพื่อโหลดข้อมูลรวมทั้งค่าของตัวแปรล่าสุดออกมาแสดงอีกครั้งหนึ่ง
select	ใช้สำหรับทำหน้าที่เป็นตัวแสดงตัวเลือกในลักษณะเมนู โดยทำงานร่วมกับ option
setvar	ใช้สำหรับกำหนดค่าของตัวแปร
small	ทำให้ตัวอักษรเล็กลง
strong	ทำให้ตัวอักษรเด่นขึ้นซึ่งจะเป็นตัวหนาขึ้น
table	ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของตาราง
td	ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของคอลัมน์หนึ่งในตาราง
template	ทำหน้าที่เกี่ยวกับการกำหนดคำสั่งงานหรือเหตุการณ์ต่างๆ ซึ่งคำสั่งของเหตุการณ์นี้สามารถนำไปใช้ได้ทุกการ์ดในเด็คนั้น
timer	ใช้กำหนดเวลาในการทำงานอย่างใดอย่างหนึ่งเช่นกำหนดเวลาในการเชื่อมโยงไปยังอีกการ์ดหนึ่งเป็นต้น
tr	ใช้กำหนดจุดเริ่มต้นและจุดสิ้นสุดของแถวหนึ่งแถวในตาราง
u	ทำให้ตัวอักษรถูกขีดเส้นใต้
wml	เป็นจุดเริ่มต้นและจุดสิ้นสุดของเอกสาร WML

ตารางที่ 2-2 แสดงแท็กต่างๆ ของภาษา WML

2.2 ความรู้เบื้องต้นเกี่ยวกับการเขียนโปรแกรมด้วยภาษาจาวา

2.2.1 คอมไพเลอร์ภาษาจาวา เวอร์ชัน 1.2.2 ที่สนับสนุนภาษาไทย

ภาษาจาวาเวอร์ชัน 1.2.2 เป็นภาษาจาวาที่ได้รับการพัฒนามาจากโค้ดของจาวา 2SE เวอร์ชัน 1.2.2 ภาษาไทย และเข้ากันได้กับจาวารุ่นก่อนหน้านี้ทุกเวอร์ชัน ซอฟต์แวร์ทุกตัวที่เคยใช้บนจาวารุ่นก่อนๆ สามารถใช้บนจาวาภาษาไทยรุ่นนี้ได้โดยไม่ต้องมีการแก้ไขโค้ดหรือคอมไพล์ใหม่แต่อย่างใดและซอฟต์แวร์ที่พัฒนาบนจาวา 2 ไทย ก็สามารถนำไปรันบนจาวารุ่นอื่นๆ ได้เหมือนกัน (แต่อาจแสดงผลภาษาไทยไม่ถูกต้องและไม่สนับสนุน locale th_TH) โดยจาวาเวอร์ชันไทยนี้จะสนับสนุนรหัสอักขระภาษาไทยตามมาตรฐานอุตสาหกรรม ซึ่งใช้รหัสอักขระสมอ. 620-2533 (TIS620) เป็นรหัสอักขระมาตรฐาน นอกจากนี้ยังสนับสนุนรหัสอักขระภาษาไทย MS874 (Windows-874) ซึ่งใช้ในไมโครซอฟต์วินโดวส์, CP874 ซึ่งใช้บนระบบของไอบีเอ็ม, CP838 (Thai EBCDIC) ซึ่งใช้กันบนเครื่องเมนเฟรม และ MAC Thai ซึ่งใช้กันบนเครื่องแอปเปิลแมคอินทอช แต่ทั้งนี้ จาวา 2 ภาษาไทย จะใช้ TIS620 ซึ่งเป็นมาตรฐานอุตสาหกรรมเป็นหลัก

โดยคอมโพเนนต์ในภาษาจาวาเช่น JFC/Swing ทุกตัวถูกพัฒนาใหม่เพื่อให้แสดงผลและรับข้อมูลภาษาไทยได้อย่างถูกต้อง ตัวอักษรภาษาไทยทุกระดับแสดงผลเป็น 4 ระดับอย่างถูกต้องและสวยงาม ในส่วนของการรับข้อมูลจากคีย์บอร์ดจะมีการตรวจสอบลำดับตัวอักษรเพื่อให้แน่ใจว่าข้อมูลที่ป้อนเข้ามา

มีความถูกต้องโดยการตรวจสอบลำดับอักษรที่ใช้ในจาวารุ่นนี้เป็นส่วนขยายของ WTT ระดับ 2 (de facto standard) และยังมีการตรวจสอบลำดับอินพุตตัวอักษรภาษาไทยตามข้อกำหนด วทท. ระดับ 2 และฟอนต์ภาษาไทยในตัว 3 แบบ (True Type Font) โดยที่สามารถเรียกใช้ฟอนต์ชุดนี้จากที่ใดหรือระบบปฏิบัติการใดก็ได้ และจาวาภาษาไทยยังมีการเพิ่มในส่วนของการใช้ปีพุทธศักราชในจาวาด้วยโดยมีแพ็กเกจ java.util.Calendar เพื่อสนับสนุน และสนับสนุนการพิมพ์วัน เดือน ปี โดยแสดงชื่อวันและเดือนเป็นภาษาไทยได้ทั้งแบบเต็มและแบบย่อโดยมีแพ็กเกจ java.text.DateFormat ในการสนับสนุน

2.2.2 API ต่างๆ ในภาษาจาวาที่รองรับการเขียนโปรแกรมบนเครือข่าย

การติดต่อสื่อสารระหว่างคอมพิวเตอร์ในระบบเครือข่ายอินเตอร์เน็ตโดยใช้โพรโตคอล HTTP นั้นประกอบได้ด้วยส่วนที่เป็นเซิร์ฟเวอร์ และส่วนที่เป็นไคลเอนต์ โดยส่วนที่เป็นเซิร์ฟเวอร์จะทำหน้าที่ในการส่งข้อมูลให้กับไคลเอนต์ที่ทำการร้องขอข้อมูล ส่วนไคลเอนต์ก็ทำหน้าที่ในการร้องขอข้อมูลและนำมาแสดงผลตาม URL ที่ผู้ใช้ร้องขอซึ่งโพรโตคอลแว็พก็ถูกอ้างอิงมาเพื่อให้ติดต่อกับโพรโตคอล HTTP เช่นกันด้วยเหตุนี้แว็พบราวเซอร์ก็ต้องมีการติดต่อกับเซิร์ฟเวอร์ในลักษณะเดียวกัน โดยภาษาจาวาได้มี API ต่างๆ รองรับในการติดต่อกับเซิร์ฟเวอร์ในระบบเครือข่ายเช่นกัน

ในส่วนของภาษาจาวาที่ใช้ใน การเขียนโปรแกรมบนเครือข่ายนั้นประกอบด้วยแพ็กเกจใน java.net.* ซึ่งมี API ต่างๆ ที่สนับสนุนการเขียนโปรแกรมหลายตัวเช่น คลาส URL โดยมีคอนสตรัคเตอร์ (Constructor) ที่รับ url เช่น http://www.ce.kmitl.ac.th เป็นต้น ซึ่งรายละเอียดต่างๆ ในการใช้งานคลาสเหล่านี้สามารถอ่านได้จากเอกสาร API ในภาษาจาวาหรือเว็บไซต์ของบริษัทซันไมโครซิสเต็ม (SUN Microsystems)

2.3 ความรู้เบื้องต้นเกี่ยวกับการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML

การพัฒนาระบบเชิงวัตถุนั้นประกอบไปด้วยขั้นตอนสำคัญ 2 ขั้นตอนคือ ขั้นตอนของการวิเคราะห์และการออกแบบระบบเชิงวัตถุ ในการวิเคราะห์และออกแบบระบบเชิงวัตถุนี้ผู้พัฒนาจะมองภาพของระบบเป็นยูสเคสซึ่งก็คือการมองฟังก์ชันการทำงานหลักของระบบโดยขั้นตอนของการออกแบบและพัฒนาระบบผู้พัฒนาจะเขียนออกมาเป็นแผนภาพยูสเคส,แผนภาพแสดงลำดับขั้นตอนการทำงาน (Sequence Diagram), แผนภาพแสดงการทำงานร่วมกันของวัตถุ (Collaboration Diagram), แผนภาพวัตถุ และ แผนภาพสถานะ(State Diagram) ซึ่งแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

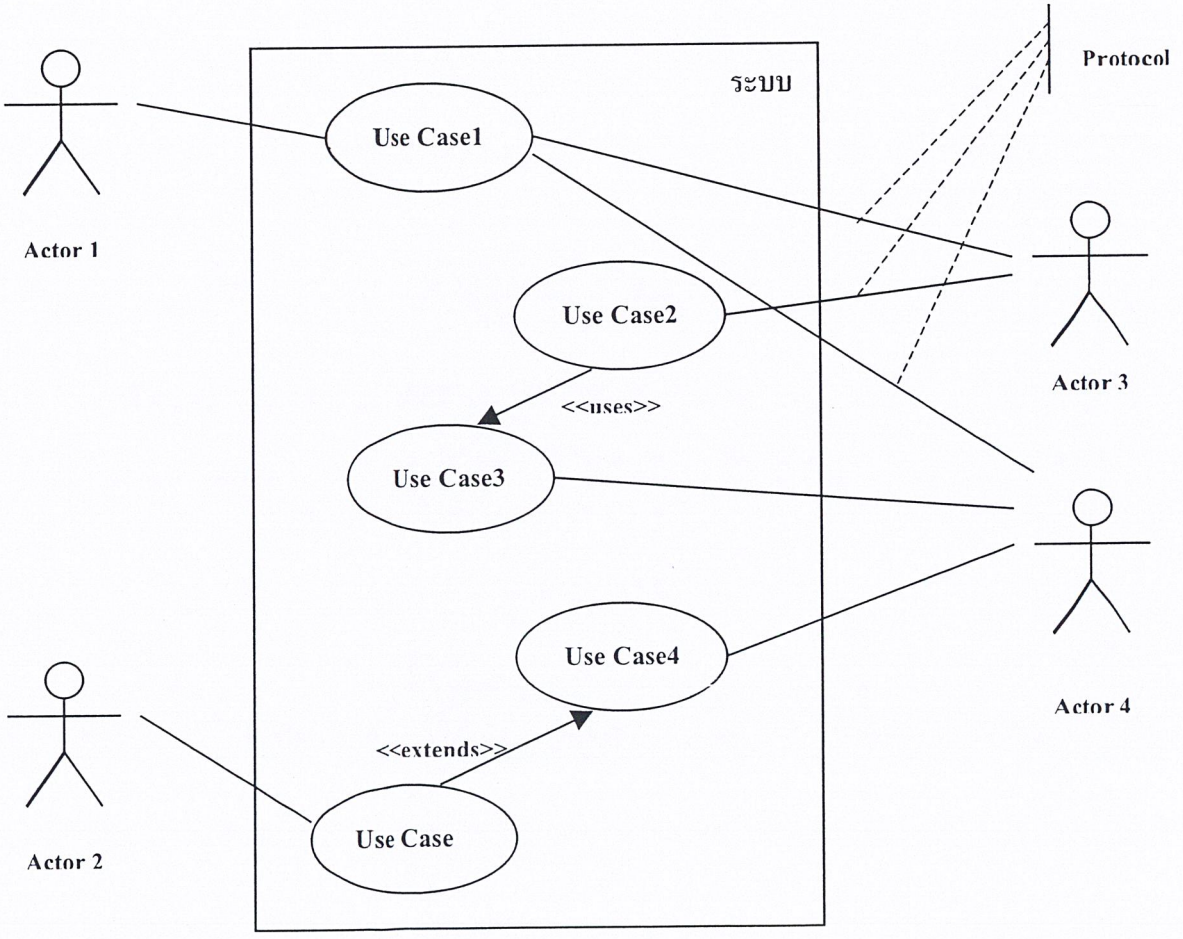
2.3.1 แผนภาพยูสเคส (Use Case)

แผนภาพคอนเท็กซ์ ที่ได้จากการวิเคราะห์นั้นจะบอกถึงการติดต่อระหว่างสิ่งแวดล้อมภายนอกกับระบบ และสำหรับในการวิเคราะห์ระบบด้วยวิธีเชิงวัตถุนั้นผู้พัฒนาสามารถสร้างแผนภาพยูสเคสซึ่งเป็นแผนภาพที่เพิ่มกรณีสำหรับการติดต่อระหว่างระบบกับแอคเตอร์ภายนอก แผนภาพยูสเคสนี้ยังแสดงการไหลของอีเวนต์ด้วย และที่สำคัญแผนภาพยูสเคสสามารถใช้ในการตรวจสอบความถูกต้องของแผนภาพคอนเท็กซ์ได้ด้วย

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แผนภาพยูสเคสนั้นจะแสดงมุมมองต่อฟังก์ชันการทำงานหลักๆ ของระบบซึ่งผู้พัฒนาระบบสามารถใช้แผนภาพยูสเคสนี้ในการสื่อสารกับผู้ใช้ระบบได้ด้วย ตัวแผนภาพยูสเคสอาจจะเป็นแผนที่หลักในการพัฒนาระบบเนื่องจากตัวแผนภาพยูสเคสนั้นได้รวมความต้องการ และรายละเอียดการทำงาน หรือลำดับขั้นตอนการทำงานของระบบไว้ภายในแล้ว สำหรับตัวยูสเคสเองนั้นก็สามารถที่จะแบ่งเป็นยูสเคสย่อยๆ ลงไปได้อีก หรืออาจจะใช้ในการสร้างลำดับขั้นตอนการทำงานซึ่งจะแสดงรายละเอียดลำดับในการติดต่อสื่อสารระหว่างวัตถุในระบบ เนื่องจากแผนภาพยูสเคสนั้นเป็นมุมมองต่อระบบในระดับสูง และมีการใช้ภาษาอธิบายระบบทั่วๆ ไปทำให้ผู้ใช้ในทุกๆ ระดับสามารถตรวจสอบแผนภาพยูสเคสได้ง่ายขึ้น โดยสามารถใช้แผนภาพยูสเคสในการสื่อสารระหว่างทีมผู้พัฒนาเองเนื่องจากว่าสัญลักษณ์ต่างๆ ที่ใช้ในแผนภาพนี้เป็นมาตรฐานที่ทุกคนรู้จัก และแผนภาพก็มีรายละเอียดของระบบที่ครบถ้วน

แผนภาพยูสเคสในระดับบนสุดนั้นจะแสดงคอนเท็กซ์ของระบบทั้งหมดซึ่งเป็นมุมมองจากฟังก์ชันการทำงานไม่ใช่มุมมองต่อคลาส และวัตถุที่จะมีในระบบโดยฟังก์ชันการทำงานของระบบนั้นจะแทนด้วยยูสเคสในรูปที่ 2-8 แสดงขอบเขตของระบบที่อยู่ในกรอบสี่เหลี่ยมโดยมีวัตถุภายนอกกระจายอยู่นอกระบบ วัตถุภายนอกเหล่านี้จะสัมพันธ์กับยูสเคสผ่านโพรโตคอลของระบบ นอกจากนั้นยูสเคสหนึ่งๆ อยากจะมีความสัมพันธ์กับยูสเคสอื่นๆ ก็สามารถแทนได้โดยการลากเส้นแสดงความสัมพันธ์ระหว่างยูสเคส

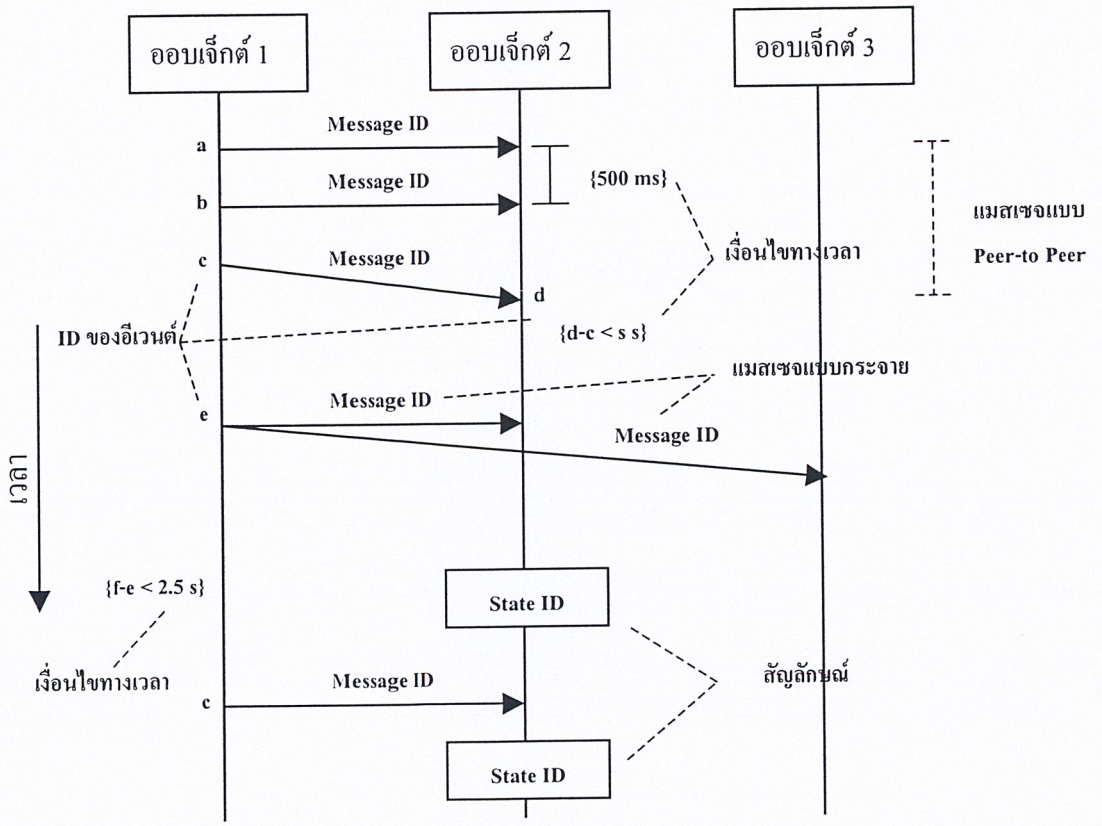


รูปที่ 2-8 แผนภาพยูสเคส

42835

2.3.2 แผนภาพแสดงลำดับขั้นตอนการทำงาน

แผนภาพแสดงขั้นตอนการทำงานจะแสดงลำดับของเมสเซจ (Message) ระหว่างวัตถุ โดยมีส่วนประกอบสำหรับแผนภาพแสดงลำดับขั้นตอนการทำงานโดยเส้นในแนวตั้งจะหมายถึงวัตถุ โดยจะมีชื่อของวัตถุเขียนอยู่ด้านบน หรือด้านล่างของเส้น ส่วนเส้นในแนวนอนนั้นแสดงเมสเซจ เส้นตรงที่แทนเมสเซจทุกเส้นจะต้องมีวัตถุต้นกำเนิดของเมสเซจส่วนจุดที่สิ้นสุดของเมสเซจก็จะเป็นวัตถุปลายทางโดยแต่ละเส้นของเมสเซจจะมีชื่อของเมสเซจกำกับอยู่โดยชื่อของเมสเซจนี้ในขั้นตอนต่อไปอาจจะมีการเพิ่มรายการของพารามิเตอร์ และค่าที่คืนกลับมา(Return Value) ด้วย สำหรับเวลานั้นจะนับจากข้างล่างดังแสดงในรูปที่ 2-9 โดยสเกลของแกนตั้งนั้นจะบอกเพียงลำดับของเมสเซจเท่านั้น ไม่สามารถใช้เป็นอัตราส่วนในการคำนวณหาเวลาระหว่างเมสเซจได้



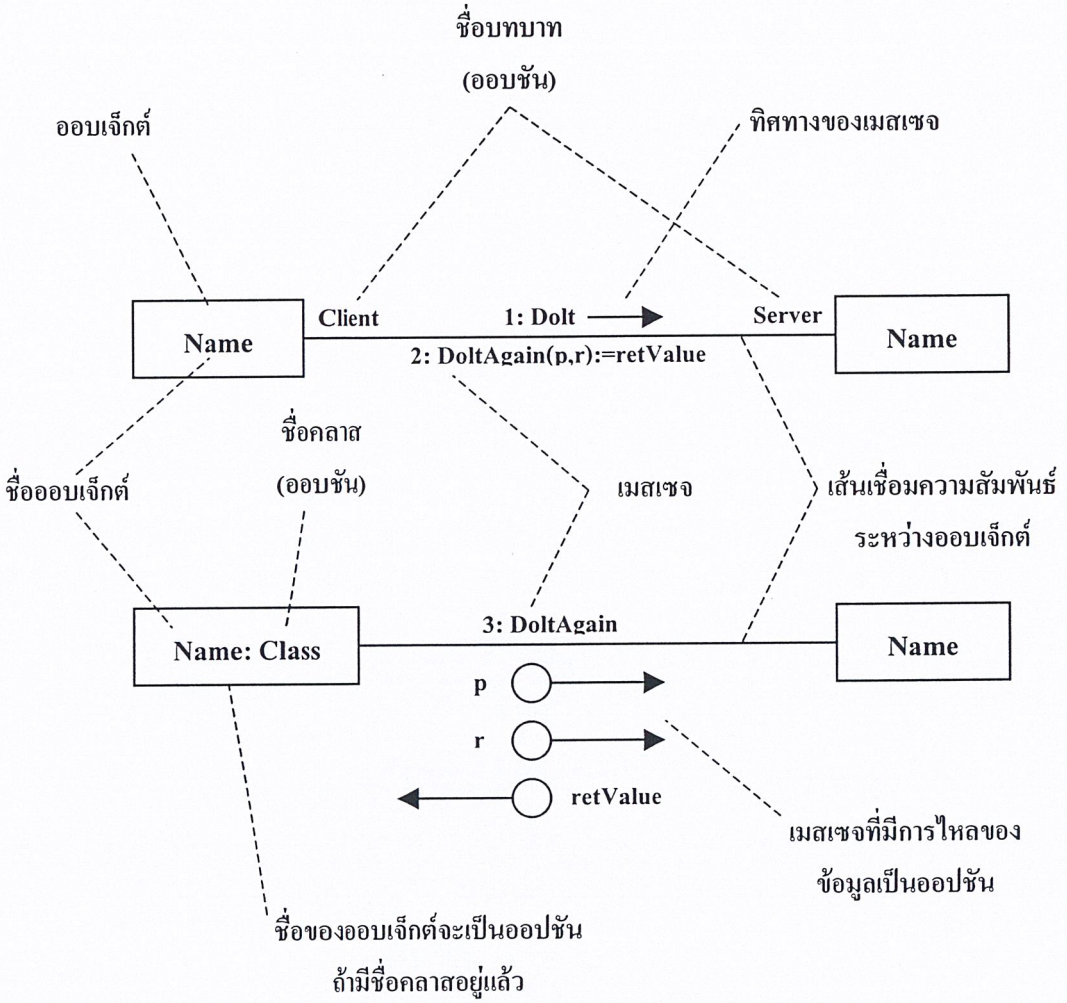
รูปที่ 2-9 แผนภาพแสดงขั้นตอนการทำงาน

2.3.3 แผนภาพแสดงการทำงานร่วมกันของวัตถุ

แผนภาพแสดงการทำงานร่วมกันของวัตถุเป็นเครื่องมือสำหรับใช้แสดงลำดับขึ้นตอนการทำงานอีกตัวหนึ่งที่มีประโยชน์ แผนภาพแสดงการทำงานร่วมกันของวัตถุนั้นจะแสดงข้อมูลพื้นฐานเช่นเดียวกับแผนภาพแสดงลำดับขั้นตอนการทำงานแต่ความแตกต่างระหว่างแผนภาพแสดงลำดับขั้นตอนการทำงาน

กับ แผนภาพแสดงการทำงานร่วมกันของวัตถุก็คือ แผนภาพแสดงลำดับขั้นตอนการทำงานจะแสดงลำดับของเมสเซจเป็นหลัก ส่วนแผนภาพแสดงการทำงานร่วมกันของวัตถุนั้นจะแสดงโครงสร้างของวัตถุที่ทำงานร่วมกันเป็นหลักดังรูปที่ 2-10 เป็นแผนภาพแสดงการทำงานร่วมกันของวัตถุอย่างง่าย

จากแผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุนั้นประกอบด้วยส่วนต่างๆ คือ ชื่อวัตถุ ,ความสัมพันธ์ระหว่างวัตถุที่มีการแลกเปลี่ยนเมสเซจกัน, เมสเซจที่ประกอบไปด้วย หมายเลขลำดับ ตัวบ่งบอก (Identifier) และทิศทางของเมสเซจ

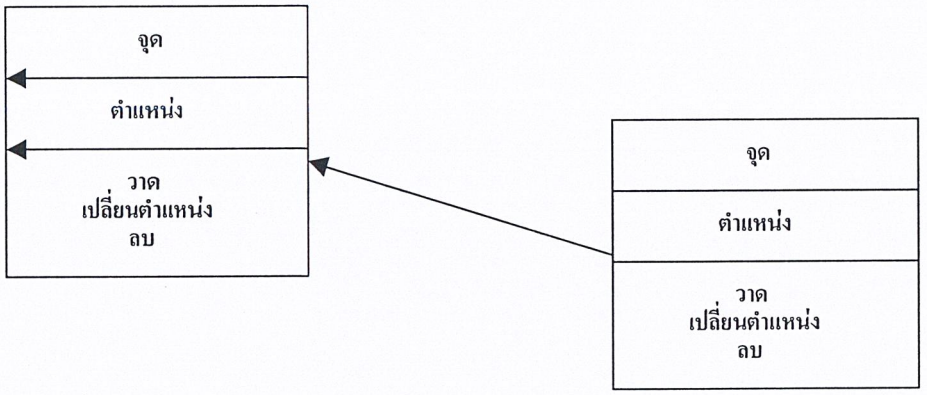


รูปที่ 2-10 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุ

2.3.4 แผนภาพของคลาส

แผนภาพของคลาสเป็นแผนภาพที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีเชิงวัตถุ เนื่องจากแผนภาพของคลาสจะแสดงโครงสร้างของวัตถุและคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย แผนภาพของคลาสจะเป็นโครงสร้างหลักของระบบและยังใช้ในการแยกย่อยรายละเอียด

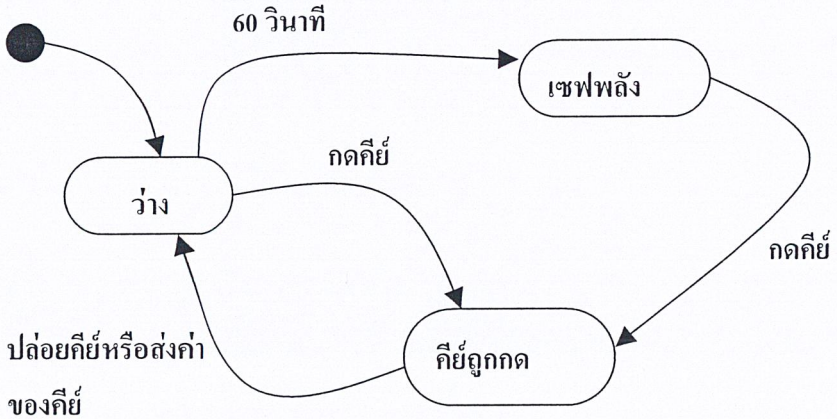
(Decomposite) ด้วยวิธีเชิงวัตถุ อย่างไรก็ดีแผนภาพของคลาสนั้นอาจจะมีหน้าตาที่คล้ายกับแผนภาพของวัตถุแต่แผนภาพของคลาสจะแสดงคลาส โดยไม่ได้แสดงอินสแทนซ์หรือวัตถุทั้งหมด และจะเน้นที่การแสดงโครงสร้างของคลาสมากกว่าความสัมพันธ์ระหว่างคลาสดังรูปที่ 2-11 เป็นตัวอย่างแสดงแผนภาพคลาสของเส้นและจุด



รูปที่ 2-11 แผนภาพคลาสของเส้นและจุด

2.3.5 แผนภาพแสดงสถานะ

ในการแสดงสถานะนั้นสามารถแสดงได้โดยการแสดงเมธอดสถานะ โดยสถานะมีความหมายคือที่ใช้เก็บเงื่อนไขของวัตถุไว้ในช่วงเวลาหนึ่งๆ โดยเงื่อนไขนี้จะมีความแตกต่างจากเงื่อนไขอื่นๆ และไม่มีความสัมพันธ์ต่อกันและกัน ความแตกต่างระหว่างสถานะในส่วนนี้หมายถึงในสถานะหนึ่งจะรับอีเวนต์ที่ต่างกันหรือมีการเปลี่ยนสถานะจากอีเวนต์ที่ได้รับต่างกันหรือจะมีการทำแอกชันสำหรับอีเวนต์นั้นที่ต่างกัน โดยการเปลี่ยนสถานะนั้นจะเกิดขึ้นเมื่อมีอีเวนต์ที่เหมาะสมเข้ามา

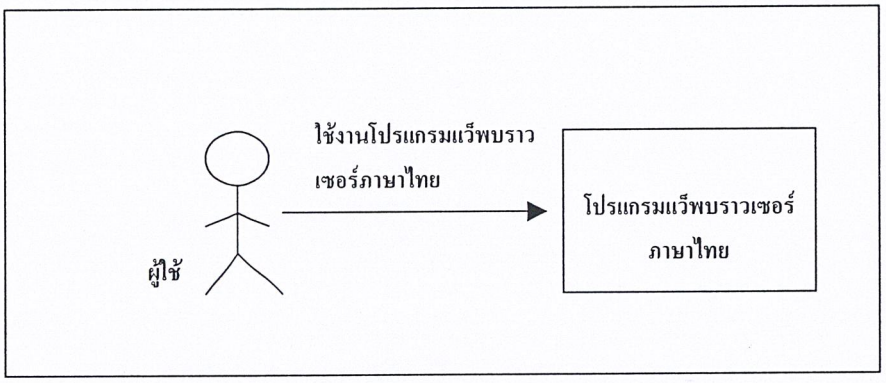


รูปที่ 2-12 แผนภาพแสดงสถานะของคีย์บอร์ด

การโมเดลวัตถุเป็นแผนภาพสถานะนั้นช่วยลดความซับซ้อนของตัววัตถุรวมทั้งการสร้างเมธอดลงได้ด้วย ดังตัวอย่างดังรูปที่ 2-12 ซึ่งประกอบด้วย 3 สถานะและที่การเปลี่ยนสถานะจำนวน 5 ทิศทาง

บทที่ 3
การสร้างและการออกแบบ

การออกแบบและการสร้างโปรแกรมแว็พบราวเซอร์ภาษาไทยในโครงงานนี้ได้ถูกออกแบบโดยการใช้หลักการวิเคราะห์และออกแบบเชิงวัตถุซึ่งใช้วิธี UML สำหรับลักษณะของความต้องการของโครงงานนี้คือ ทำให้ผู้ใช้สามารถเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่กำหนดได้ และสามารถสนับสนุนการแสดงผลเป็นภาษาไทยเพื่อจำลองการทำงานของโทรศัพท์มือถือที่สามารถแสดงผลเป็นภาษาไทยได้ นอกจากนี้ผู้ใช้ยังสามารถเขียนโปรแกรมภาษา WML และแสดงผลเพื่อจำลองการทำงานของโทรศัพท์มือถือตามที่ผู้ใช้เขียนโปรแกรมได้ ดังที่แสดงเป็นแผนภาพอธิบาย (Context Diagram) ของโปรแกรมแว็พบราวเซอร์ภาษาไทยดังรูปที่ 3-1



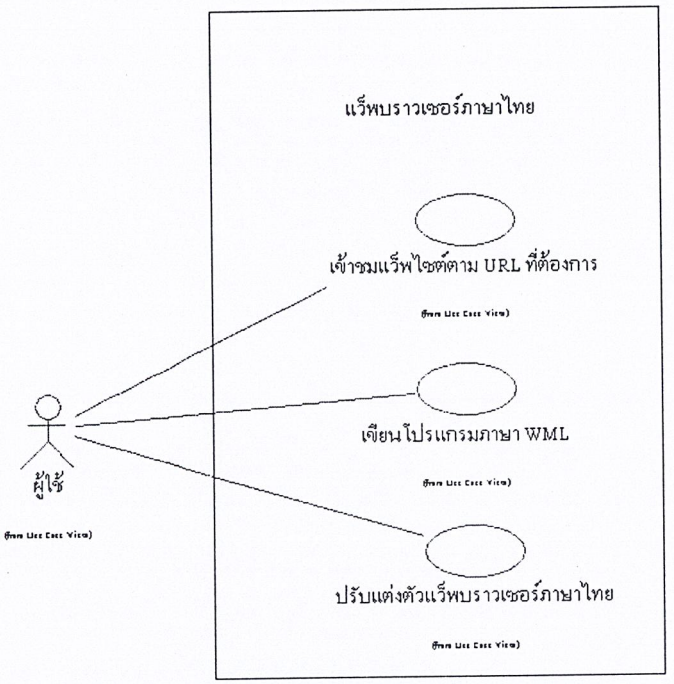
รูปที่ 3-1 แผนภาพคอนเท็กซ์ของแว็พบราวเซอร์ภาษาไทย

3.1 คอนเท็กซ์ไดอะแกรมของแว็พบราวเซอร์ภาษาไทย

แผนภาพคอนเท็กซ์ดังที่แสดงไว้ในรูปที่ 3-1 เป็นส่วนที่อธิบายการติดต่อระหว่างผู้ใช้ ซึ่งแสดงให้เห็นสิ่งแวดล้อมของโปรแกรมแว็พบราวเซอร์ภาษาไทย ผู้ใช้จะใช้โปรแกรมแว็พบราวเซอร์ในการเข้าชมแว็พไซต์ต่างๆ ตามที่ผู้ใช้ต้องการตามที่ URL ที่ต้องการจะเข้าไปชมแว็พไซต์เหล่านั้น และหากผู้ใช้ต้องการจะเขียนโปรแกรมภาษา WML ก็สามารถใช้เครื่องมือต่างๆที่มีอยู่บนโปรแกรมแว็พบราวเซอร์ในการพัฒนาโปรแกรมซึ่งอาจจะเป็นทั้งภาษาไทยหรือภาษาอังกฤษก็ได้ อีกทั้งยังสามารถกำหนดรูปแบบของหน้าจอของบราวเซอร์ได้ตามต้องการ เช่น เปลี่ยนรูปแบบฟอนต์, ขนาดของฟอนต์, สีของพื้นหลังหรือสีของตัวอักษรเหล่านี้เป็นต้น

3.2 ยูสเคส (Use Case) ของแว็พบราวเซอร์ภาษาไทย

แผนภาพยูสเคสของแว็พบราวเซอร์ภาษาไทยดังแสดงในรูปที่ 3-2 ซึ่งจะแสดงถึงรายละเอียดโดยรวมของโปรแกรมแว็พบราวเซอร์ภาษาไทย แผนภาพยูสเคสทำให้ทราบฟังก์ชันการทำงานหลักๆ ของโปรแกรมแว็พบราวเซอร์ว่าประกอบด้วย 3 ส่วนหลักๆ คือ ส่วนที่ผู้เข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการ ส่วนที่ผู้ใช้เขียนโปรแกรมภาษา WML และส่วนที่ผู้ใช้ปรับแต่งตัวของโปรแกรมแว็พบราวเซอร์ภาษาไทยโดยมีผู้ใช้เป็นแอกเตอร์ของระบบที่เข้าไปใช้งานในระบบซึ่งมีฟังก์ชั่นรวมต่างๆ เป็นยูสเคสมี 3 ส่วนหลักๆ ดังที่กล่าวมาแล้ว

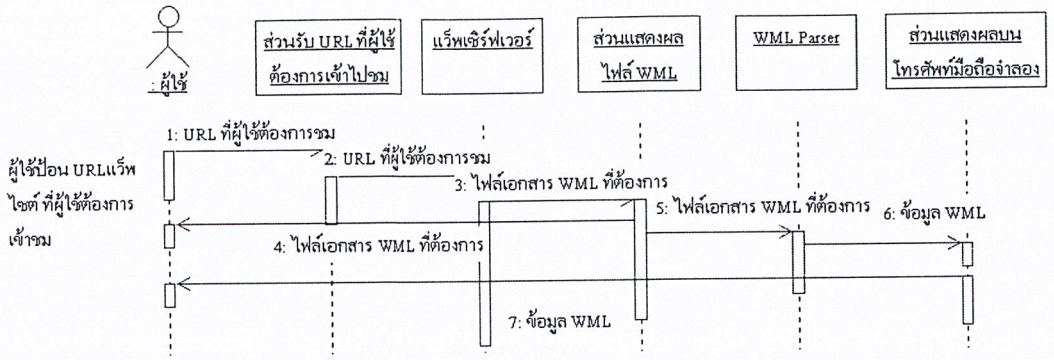


รูปที่ 3-2 แสดงยูสเคสของแว็พบราวเซอร์ภาษาไทย

3.3 ลำดับขั้นตอนการทำงานของแว็พบราวเซอร์ภาษาไทย

3.3.1 ลำดับขั้นตอนการทำงานของการเข้าชมแว็พไซต์ต่างๆตาม URL ที่ต้องการ

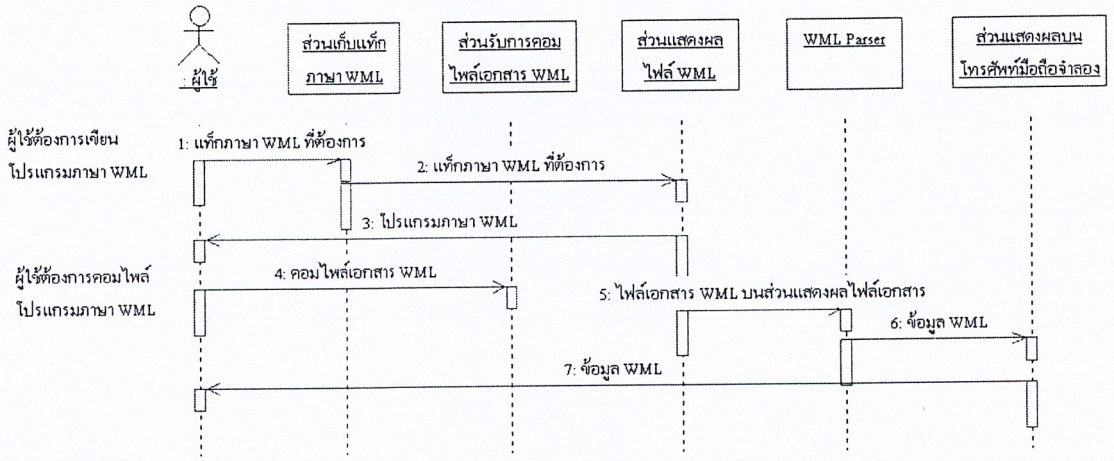
ในการเข้าชมแว็พไซต์ต่างๆตาม URL ที่ต้องการ เริ่มจากผู้ใช้ป้อน URL ที่ผู้ใช้ต้องการลงไปในช่องสำหรับป้อน URL ซึ่งจะส่งให้กับแว็พเซิร์ฟเวอร์ที่ถูกระบุไว้ใน URL ที่ผู้ใช้ต้องการชม เมื่อได้รับ URL ไปแว็พเซิร์ฟเวอร์จะส่งไฟล์เอกสาร WML มาให้ ตัวโปรแกรมจะนำไฟล์เอกสาร WML ดังกล่าวส่งให้กับส่วนแสดงผลไฟล์เอกสาร WML ซึ่งจะส่งแสดงผลให้กับผู้ใช้และขณะเดียวกันก็จะส่งไฟล์เอกสารนี้ให้กับ WAP Parser ทำการ Parse เอกสาร WML เพื่อนำมาแสดงผลบนโทรศัพท์มือถือที่จำลองการทำงานของมือถือเอาไว้ให้ผู้ใช้เห็น และดูผลลัพธ์จากส่วนแสดงผลบนโทรศัพท์มือถือที่จำลองการทำงานไว้ดังรูปที่ 3-3



รูปที่ 3-3 แผนภาพลำดับขั้นตอนการทำงานของการเข้าชมแว็พไซต์ต่างๆตาม URL ที่ต้องการ

3.3.2 ลำดับขั้นตอนการทำงานของการเขียนโปรแกรมภาษา WML บนแว็พบราวเซอร์ภาษาไทย

ในการเขียนโปรแกรมภาษา WML บนแว็พบราวเซอร์ภาษาไทย เมื่อผู้ใช้ต้องการเขียนโปรแกรมภาษาWML ผู้ใช้สามารถเขียนเองด้วยคีย์บอร์ดก็ได้หรือการใช้เครื่องมือปุ่มต่างๆ ซึ่งเป็นแท็กภาษา WML ที่อยู่บนเมนูแท็กภาษา WML ก็ได้ และเมื่อผู้ใช้เขียนโปรแกรมเสร็จแล้วก็สามารถคอมไพล์เพื่อดูผลที่ได้จากการเขียนโปรแกรม ซึ่งจะแสดงผลลัพธ์บนหน้าจอแสดงผลของโทรศัพท์มือถือจำลอง สำหรับการคอมไพล์ โปรแกรมจะอ่านข้อมูลที่เป็นไฟล์เอกสาร WML ส่งให้ WML Parser เพื่อ Parse และแสดงผลลัทธ์บนหน้าจอโทรศัพท์มือถือจำลองให้กับผู้ใช้ เหมือนกับการป้อน URL เข้าไปนั่นเอง ดังแสดงในรูปที่

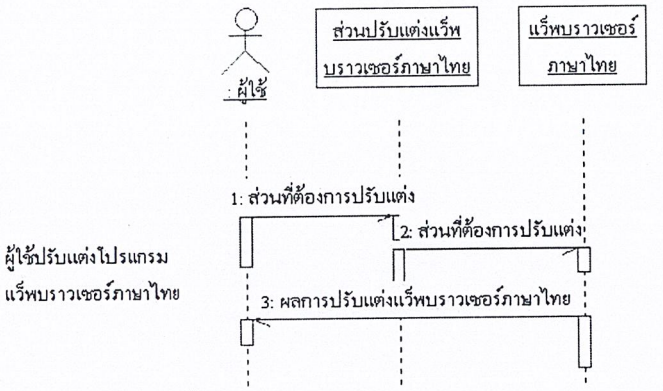


3-4

รูปที่ 3-4 แผนภาพแสดงลำดับขั้นตอนการทำงานของการเขียนโปรแกรมภาษา WML บนแว็พบราวเซอร์ภาษาไทย

3.3.3 ลำดับขั้นตอนการทำงานของการปรับแต่งแว็พบราวเซอร์ภาษาไทย

สำหรับการปรับแต่งแว็พบราวเซอร์ เมื่อผู้ใช้ต้องการปรับแต่งส่วนของโปรแกรม ผู้ใช้ก็สามารถปรับแต่งในส่วนที่ต้องการ แล้วผลของการปรับแต่งโปรแกรมแว็พบราวเซอร์ภาษาไทยก็จะแสดงให้ผู้ใช้เห็น ดังรูปที่ 3-5

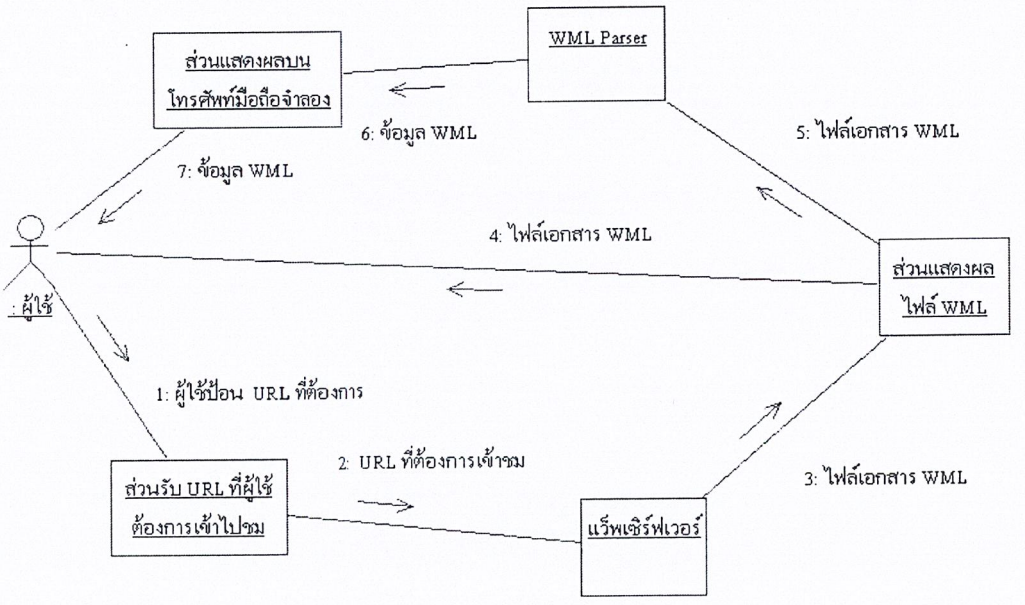


รูปที่ 3-5 แผนภาพแสดงลำดับขั้นตอนการทำงานของการปรับแต่งแว็พบราวเซอร์ภาษาไทย

3.4 การทำงานร่วมกันระหว่างวัตถุของแว็พบราวเซอร์ภาษาไทย

3.4.1 การทำงานร่วมกันระหว่างวัตถุของการเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการ

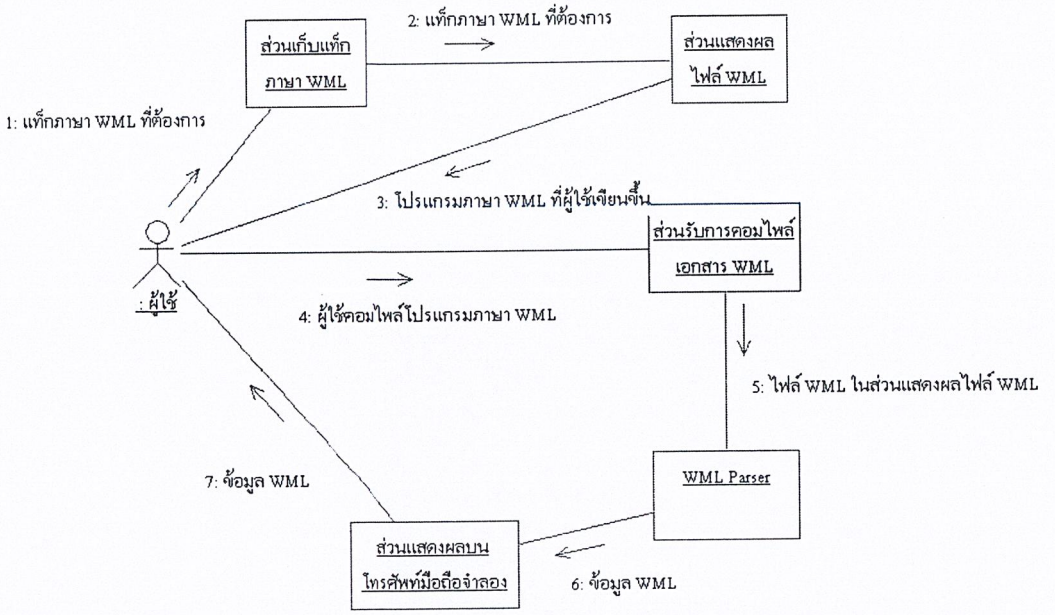
การทำงานโดยรวมได้อธิบายไว้แล้วในส่วนของแผนภาพแสดงลำดับขั้นตอนการทำงาน สำหรับแผนภาพนี้จะทำให้เห็นความสัมพันธ์ระหว่างวัตถุในแต่ละส่วนได้อีกมุมมองหนึ่งได้ดังแสดงในรูปที่ 3-6



รูปที่ 3-6 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการ

3.4.2 การทำงานร่วมกันระหว่างวัตถุของการเขียนโปรแกรมของผู้ใช้บนแว็พบราวเซอร์ภาษาไทย

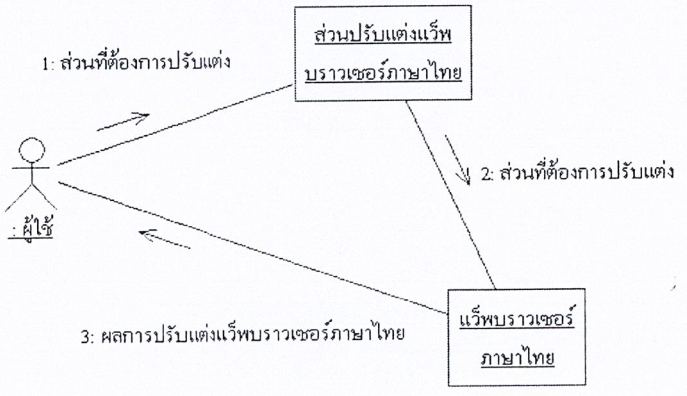
ดังรูปที่ 3-7 แสดงให้เห็นว่าเมื่อต้องการเขียนโปรแกรมภาษา WML บนโปรแกรมแว็พบราวเซอร์ภาษาไทย วัตถุในแต่ละส่วนจะสัมพันธ์และทำงานร่วมกันได้



รูปที่ 3-7 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการเขียนโปรแกรมของผู้ใช้บนแว็พบราวเซอร์ภาษาไทย

3.4.3 การทำงานร่วมกันระหว่างวัตถุของการปรับแต่งแว็พบราวเซอร์ภาษาไทย

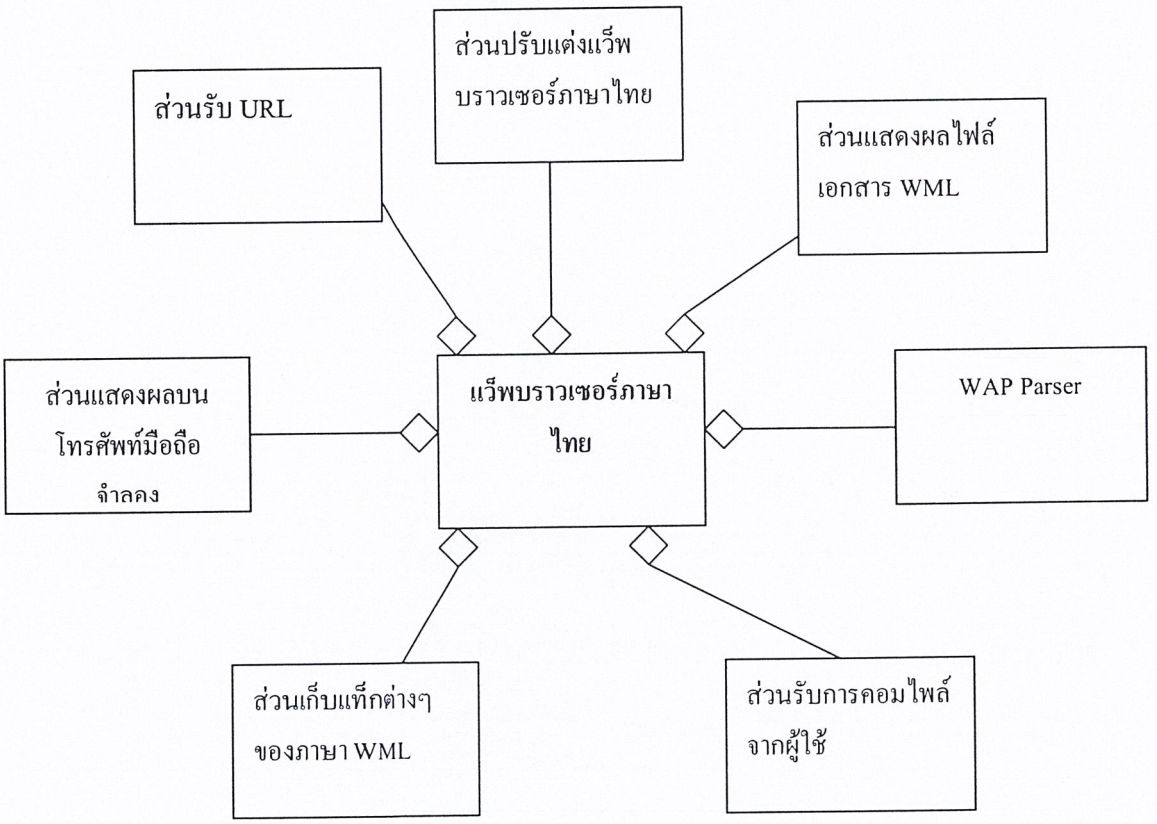
ในแผนภาพนี้จะทำให้เห็นว่าวัตถุต่างๆ ในระบบของทำงานร่วมกันได้เมื่อต้องการปรับแต่งซึ่งจะแสดงให้เห็นดังรูปที่ 3-8



รูปที่ 3-8 แผนภาพแสดงการทำงานร่วมกันระหว่างวัตถุของการปรับแต่งแว็พบราวเซอร์ภาษาไทย

3.5 คลาสต่างๆ ของแว็พบราวเซอร์ภาษาไทย

จากการวิเคราะห์พบว่าแว็พบราวเซอร์ภาษาไทยในโครงงานนี้ประกอบไปด้วยส่วนหลักๆ แบ่งได้เป็น ส่วนที่รับ URL ซึ่งทำหน้าที่ในการรับ URL จากผู้ใช้เพื่อส่งต่อให้กับ WAP Parser, ส่วนแสดงผลบนโทรศัพท์มือถือจำลองจะเป็นตัวทำหน้าที่ในการแสดงผลและทำงานตามแบบจำลองการทำงานของโทรศัพท์มือถือ, ส่วนเมนูแท็กต่างๆ ของภาษา WML จะทำหน้าที่เก็บแท็กต่างๆ ของภาษา WML และแสดงแท็กต่างๆ ตามที่ผู้ใช้ต้องการให้กับส่วนแสดงผลไฟล์เอกสาร WML, ส่วนรับการคอมไพล์จากผู้ใช้เป็นส่วนที่รับการติดต่อกับผู้ใช้เพื่อให้ผู้ใช้ได้คอมไพล์ไฟล์เอกสารที่แสดงบนส่วนแสดงผลไฟล์เอกสาร WML, ส่วนแสดงผลไฟล์เอกสาร WML มีหน้าที่แสดงไฟล์เอกสาร WML ที่ผู้ใช้เขียนขึ้นหรือที่ได้จาก URL, ส่วน WAP Parser จะทำหน้าที่ในการParse ไฟล์เอกสาร WML เพื่อแสดงผลบนโทรศัพท์มือถือจำลอง

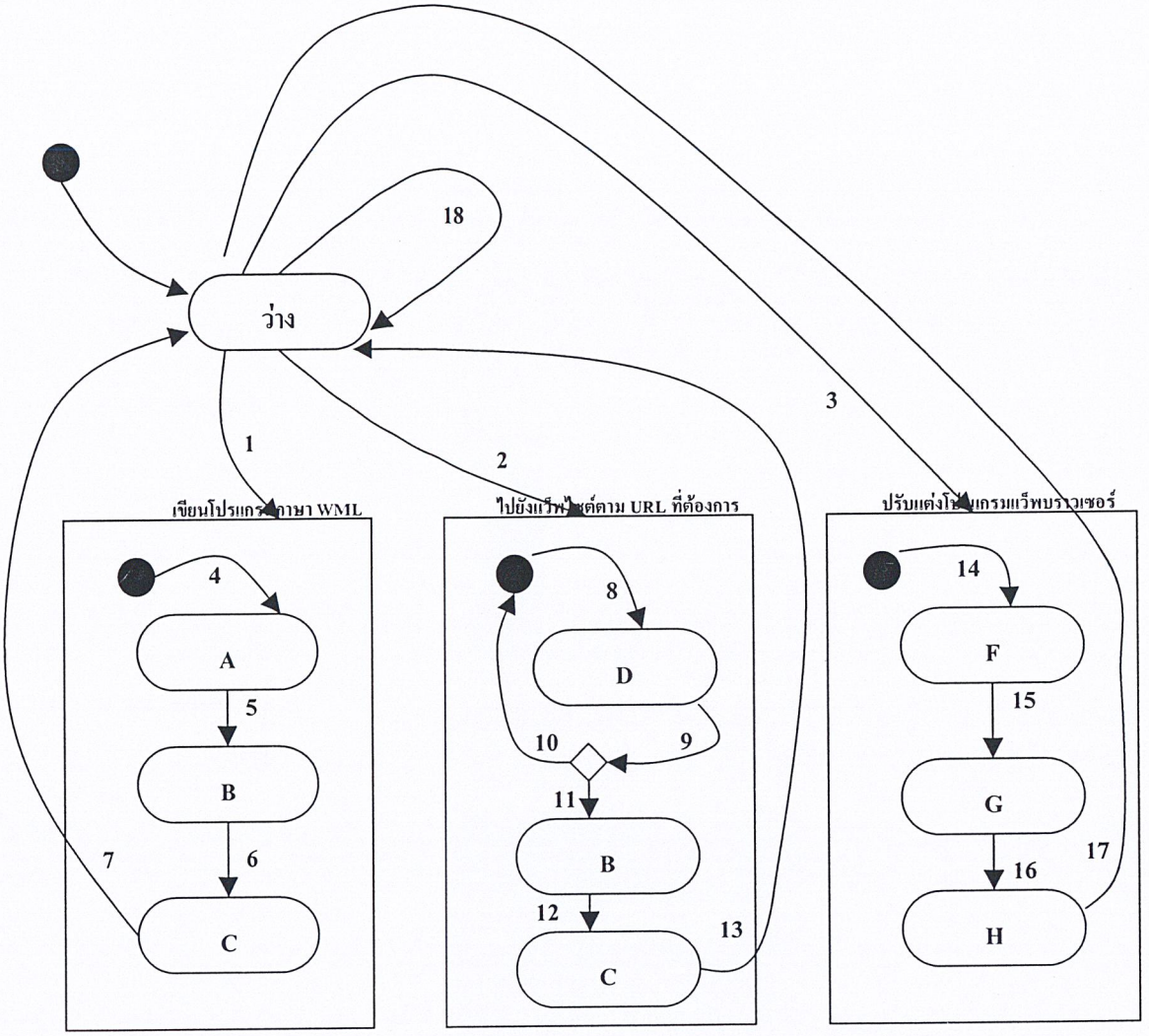


รูปที่ 3-9 แผนภาพคลาสของแว็พบราวเซอร์ภาษาไทย

3.6 ไดอะแกรมสถานะของแว็พบราวเซอร์ภาษาไทย

แผนภาพสถานะแสดงให้เห็นว่าแว็พบราวเซอร์ภาษาไทยนั้นประกอบด้วย 4 สถานะหลักๆ คือ สถานะว่างเป็นสถานะที่โปรแกรมแว็พบราวเซอร์ภาษาไทยเพื่อรอให้ผู้ใช้มาใช้งานโปรแกรม สถานะการเขียนโปรแกรมเป็นสถานะที่ผู้ใช้เข้าสู่โหมดการเขียนโปรแกรมให้ผู้ใช้จะเขียนโปรแกรม จากนั้นจะเข้าสู่สถานะแสดงผลข้อมูลไฟล์เอกสาร WML และรอการคอมไพล์ซึ่งจะต้องมีไฟล์เอกสาร WML มาด้วย ในช่วงนี้จะรับไฟล์เอกสาร WML มาคอมไพล์และส่งข้อมูล WML มาแสดงผลในส่วนแสดงผลโทรศัพท์มือ

ถือจำลอง ถ้าเป็นสถานะของการเข้าชมแว็พไซต์ต่างๆ ตาม URL ที่ต้องการก็จะรอรับ URL ที่ผู้ใช้ต้องการแล้วส่ง URL ที่ต้องการดูไปยังแว็พเซิร์ฟเวอร์ที่ต้องการ และถ้าตรวจพบว่า URL ที่ต้องการพบก็จะส่งไฟล์เอกสาร WML นั้นให้กับส่วนคอมไพล์ไฟล์เอกสาร WML และแสดงผลบนส่วนแสดงผลโทรศัพท์มือถือจำลองและถ้าไม่พบก็จะแสดงผลให้ผู้ใช้เห็นว่าเกิดความผิดพลาด ดังรูปที่ 3-10



รูปที่ 3-10 แผนภาพสถานะของแว็พบราวเซอร์ภาษาไทย

A = รับข้อมูลแท็กภาษา WML มาแสดงผลบนส่วนแสดงผลเอกสาร WML

B= รับไฟล์เอกสาร WML มาทำการ Parse

C= แสดงผลข้อมูล WML บนโทรศัพท์มือถือจำลอง

D= แสดง URL ที่ผู้ใช้ต้องการจะไป

F= รับส่วนที่ต้องการปรับแต่งบราวเซอร์

G= ปรับแต่งส่วนที่ต้องการปรับแต่งบราวเซอร์

H= แสดงส่วนปรับแต่งบราวเซอร์

1= ผู้ใช้ต้องการเขียนโปรแกรม

2= ผู้ใช้ต้องการไปยังแว็พไซต์ที่ต้องการ

3= ผู้ใช้ต้องการปรับแต่งบราวเซอร์

4= ข้อมูลแท็กภาษา WMLของผู้ใช้

5=ไฟล์เอกสาร WML ^ ร้องขอการคอมไพล์

6= ข้อมูล WML

7=ข้อมูล WML

8= URL ที่ต้องการจะไป

9= ตรวจหา URL ว่ามีหรือไม่

10= [ไม่มี] / แสดงข้อผิดพลาด

11= [มี] ^ ไฟล์เอกสาร WML จากแว็พเซิร์ฟเวอร์

12= ข้อมูล WML

13= ข้อมูล WML

14= ส่วนที่ต้องการปรับแต่ง

15=ส่วนที่ต้องการปรับแต่ง

16 =ส่วนที่ต้องการปรับแต่ง

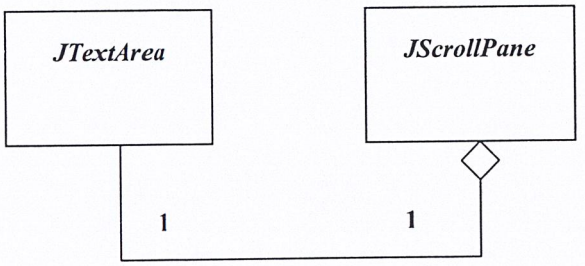
17= ส่วนที่ต้องการปรับแต่ง

18 = แสดงบราวเซอร์

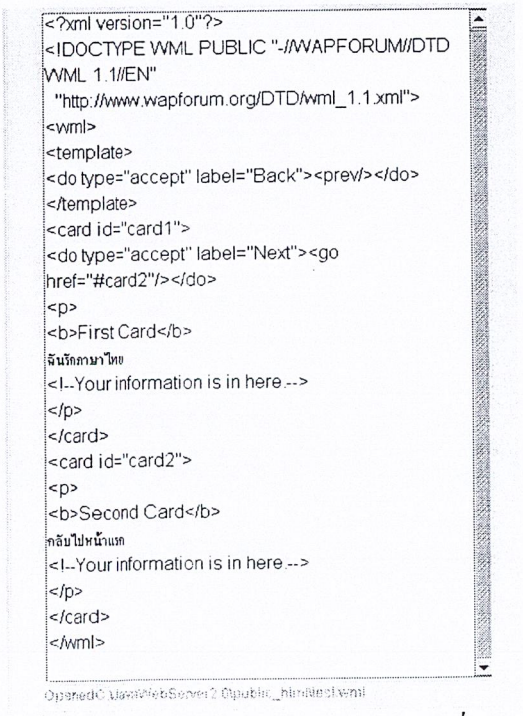
3.7 คลาสต่างๆ ที่สร้างขึ้นในโปแกรมแว็พบราวเซอร์ภาษาไทย

3.7.1 ส่วนแสดงผลไฟล์เอกสาร WML

ในส่วนแสดงผลไฟล์เอกสาร WMLนี้ได้ใช้คลาสของภาษาจาวาในการสร้างโดยประกอบไปด้วย 2 คลาสคือ คลาส JScrollPane และคลาส JTextArea ซึ่งมีความสัมพันธ์ดังรูปที่ 3-11 และรูปที่ 3-12 แสดงในส่วนแสดงผลในโปรแกรมแว็พบราวเซอร์ภาษาไทยที่ได้สร้างขึ้น



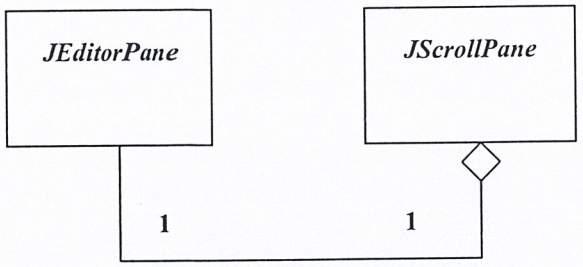
รูปที่ 3-11 ความสัมพันธ์ระหว่าง JTextArea และ JScrollPane



รูปที่ 3-12 ส่วนแสดงผลไฟล์เอกสาร WML ที่ได้สร้างขึ้น

3.7.2 ส่วนแสดงผลบนโทรศัพท์มือถือจำลอง

ในส่วนของการแสดงผลบนโทรศัพท์มือถือจำลองที่ได้สร้างขึ้นมานั้นเป็นส่วนที่ไฟล์เอกสาร HTML ที่ถูกแปลงมาจากส่วน WAP Parser นำมาแสดงผลซึ่งได้ใช้คลาส JEditorPane และคลาส JScrollPaneในการสร้างซึ่งได้แสดงความสัมพันธ์ดังรูปที่ 3-13 และรูปที่ 3-14 แสดงส่วนที่ได้สร้างขึ้นมาเป็นโปรแกรมเรียบร้อยแล้ว

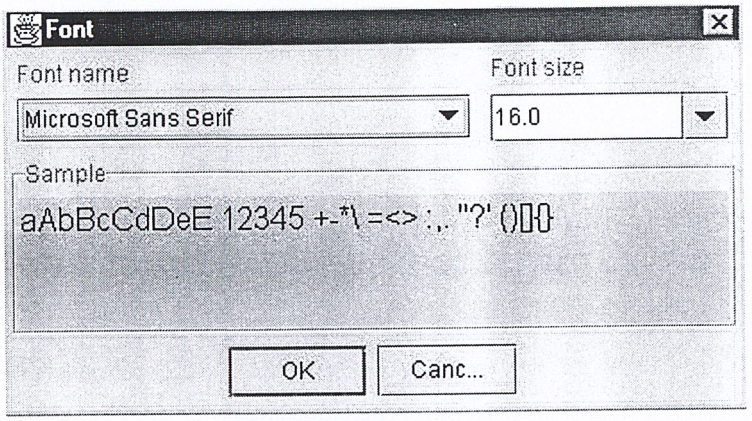


รูปที่ 3-13 แสดงความสัมพันธ์ระหว่าง JEditorPane และ JScrollPane

รูปที่ 3-14 ส่วนแสดงผลบนโทรศัพท์มือถือจำลอง

3.7.3 ส่วนปรับแต่งแว็พบราวเซอร์ภาษาไทย

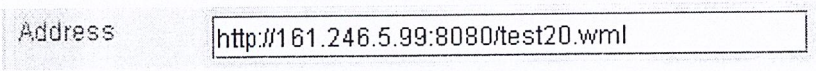
ในส่วนนี้เราได้ใช้คลาสต่างๆหลายคลาสเช่นการปรับแต่งขนาดของตัวอักษรและรูปแบบตัวอักษรซึ่งได้ใช้คลาส JfontChooser ในการปรับแต่งสิ่งเหล่านี้ซึ่งได้แสดงเมื่อคลาสเหล่านี้ถูกใช้งาน



รูปที่ 3-15 ผลจากคลาส JFontChooser ถูกเรียกใช้งาน

3.7.4 ส่วนการรับ URL จากผู้ใช้

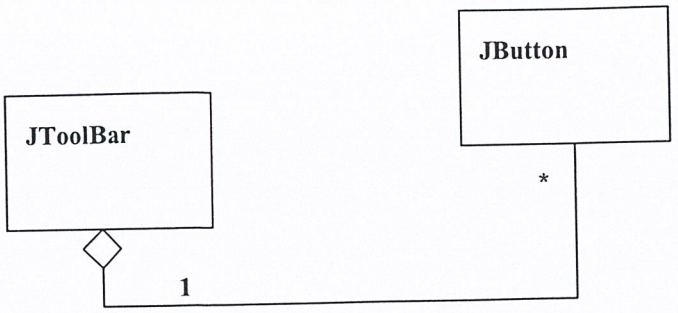
ในการรับ URL จากผู้ใช้นั้นเราได้ใช้คลาส JtextField ในการรับค่า URL จากผู้ใช้และส่งค่าตัวแปรที่ได้รับโดยใช้แมสทอด(Method) getText ส่งให้กับส่วนอื่นๆ ต่อไปดังแสดงในรูปที่ 3-16



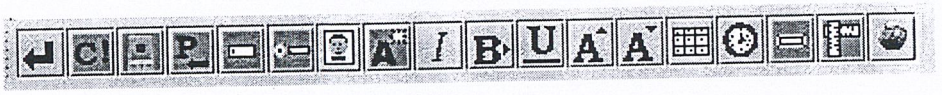
รูปที่ 3-16 ส่วนการรับ URL จากผู้ใช้

3.7.5 ส่วนเก็บแท็กต่างๆ ของภาษา WML

ในส่วนเก็บแท็กต่างๆ ของภาษา WML เพื่อให้ผู้ใช้ได้เรียกมาใช้งานนั้นเราได้เก็บแท็กต่างๆ โดยแท็กต่างๆเหล่านี้จะใช้คลาส JButton ในการเก็บแท็กแต่ละแท็กเอาไว้เพื่อให้ผู้ใช้กดปุ่มแล้วแสดงผลในส่วนแสดงผลไฟล์เอกสาร WML โดยการดักจับ ActionListener ของ JButton และจัดกลุ่มแท็กต่างๆไว้รวมกันไว้ในคลาส JToolBar ซึ่งสามารถแสดงเป็นความสัมพันธ์ระหว่างคลาสได้ดังรูปท์ 3-17 และรูปที่ 3-18 แสดงส่วนที่อยู่ในโปรแกรมแว็พบราวเซอร์ภาษาไทยที่ได้สร้างขึ้น



รูปที่ 3-17 แสดงความสัมพันธ์ระหว่าง JButton และ JToolBar



รูปที่ 3-18 แสดงปุ่มของแท็กต่างๆ บนแว็พบราวเซอร์ภาษาไทย

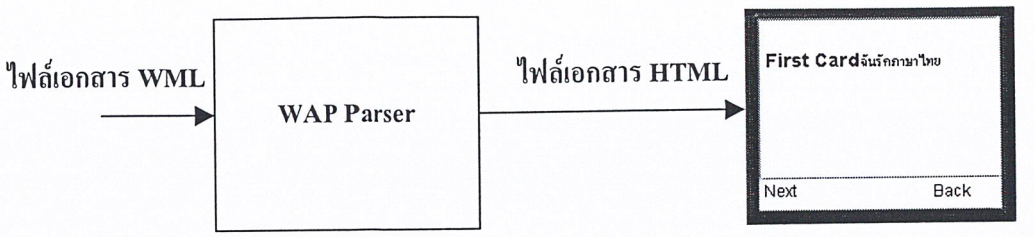
3.7.6 ส่วนรับการคอมไพล์จากผู้ใช้

เป็นส่วนที่รับการคอมไพล์เมื่อผู้ใช้เขียนโปรแกรมเสร็จแล้วและต้องการดูผลการจำลองการทำงานในส่วนโทรศัพท์มือถือจำลองโดยใช้คลาส JButton ในการทำและส่งแอคชันให้กับส่วน WAP Parser รับไฟล์เอกสาร WML ในส่วนแสดงผลไฟล์เอกสาร WML ไปทำการแปลงเป็นไฟล์ HTML ดังแสดงในรูปที่ 3-19

รูปที่ 3-19 แสดงส่วนรับการคอมไพล์จากผู้ใช้

3.7.7 ส่วน WAP Parser

โดยเหตุผลที่เราต้องการจำลอง การทำงานของโทรศัพท์มือถือโดยที่โครงงานนี้ได้แสดงบน JEditorPane ซึ่งเป็นคลาสๆหนึ่งของคอมโพแนนต์ SWING โดยการ import javax.swing.JEditorPane ในภาษาจาวา ซึ่งสามารถแสดงผลเป็นเอกสารไฟล์ HTML ได้ ซึ่งหลักการเพื่อให้เราสามารถจำลองการทำงานบนโทรศัพท์มือถือจำลองได้นั้นเราจะต้องทำการแปลงไฟล์เอกสาร WML ที่ได้รับมาจากเซิร์ฟเวอร์หรือจากโปรแกรมที่ผู้ใช้เขียนขึ้นมาเป็นไฟล์เอกสารไฟล์ HTML ซึ่งสามารถแสดงผลบนคลาส JEditorPane ได้ แต่การแสดงผลบางแท็กอาจไม่ได้ตามที่ต้องการ เนื่องจากลักษณะการโต้ตอบของ WML จะไม่เหมือนของ HTML



รูปที่ 3-20 โครงสร้างการแปลง (Conversion) ของ WAP Parser

การแปลงจากแท็กของภาษา WML เป็นแท็กของภาษา HTML

เนื่องจากภาษา WML ได้ถูกกำหนดขึ้นโดยมีภาษา HTML เป็นต้นแบบ การแปลงจากแท็กของภาษา WML เป็นแท็กของภาษา HTML สำหรับบางแท็กจึงไม่ค่อยซับซ้อนนัก แต่ในบางแท็กต้องอาศัยความเข้าใจในการแสดงผลของแท็กต่างๆ ในภาษา WML เพื่อจัดการการแสดงผลบนหน้าจอมือถือจำลองได้อย่างถูกต้องตามไวยากรณ์ของภาษา จึงแปลงเฉพาะบางแท็กที่เป็นการแสดงผลออกสู่ JEditorPane สำหรับบางแท็กที่ไม่มีใน HTML ส่วนนั้นได้จัดการเขียนโปรแกรมควบคุมการแสดงผลเอง

แท็กที่ถูกทำการแปลงได้แก่แท็กที่แสดงในตารางที่ 3-1 ข้างล่าง

WML	HTML
<wml> มีแท็กปิด	<html> มีแท็กปิด
<card id="name" title="label"> มีแท็กปิด	<body title="title"> มีแท็กปิด
<head> มีแท็กปิด	เหมือนกัน
<p align="alignment"> มีแท็กปิด	เหมือนกัน
<a href="href"> มีแท็กปิด	เหมือนกัน
<b> มีแท็กปิด	เหมือนกัน
<big> มีแท็กปิด	เหมือนกัน
  ไม่มีแท็กปิด หรือ ไม่มีแท็กปิด	 ไม่มีแท็กปิด
<i> มีแท็กปิด	เหมือนกัน
<input name="variable" type="type" value="value"/> ไม่มีแท็กปิด	เหมือนกันไม่มี / ปิดท้าย
	เหมือนกันไม่มี / ปิดท้าย
มีแท็กปิด	เหมือนกัน
มีแท็กปิด	เหมือนกัน
<small> มีแท็กปิด	เหมือนกัน

WML	HTML
<u> มีแท็กปิด	เหมือนกัน
<table columns="number" align="alignment"> มีแท็กปิด	เหมือนกัน
<td> มีแท็กปิด	เหมือนกัน
<tr> มีแท็กปิด	เหมือนกัน

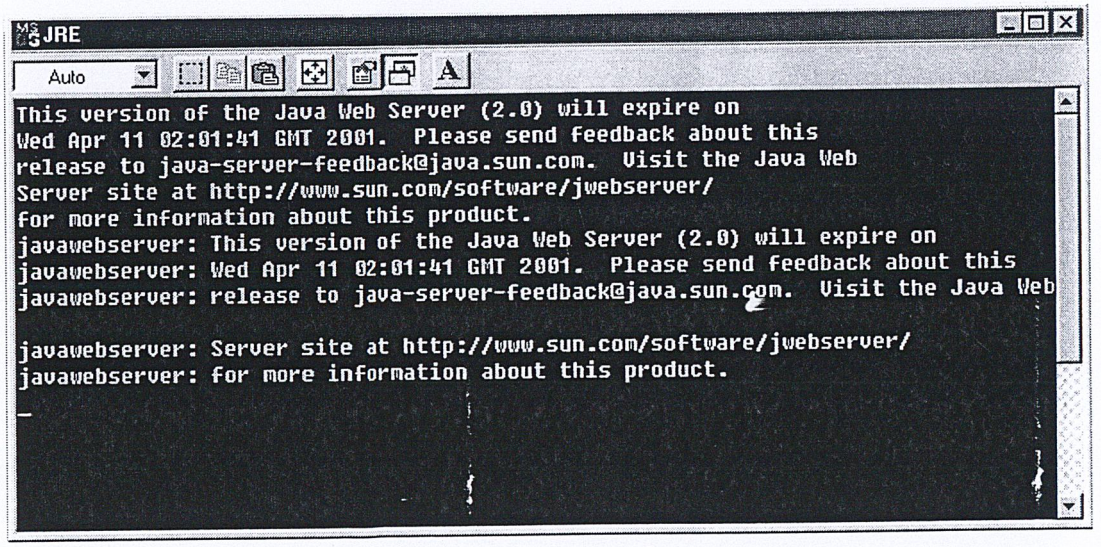
ตารางที่ 3-1 แท็กของภาษา WML ที่ถูกแปลงเป็น HTML

แท็กที่ได้เขียนโปรแกรมจัดการการแสดงผลเองได้แก่ <go....>, <do.....>, <prev/>, <refresh>, <noop/> ส่วนแท็กที่ยังทำการแสดงผลไม่ได้ได้แก่ <select>, <option>, <optgroup>, <fieldset>, <timer>, <setvar>, <anchor>, <setvar>

3.8 การสร้างแว็พบราวเซอร์ภาษาไทย

ในการสร้างแว็พบราวเซอร์ภาษาไทยโครงงานนี้ได้เขียนขึ้นมาด้วยภาษาจาวาซึ่งมีขั้นตอนต่างๆ ในการทำโครงงานดังนี้

3.7.1 ติดตั้งเซิร์ฟเวอร์เพื่อใช้ในการทำโครงงานนี้โดยต้องการให้ไฟล์เอกสาร WML ที่เขียนขึ้นนั้นเป็นภาษาไทยเพื่อทดลองกับแว็พบราวเซอร์ภาษาไทยโดยโครงงานนี้เลือกใช้แว็พเซิร์ฟเวอร์ของ JavaWebSever 2.0 ในการทำเป็นแว็พเซิร์ฟเวอร์ หลังจากติดตั้งเซิร์ฟเวอร์แล้ว ทำการรันเซิร์ฟเวอร์โดยเรียกจาก Dos Prompt ด้วยคำสั่ง c:\javawebserver2.0\bin\httpd.exe ถ้าเซิร์ฟเวอร์รันได้ถูกต้องไม่มีปัญหาจะให้ผลลัพธ์ดังรูปที่ 3-21



รูปที่ 3-21 แสดงผลการรัน JavaWebServer 2.0 อย่างถูกต้อง

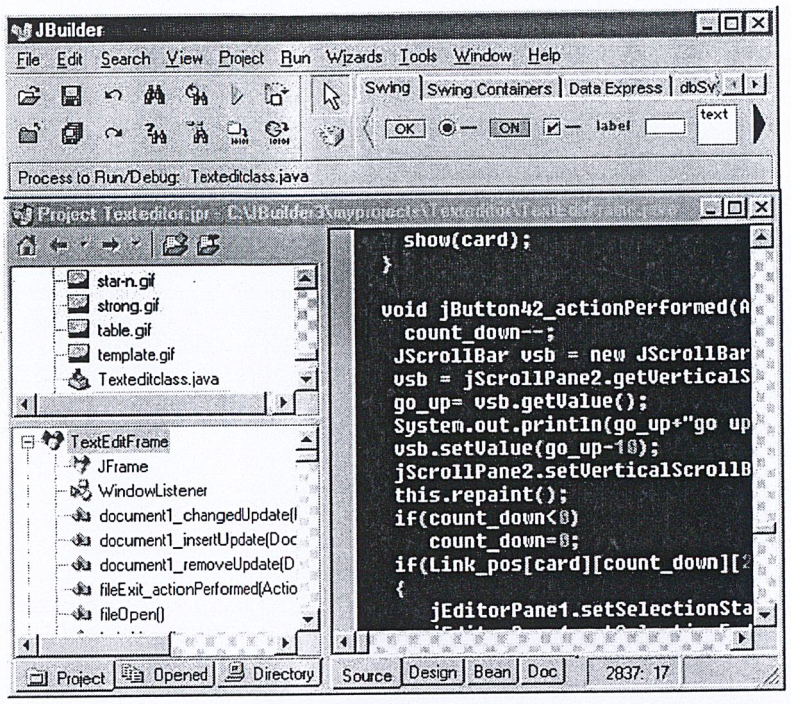
สำหรับไฟล์ที่ใช้ทดสอบจะถูกจัดเก็บอยู่ในไดเร็กทอรี่ public_html โดยการเรียกใช้ข้อมูลจากเซิร์ฟเวอร์จะใช้โพรโตคอล HTTP พอร์ต 8080 ดังนี้ http://161.246.5.99:8080/test.wml

3.7.2 ติดตั้งโปรแกรมตัวคอมไพเลอร์ที่สนับสนุนภาษาไทยซึ่งในโครงงานนี้ได้ใช้ภาษาจาวาเวอร์ชัน 1.2.2 ภาษาไทย

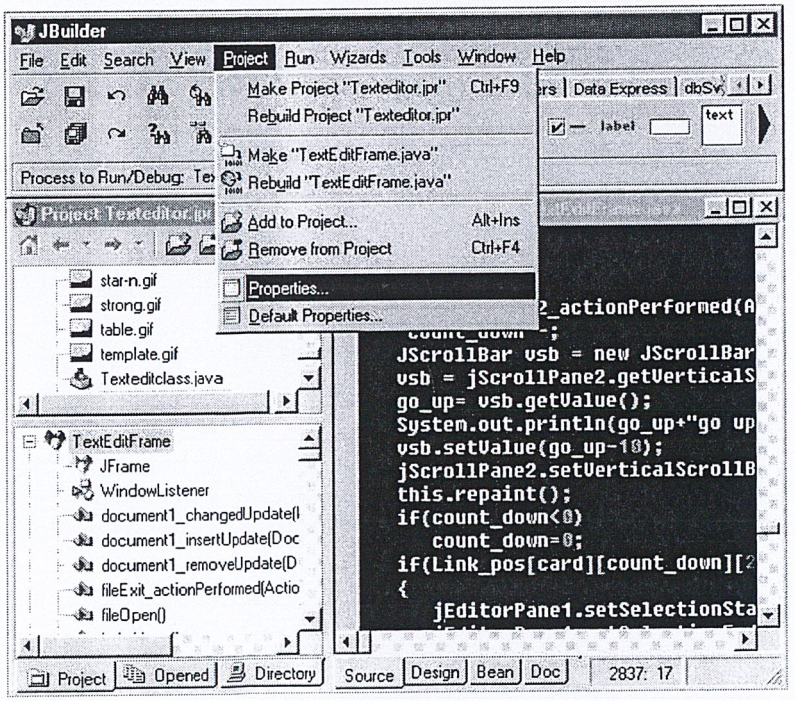
หลังจากติดตั้งเสร็จแล้วให้ทำการเซตฟอนต์ของคอมไพเลอร์ชื่อไฟล์ font.propoties.th ซึ่งเก็บอยู่ในไดเร็กทอรี่ย่อย c:\jdk1.2.2\jre\lib\ โดยเซตดังนี้

```
dialog.0=MS Sans Serif, ANSI_CHARSET
dialog.1=Angsana New, THAI_CHARSET,NEED_CONVERTED
dialog.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
dialog.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED

dialoginput.0=Cordia New, ANSI_CHARSET
dialoginput.1=Cordia New, THAI_CHARSET,NEED_CONVERTED
dialoginput.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
dialoginput.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED

serif.0=MS Sans Serif, ANSI_CHARSET
serif.1=Angsana New, THAI_CHARSET,NEED_CONVERTED
serif.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
serif.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED

sansserif.0=MS Sans Serif, ANSI_CHARSET
sansserif.1=Angsana New, THAI_CHARSET,NEED_CONVERTED
sansserif.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
sansserif.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED

monospaced.0=Courier New, ANSI_CHARSET
monospaced.1=Cordia New, THAI_CHARSET,NEED_CONVERTED
monospaced.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
monospaced.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED
```

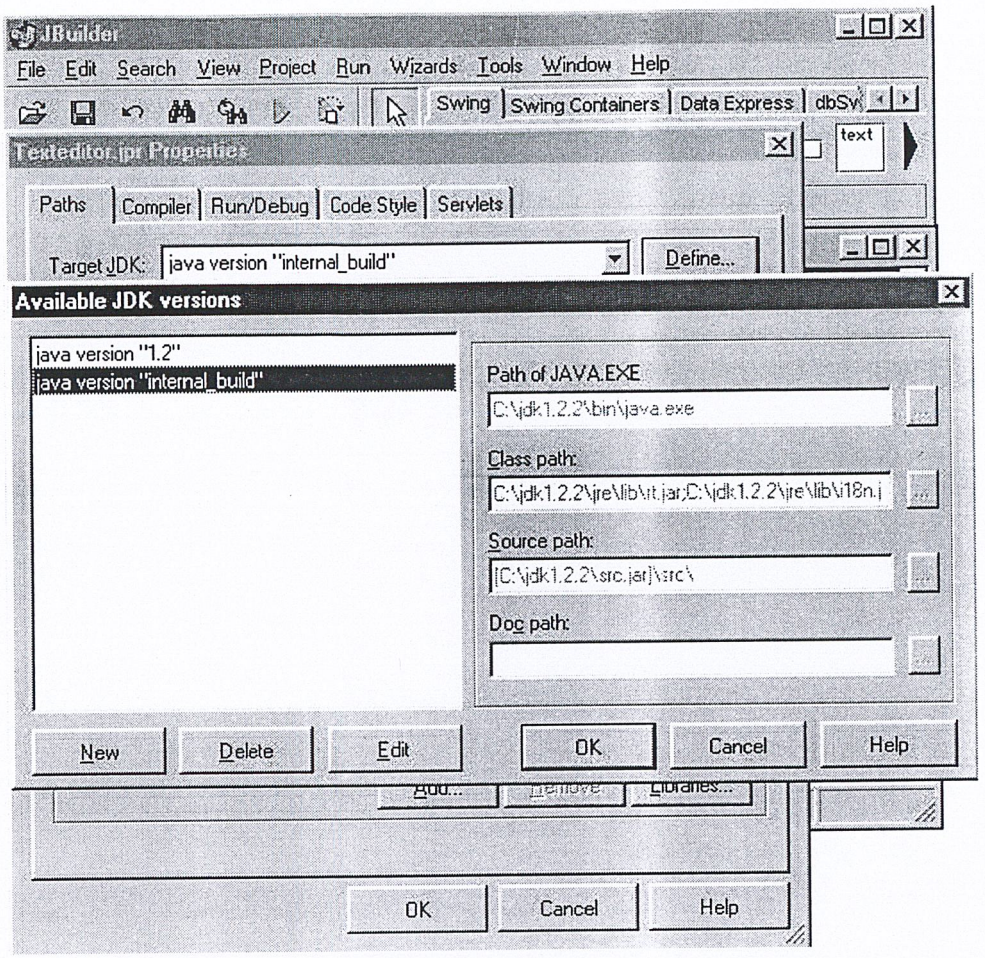
3.7.3 ติดตั้งโปรแกรม Jbuilder 3.0 เพื่อเป็นเครื่องมือในการสร้างแว็พบราวเซอร์ภาษาไทย เมื่อติดตั้งเสร็จแล้ว ต้องมีการเซตให้ตัวคอมไพเลอร์เป็นคอมไพเลอร์ของ jdk1.2.2 ภาษาไทย เพื่อให้สามารถแสดงผลภาษาไทยได้ดังรูปต่อไปนี้



รูปที่ 3-22 โปรแกรม Jbuilder 3.0



รูปที่ 3-23 แสดงการเซตพรอพอตี้ (Propoties) ของโปรแกรม Jbuilder 3.0



รูปที่ 3-24 แสดงการเซตคอมไพเลอร์โปรแกรม Jbuilder 3.0 ให้เป็นของ JDK1.2.2 ภาษาไทย

ให้กดปุ่ม Define เพื่อเปลี่ยนคอมไพเลอร์ให้กับโปรแกรม จากนั้นเลือกคอมไพเลอร์ซึ่งมีชื่อไฟล์ว่า java.exe จากไดเร็กทอรี่ที่เก็บจาวาเวอร์ชั่นภาษาไทยไว้ โปรแกรมจะทำการเซต Class path และ Source path ให้เองโดยอัตโนมัติ เสร็จแล้วกดปุ่ม OK โปรแกรมจะทำการเซตค่าสภาพแวดล้อมต่างๆ ให้กับคอมไพเลอร์ตัวใหม่ซึ่งใช้เวลานานพอสมควร

บทที่ 4
ผลการทดสอบและปัญหาที่เกิดขึ้น

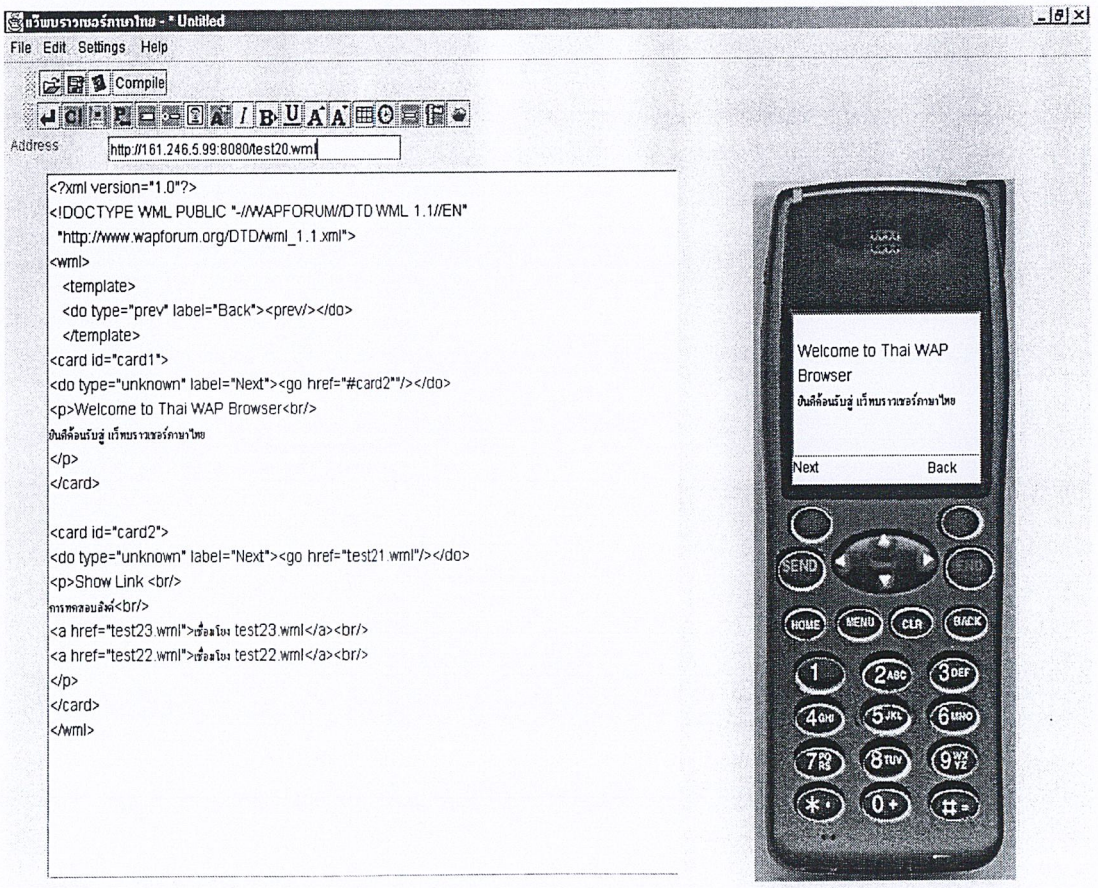
ในโครงงานแว็พบราวเซอร์ภาษาไทยนี้มีวัตถุประสงค์เพื่อต้องการให้แว็พบราวเซอร์นั้นสามารถแสดงผลเป็นภาษาไทยได้โดยได้รับมาจากผู้ใช้ 2 ทางคือจาก URL ที่ผู้ใช้ต้องการจะเข้าไปชมแว็พไซต์และจากการเขียนโปรแกรมของผู้ใช้ดังนั้นผลการทดสอบจึงได้แบ่งเป็น 2 ส่วนหลักคือ

4.1 ผลการทดสอบจากการที่ผู้ใช้เข้าไปชมแว็พไซต์ตาม URL ที่ต้องการ

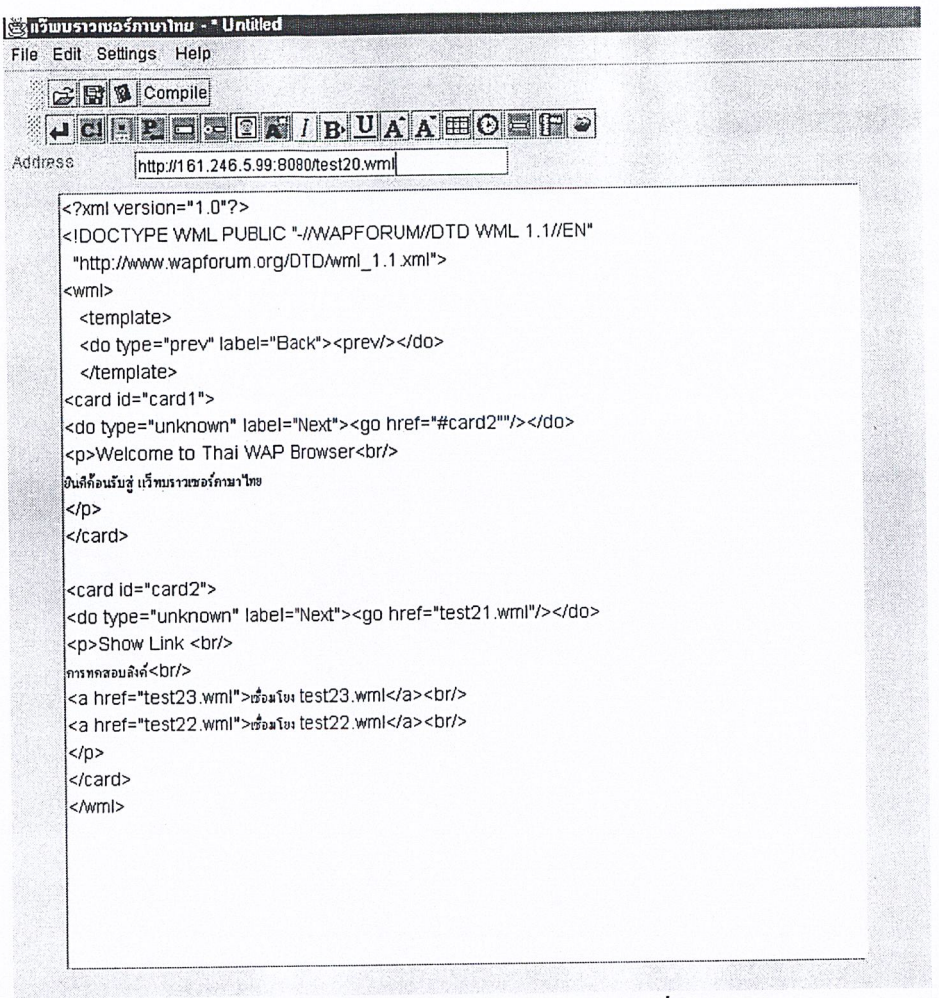
ในการทดสอบจะใส่ URL ที่ต้องการซึ่งผลการทดสอบจะถูกแสดงออกมาเป็น 2 ส่วนคือส่วนที่แสดงผลในส่วนแสดงผลไฟล์เอกสาร WML และส่วนโทรศัพท์มือถือจำลองซึ่งผลการทดลองเป็นดังนี้

4.1.1 ผลการทดสอบในส่วนแสดงผลไฟล์เอกสาร WML

ในการทดสอบนั้นได้ผลการทดลองคือแว็พบราวเซอร์ภาษาไทยสามารถแสดงผลเป็นภาษาไทยได้ในส่วนที่ใช้แสดงผลไฟล์เอกสาร WML ดังแสดงในรูปที่ 4-1และ รูปที่ 4-2



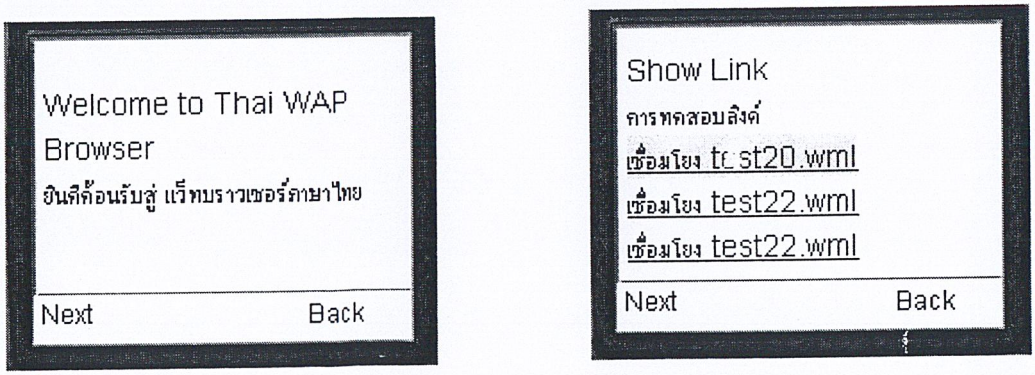
รูปที่ 4-1 ผลลัพธ์หลังจากที่ผู้ใช้ใส่ URL ตามที่ต้องการ



รูปที่ 4-2 รูปขยายการแสดงผลไฟล์เอกสาร WML ที่ได้จาก URL

4.1.2 ผลการทดสอบการแสดงผลบนหน้าจอโทรศัพท์มือถือจำลอง

จากการทดสอบนั้นผลที่ได้คือแว็พบราวเซอร์ภาษาไทยสามารถแสดงผลเป็นภาษาไทยได้บนหน้าจอของโทรศัพท์มือถือจำลองซึ่งได้แสดงไว้ในรูปที่ 4-3 และเนื่องจากแว็พเซิร์ฟเวอร์ปัจจุบันไม่มีการสนับสนุนข้อมูลที่เป็นภาษาไทยโครงงานนี้จึงได้สร้างเซิร์ฟเวอร์ขึ้นมาเอง



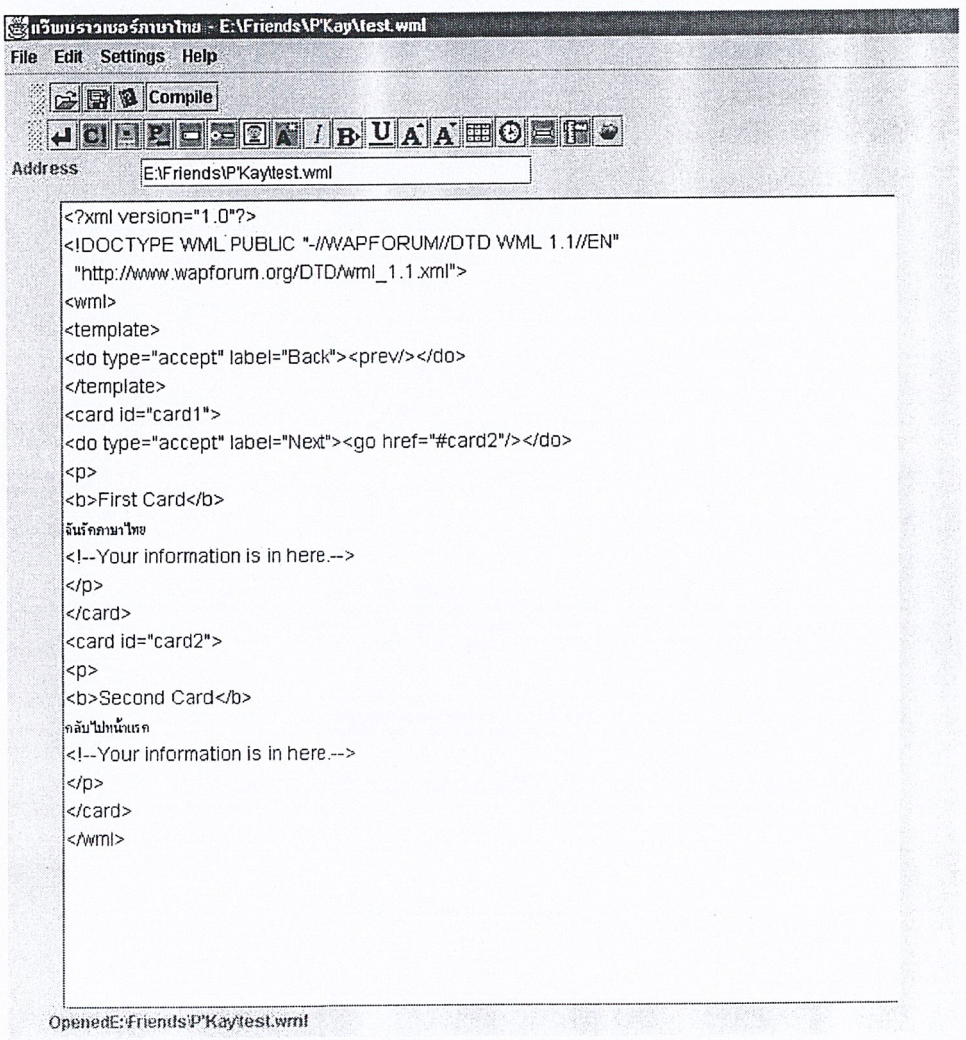
รูปที่4-3 ผลลัพธ์จากหน้าจอแสดงผลของไฟล์ WMLที่ได้จาก URL

4.2 ผลการทดสอบจากการเขียนโปรแกรมของผู้ใช้

ในการเขียนโปรแกรมภาษา WML ด้วยโปรแกรมแว็พบราวเซอร์ภาษาไทย ผู้ใช้จะเขียนลงไปในส่วนแสดงผลไฟล์เอกสาร WML และทำการคอมไพล์ ผลลัพธ์จะถูกแสดงผลบนหน้าจอโทรศัพท์มือถือจำลองซึ่งผลการทดสอบเป็นดังนี้

4.2.1 ผลการทดสอบการเขียนโปรแกรมภาษา WML

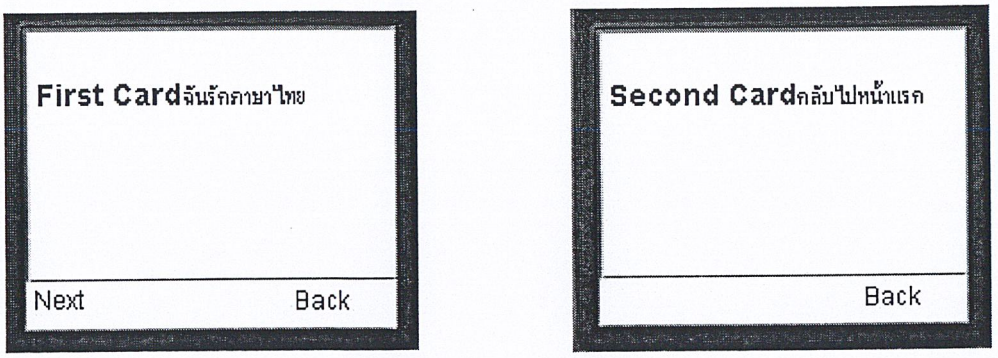
ในการทดสอบนี้ได้ทดลองโดยการพิมพ์ตัวอักษรภาษาไทยลงไปซึ่งผลที่ได้นั้น คือสามารถแสดงผลเป็นภาษาไทยได้ดังรูปที่ 4-4



รูปที่ 4-4 แสดงไฟล์ WML ที่ผู้ใช้เขียนโปรแกรมขึ้นมา

4.2.2 ผลการทดสอบในโทรศัพท์มือถือจำลอง

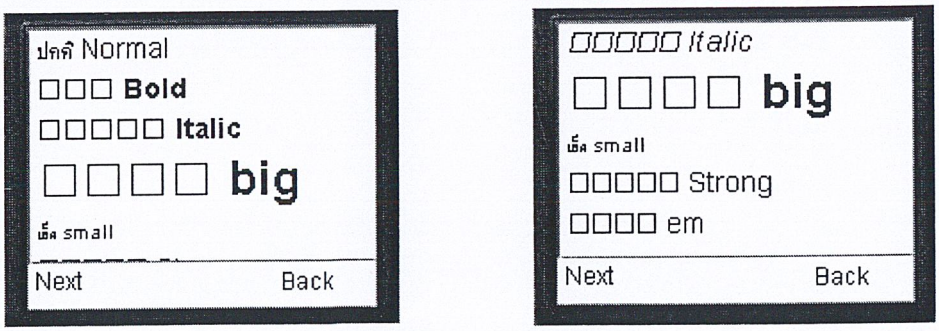
เมื่อผู้ใช้เขียนโปรแกรมเสร็จจะต้องคอมไพล์ ซึ่งการทดสอบคือได้เขียนโปรแกรมเป็นภาษาไทย แล้วคอมไพล์ผลการทดสอบคือส่วนโทรศัพท์มือถือจำลองสามารถแสดงผลเป็นภาษาไทยได้ดังรูปที่ 4-5



รูปที่ 4-5 ผลลัพธ์ที่ได้หลังจากการคอมไพล์โปรแกรมที่ผู้ใช้ทดลองเขียนขึ้นมา

4.3 ปัญหาที่เกิดขึ้นในการแสดงตัวอักษรภาษาไทยแบบต่างๆ ในแว็พบราวเซอร์ภาษาไทย

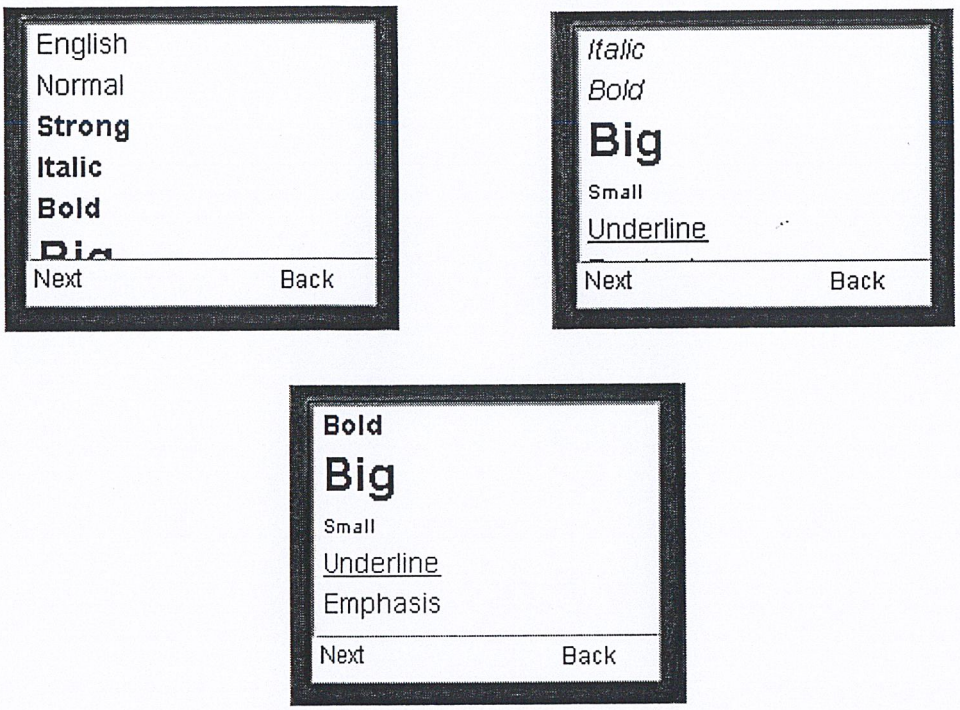
ปัญหาที่เกิดขึ้นเกี่ยวกับการแสดงผลตัวอักษรภาษาไทยของโปรแกรมแว็พบราวเซอร์ภาษาไทย นั้นก็คือการแสดงผลบนจอบราวเซอร์ของโทรศัพท์มือถือจำลองสำหรับบางแท็กที่เกี่ยวกับการกำหนดรูปแบบอักษรภาษาไทยซึ่งแท็กที่เกิดปัญหาได้แก่ แท็กอักษรตัวหนา (<b/>), แท็กอักษรตัวเอียง (<i/>), แท็กอักษรตัวเข้ม (), แท็กอักษรตัวใหญ่ (<big>), แท็ก emphasis ()โดยสามารถรองรับได้เพียง 2 แท็กคือแท็กอักษรตัวเล็ก (<small/>), และตัวอักษรปกติ ปัญหาเหล่านี้เนื่องมาจากคอมไพเลอร์ภาษาจาวาเวอร์ชัน 1.2.2 นี้ยังไม่สมบูรณ์นัก ซึ่งคาดว่าเวอร์ชัน 1.4 ที่กำลังจะออกสู่ตลาดน่าจะมีความสามารถมากกว่านี้สำหรับการแสดงผลภาษาไทย



รูปที่ 4-6 ปัญหาที่เกิดขึ้นในการแสดงผลภาษาไทยจากการกำหนดรูปแบบตัวอักษรแบบต่างๆ

4.4 ปัญหาที่เกิดขึ้นในการแสดงตัวอักษรภาษาอังกฤษแบบต่างๆ ในแว็พบราวเซอร์ภาษาไทย

ในการกำหนดรูปแบบตัวอักษรที่เป็นตัวอักษรภาษาอังกฤษนั้น สำหรับบางแท็กไม่สามารถแสดงได้ตามรูปแบบที่กำหนดไว้จากมาตรฐานที่กำหนดไว้ในภาษา WML ซึ่งมีทั้งหมด 7 แท็ก ในส่วนที่แสดงผลออกมาไม่ถูกต้องนั้นมีจำนวน 1 แท็กคือแท็ก emphasis ซึ่งใช้ในการเน้นคำ



รูปที่ 4-7 ปัญหาที่เกิดขึ้นในการแสดงผลภาษาอังกฤษจากการกำหนดรูปแบบตัวอักษรแบบต่างๆ

บทที่ 5
บทวิเคราะห์และสรุป

โครงงานนี้เป็นโครงงานที่สร้างโปรแกรมแว็พบราวเซอร์ภาษาไทยเพื่อให้สามารถแสดงผลเป็นภาษาไทยได้โดยผู้ใช้สามารถใช้แว็พบราวเซอร์นี้ในการเข้าชมแว็พไซต์ต่างๆ ได้ อีกทั้งยังสามารถเขียนโปรแกรมภาษา WML และดูผลการเขียนโปรแกรมได้จากส่วนโทรศัพท์มือถือจำลองได้ แต่โครงงานนี้ยังมีข้อจำกัดบางอย่างในการรองรับการแสดงผลบนโทรศัพท์มือถือจำลองของตัวภาษาไทยในการจัดรูปแบบของตัวอักษรและยังมีแท็กบางแท็กที่ยังไม่สามารถแสดงได้

5.1 สรุปและวิเคราะห์การทดสอบ

โปรแกรมแว็พบราวเซอร์ภาษาไทยที่สร้างขึ้นจากโครงงานนี้ยังไม่สมบูรณ์ในหลายๆ ด้านเช่น การแสดงผลและการจำลองการทำงานของโทรศัพท์มือถือ โดยทำการวิเคราะห์ผลการทดสอบและเปรียบเทียบกับแว็พบราวเซอร์อื่นๆทั่วไป สามารถแยกผลการทดสอบออกเป็นส่วนๆ ดังนี้

5.1.1 การทดสอบการเข้าแว็พไซต์ต่างๆ สามารถเข้าชมแว็พไซต์ต่างๆ ได้ แต่บางแว็พอาจแสดงผลไม่ถูกต้องเนื่องการเขียนแว็พไซต์ของที่ต่างๆ อาจเขียนไม่ถูกต้องตามหลักของภาษา WML ที่แท้จริง จากการทดสอบในแว็พไซต์ที่ดีก็สามารถเข้าชมได้และแสดงผลได้ดีพอสมควร

5.1.2 การทดสอบการเขียนโปรแกรม WML แล้วแสดงผลออกสู่หน้าจอแสดงผล สามารถแสดงผลได้ตามวัตถุประสงค์ โดยให้ผู้ใช้เขียนโปรแกรมภาษา WML เสร็จแล้ว save ลงไฟล์ จากนั้นกดปุ่ม Compile ผลลัพธ์จะแสดงให้เห็นบนหน้าจอแสดงผล

5.1.3 การทดสอบการแสดงผลข้อความต่างๆ สามารถแสดงผลได้ทั้งภาษาอังกฤษและภาษาไทย เมื่อเปรียบเทียบกับบราวเซอร์อื่นๆ ซึ่งแสดงผลได้เพียงแค่ภาษาอังกฤษเท่านั้น เพียงแต่การแสดงผลภาษาไทยนั้นยังไม่สมบูรณ์ดีพอ สามารถแสดงได้เพียงตัวอักษรปกติและตัวเล็กด้วยการกำหนดรูปแบบโดยแท็ก <small>...</small> แต่แท็กอื่นๆ นั้นไม่สามารถแสดงผลได้ถูกต้องเช่น <big>..</big>, .. และ .. เป็นต้น

5.1.4 การทดสอบการแสดงผลการลิงค์ไปยังการ์ดอื่นๆ และการลิงค์ไปยังแว็พเพจต่างๆ สามารถทำงานได้ถูกต้องตามรูปแบบการแสดงผลไฟล์ WML

5.1.5 การทดสอบการแสดงผลรูปภาพ สามารถแสดงรูปภาพไฟล์นามสกุล .bmp .gif .jpg ได้แต่ไม่สามารถแสดงผล .wbmp ได้ สามารถแสดงผลได้ออกมาเป็นภาพสีได้

5.1.6 การทดสอบการรับข้อมูลอินพุต สามารถรับอินพุตได้แต่ไม่สามารถรับส่งข้อมูลแบบ cgi ได้ เนื่องจากการแสดงผลแบบนี้สำหรับ WML นั้นค่อนข้างซับซ้อนและยุ่งยากต่อการจัดการให้แสดงผลได้ถูกต้อง

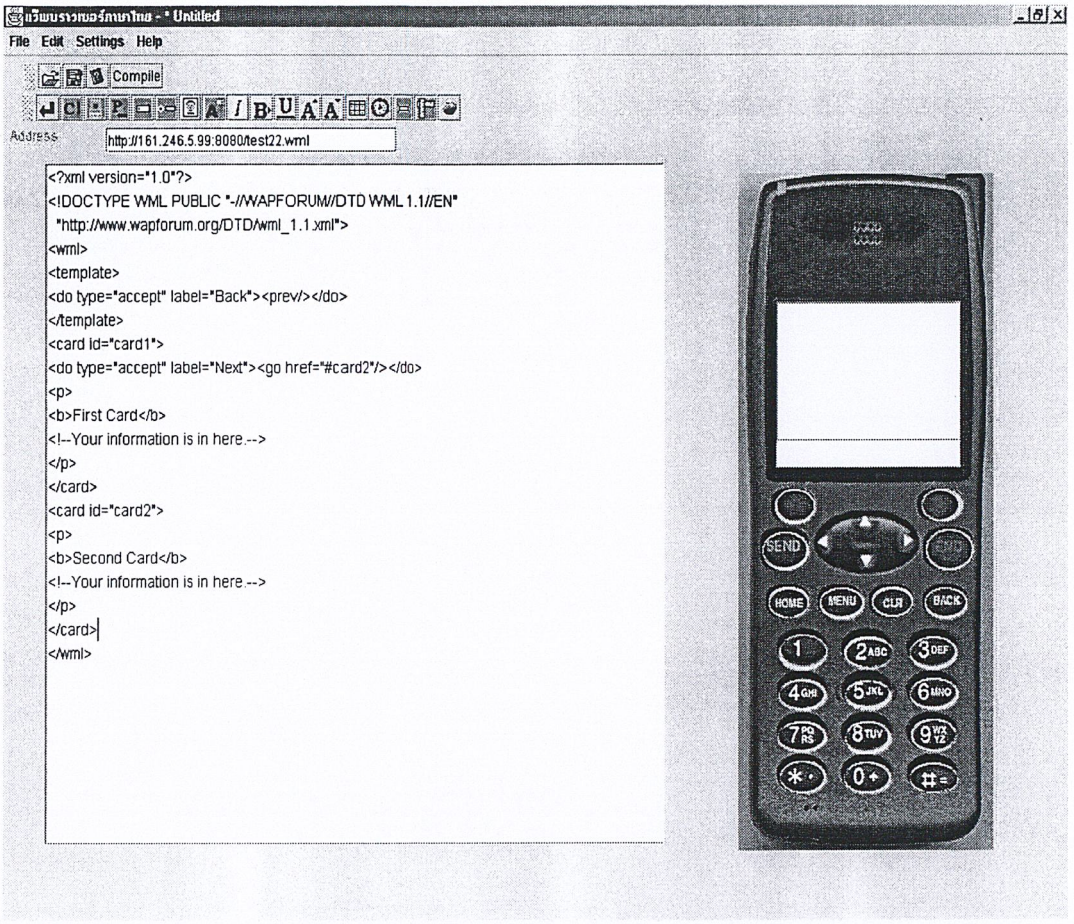
5.2 แนวทางการพัฒนา

5.2.1 ในส่วนของการพัฒนาให้แว็พบราวเซอร์ภาษาไทยสามารถสนับสนุนการรับการแสดงผลจากส่วน WML Script และการรับ CGIได้ทั้งสองแบบคือ GET และ POST ได้สมบูรณ์แบบมากขึ้น และสนับสนุนได้ทุกแท็กของภาษา WML

5.2.2 ในส่วนของการพัฒนาให้แว็พบราวเซอร์ภาษาไทยมีความฉลาดขึ้น (Intelligent) เช่น สามารถรับสคริปต์จากผู้ใช้เพื่อสามารถกำหนดการทำงานให้กับตัวบราวเซอร์โดยอัตโนมัติ

ภาคผนวก ก.

ผลลัพธ์บางส่วนที่ได้จากการทดสอบโปรแกรมแว็พบราวเซอร์ภาษาไทย โดยเขียนโปรแกรมภาษา WML ขึ้นมาเพื่อทดสอบแท็กต่างๆ ที่สำคัญสำหรับการแสดงผลออกสู่หน้าจอของมือถือจำลอง



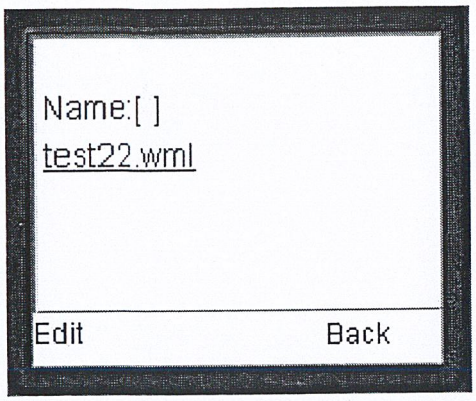
รูปที่ ก-1 ตัวอย่างโปรแกรมแว็พบราวเซอร์ภาษาไทย

รูปที่ ก-2 รูปขยายหน้าจอการแสดงผลข้อความ

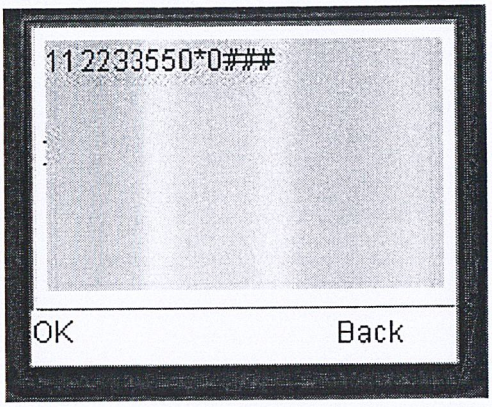
รูปที่ ก-3 รูปขยายหน้าจอการแสดงผลลิงค์

รูปที่ ก-4 รูปขยายหน้าจอการแสดงผลรูปภาพ

รูปที่ ก-5 รูปขยายการแสดงผลการกำหนดรูปแบบตัวอักษร



รูปที่ ก-6 รูปขยายหน้าจอการแสดงผลการรับอินพุต



รูปที่ ก-7 รูปขยายหน้าจอการแสดงผลการป้อนอินพุต

ภาคผนวก ข.

ซอร์สโค้ด (Source Code) ส่วนที่เป็นพาร์เซอร์ (Parser) ของโปรแกรมแว็พบราวเซอร์ภาษาไทย ซึ่งเป็นส่วนหลักที่สำคัญมากในการทำงานเพื่อแสดงผลไฟล์เอกสาร WML โดยสร้างเป็นฟังก์ชันสำหรับให้โปรแกรมเรียกใช้งาน

```
public void Parser(String urls) {
    try {
        i=0;error=0;check_close_tag=0;no_card=-1;card=0;open_tag=0;
        hist=-1;count_char=0;
        tag_wml=0;tag_head=0;
        tag_template=0;tag_td=0;tag_tr=0;tag_table=0;
        tag_card=0;
        tag_p=0;tag_postfield=0;tag_prev=0;
        tag_a=0;tag_anchor=0;has_tag_a=0;
        tag_b=0;tag_big=0;tag_br=0;
        tag_do=0;
        tag_em=0;
        tag_fieldset=0;
        tag_go=0;
        tag_i=0;tag_img=0;
        tag_refresh=0;
        tag_onevent=0;tag_optgroup=0;tag_option=0;
        tag_select=0;tag_small=0;tag_strong=0;
        tag_u=0;special=0;set_back=0;set_prev=0;intp=0;
        Temp_str="";Str_www="";host="";count_link=-1;count_down=-1;
        count_input=-1;load_pic=0;
        int j,k;
        for (i=0;i<10;i++){
            Card[i][0]="#";
            Card[i][1]="";
            Card[i][2]="";
            Input_value[i][0]="";
            Input_value[i][1]="";
```

```
    Input_value[i][2]="";
    Input_value[i][3]="";
    Input_value[i][4]="";
    Input_value[i][5]="";
    Input_value[i][6]="";
    Input_value[i][7]="";
    Input_value[i][8]="";
    Input_value[i][9]="";
    for (j=0;j<20;j++){
        Link_pos[i][j][0]=0;
        Link_pos[i][j][1]=0;
        Link_pos[i][j][2]=0;
        Link_pos[i][j][3]=0;
        Link_a[i][j]="";
} }
String Str_pic="";
jTextArea1.setText("");
left.setText("");
right.setText("");
jTextArea4.setText("");// 22
num_card=0;
num_card1=0;
num_link=0;
num_link1=0;
FileLink[0] = new FileOutputStream("C:/temph/tmp0.html");
FileLink[1] = new FileOutputStream("C:/temph/tmp1.html");
FileLink[2] = new FileOutputStream("C:/temph/tmp2.html");
FileLink[3] = new FileOutputStream("C:/temph/tmp3.html");
FileLink[4] = new FileOutputStream("C:/temph/tmp4.html");
FileLink[5] = new FileOutputStream("C:/temph/tmp5.html");
FileLink[6] = new FileOutputStream("C:/temph/tmp6.html");
FileLink[7] = new FileOutputStream("C:/temph/tmp7.html");
FileLink[8] = new FileOutputStream("C:/temph/tmp8.html");
FileLink[9] = new FileOutputStream("C:/temph/tmp9.html");
```

```
int a;
a=urls.lastIndexOf("/");
if((urls.indexOf("/"))!=(a-1)){
    host=urls.substring(0,urls.lastIndexOf("/")+1);
    System.out.println(host+"host");
}
else{
    host=urls;
    System.out.println(host+"host");
    urls=urls+"/index.wml";
}
URL url = new URL(urls);
jTextField1.setText(urls);
InputStream in = url.openStream();
DataInputStream dis = new DataInputStream(url.openConnection().getInputStream());
Reader reader = new InputStreamReader (in);
BufferedReader bufferedReader = new BufferedReader(reader);
String line;
jTextArea1.setFont(new Font("dialog",Font.PLAIN,14));
while ((line = bufferedReader.readLine()) != null)
 jTextArea1.append(line+"\n");
bufferedReader.close();
int s,l=0;
while ((s = dis.read()) != -1){
special=0;
switch(s) {
    case '<' : {
        open_tag=1;
        break;
    }
    case '/' :  {  //Check close tag
        if (open_tag ==1){
            check_close_tag = 1;
            s = dis.read();
```

```
if (s == 'a') {  //close tag a....
    s = dis.read();
    if (s=='>'){    //close tag </a>
        tag_a--;
        if(count_link!=-1)
            Link_pos[no_card][count_link][1]=count_char+5;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('a');
        FileLink[no_card].write('>');
    }
    else if (s == 'n') { // close tag </anchor>
        tag_anchor--;
        while((s=dis.read())!='>') ;
    }  }
    else if (s == 'b') {          //close tag b....
        s=dis.read();
        if (s == '>') {      //close tag </b>
            tag_b--;
            FileLink[no_card].write('<');
            FileLink[no_card].write('/');
            FileLink[no_card].write('b');
            FileLink[no_card].write('>');
        }
        else if (s == 'r') { //close tag </br>
            tag_br--;
            while((s=dis.read())!='>') ;
        }
        else if (s == 'i') { //close tag </big>
            tag_big--;
            FileLink[no_card].write('<');
            FileLink[no_card].write('/');
            FileLink[no_card].write('b');
            FileLink[no_card].write('i');
```

```
                FileLink[no_card].write('g');
                FileLink[no_card].write('>');
                while((s=dis.read())!='>');
        }   }
        else if (s == 'd') {          //close tag </do>
        tag_do--;
        while ((s = dis.read())!='>');
    }
    else if (s == 'e') {      //close tag </em>
        tag_em--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('e');
        FileLink[no_card].write('m');
        FileLink[no_card].write('>');
        while ((s = dis.read())!='>');
    }
    else if (s == 'f'){       //close tag </fieldset>
        tag_fieldset--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('f');
        FileLink[no_card].write('i');
        FileLink[no_card].write('e');
        FileLink[no_card].write('l');
        FileLink[no_card].write('d');
        FileLink[no_card].write('s');
        FileLink[no_card].write('e');
        FileLink[no_card].write('t');
        FileLink[no_card].write('>');
        while ((s = dis.read())!='>');
    }
    else if (s  == 'g') {          // close tag </go>
        while ((s = dis.read())!='>');
```

```
    tag_go--;
}
else if (s == 'h') {      //close tag </head>
    while ((s = dis.read())!='>');
    tag_head--;
    for (i=0;i<10;i++) {
        FileLink[i].write('<');
        FileLink[i].write('/');
        FileLink[i].write('h');
        FileLink[i].write('e');
        FileLink[i].write('a');
        FileLink[i].write('d');
        FileLink[i].write('>');
} }
else if (s == 'i') {      //close tag i...
    s = dis.read();
    if (s == '>') {              // close tag </i>
        tag_i--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('i');
        FileLink[no_card].write('>');
} }
else if (s == 'o')  {  //close tag o....
    s = dis.read();
    if (s =='n') {  // close tag </onevent>
        tag_onevent--;
        while ((s= dis.read())!='>');
    }
    else if (s == 'p'){
        while ((s= dis.read())!='t');
        s=dis.read();
        if (s=='g') {   //close tag </optgroup>
            tag_optgroup--;
```

```
            while ((s= dis.read())!='>');
        }
        else if (s == 'i') {       // close tag </option>
            tag_option--;
            while ((s= dis.read())!='>') ;
} } }
else if (s == 'p') {      //close tag </p>
    s = dis.read();
    if (s == '>'){
        tag_p=0;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('p');
        FileLink[no_card].write('>');
    }
    else if (s == 'r') {       //close tag </prev>
        tag_prev--;
        while((s=dis.read())!='>');
    }
    else if (s=='o') {        //close tag </postfield>
        tag_postfield--;
        while((s=dis.read())!='>');
} }
else if (s == 'r') {       // close tag </refresh>
    tag_refresh--;
    while((s=dis.read())!='>');
}
else if (s == 's') {      //close tag s...
    s = dis.read();
    if (s == 'm') {      //close tag </small>
        tag_small--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('s');
```

```
            FileLink[no_card].write('m');
            FileLink[no_card].write('a');
            FileLink[no_card].write('l');
            FileLink[no_card].write('l');
            FileLink[no_card].write('>');
            while((s=dis.read())!='>');
        }
        else if (s == 't') {        // close tag </strong>
            tag_strong--;
            FileLink[no_card].write('<');
            FileLink[no_card].write('/');
            FileLink[no_card].write('b');
            FileLink[no_card].write('>');
            while((s=dis.read())!='>');
        }
        else if (s == 'e'){         //close tag </select>
            tag_select--;
            while((s=dis.read())!='>');
    }   }
    else if (s=='c'){               // close tag card
        while((s=dis.read())!='>');
        tag_card--;
        for(i=0;i<20;i++){
            Link_pos[no_card][i][2]=count_link;
        }
        if (count_link!=-1)
        System.out.println(Link_pos[no_card][count_link][2]+"links1");
        System.out.println(count_link+"links2");
    }
    else if (s == 't'){         //close tag t...
        s = dis.read();
        if (s == 'd') {         //close tag </td>
            tag_td--;
            FileLink[no_card].write('<');
```

```
        FileLink[no_card].write('/');
        FileLink[no_card].write('t');
        FileLink[no_card].write('d');
        FileLink[no_card].write('>');
        s = dis.read();
    }
    else if (s == 'r') {      //close tag </tr>
        tag_tr--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('t');
        FileLink[no_card].write('r');
        FileLink[no_card].write('>');
        s=dis.read();
    }
    else if (s == 'a') {      // close tag </table>
        tag_table--;
        FileLink[no_card].write('<');
        FileLink[no_card].write('/');
        FileLink[no_card].write('t');
        FileLink[no_card].write('a');
        FileLink[no_card].write('b');
        FileLink[no_card].write('l');
        FileLink[no_card].write('e');
        FileLink[no_card].write('>');
        while((s=dis.read())!='>');
    }
    else if (s == 'e') {          // close tag </template>
        tag_template--;
        while ((s= dis.read())!='>') ;
} }
else if (s == 'u'){
    tag_u--;
    FileLink[no_card].write('<');
```

```
            FileLink[no_card].write('/');
            FileLink[no_card].write('u');
            FileLink[no_card].write('>');
            while((s=dis.read())!='>');
        }
        else if (s == 'w') {     // close tag </wml>
            tag_wml--;
            for (i=0;i<10;i++){
                FileLink[i].write('<');
                FileLink[i].write('/');
                FileLink[i].write('b');
                FileLink[i].write('o');
                FileLink[i].write('d');
                FileLink[i].write('y');
                FileLink[i].write('>');
                FileLink[i].write('<');
                FileLink[i].write('/');
                FileLink[i].write('h');
                FileLink[i].write('t');
                FileLink[i].write('m');
                FileLink[i].write('l');
                FileLink[i].write('>');
            }
            while((s=dis.read())!='>');
    }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[nc_card].write(s);
    }
    open_tag=0;
    break;
}
case '!' : {
    if (open_tag ==1){
```

```
            while ((s = dis.read()) != '>') ;
            if (s == '>'){
                error=0;
            }
            else error=1;
        }
        if ((tag_p!=0)&&(open_tag!=1)){
            count_char++;
            FileLink[no_card].write(s);
        }
        open_tag=0;
        break;
    }
    case '?' :  {
        if (open_tag ==1){
            while ((s=dis.read())!='>') ;
            if (s == '>'){
                error=0;
            }
            else error=1;
        }
        if ((tag_p!=0)&&(open_tag!=1)){
            count_char++;
            FileLink[no_card].write(s);
        }
        open_tag=0;
        break;
    }
    case 'W' ://tag <wml>
    case 'w' :{
        if (open_tag ==1){
            s = dis.read();
            if (s == 'm'){
                s = dis.read();
```

```
            if (s == 'l'){
                s = dis.read();
                if (s == '>'){
                    tag_wml++;
                    for (i=0;i<10;i++){
                        FileLink[i].write('<');
                        FileLink[i].write('h');
                        FileLink[i].write('t');
                        FileLink[i].write('m');
                        FileLink[i].write('l');
                        FileLink[i].write('>');
    }   }   }   }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'H' :  //tag <head>
case 'h' :  {
    if (open_tag ==1){
        s = dis.read();
        if (s == 'e'){
            s = dis.read();
            if (s == 'a'){
                s = dis.read();
                if (s == 'd'){
                    s = dis.read();
                    if (s == '>'){
                        tag_head++;
                        for (i=0;i<10;i++){
                            FileLink[i].write('<');
                            FileLink[i].write('h');
```

```
                    FileLink[i].write('e');
                    FileLink[i].write('a');
                    FileLink[i].write('d');
                    FileLink[i].write('>');
    } } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'T' :  //tag <template>
case 't' :  {
    if (open_tag ==1){
        s = dis.read();
        if (s == 'e'){
            s = dis.read();
            if (s == 'm'){
                s = dis.read();
                if (s == 'p'){
                    s = dis.read();
                    if (s == 'l'){
                        s = dis.read();
                        if (s == 'a'){
                            s = dis.read();
                            if (s == 't'){
                                s = dis.read();
                                if (s == 'e'){
                                    s = dis.read();
                                    if (s == '>'){
                                        tag_template++;
                                    }
                                    else if(s==' ')  {
```

```
                                tag_template++;
                                while((s = dis.read())!='>') ;
} } } } } } } }
else if (s == 'd') {          // tag <td>
    s = dis.read();
    if (s == '>'){
        tag_td++;
        FileLink[no_card].write('<');
        FileLink[no_card].write('t');
        FileLink[no_card].write('d');
        FileLink[no_card].write('>');
} }
else if (s == 'r') {          //tag <tr>
    s = dis.read();
    if (s == '>'){
        tag_tr++;
        FileLink[no_card].write('<');
        FileLink[no_card].write('t');
        FileLink[no_card].write('r');
        FileLink[no_card].write('>');
} }
else if (s == 'a')   {   //tag <table>
    s = dis.read();
    if (s == 'b'){
        s = dis.read();
        if (s == 'l'){
            s = dis.read();
            if (s == 'e'){
                s = dis.read();
                if (s == '>'){
                    tag_table++;
                    System.out.println("<table>");
                    FileLink[no_card].write('<');
                    FileLink[no_card].write('t');
```

```
                FileLink[no_card].write('a');
                FileLink[no_card].write('b');
                FileLink[no_card].write('l');
                FileLink[no_card].write('e');
                FileLink[no_card].write('>');
            }
            else if (s == ' '){
                tag_table++;
                FileLink[no_card].write('<');
                FileLink[no_card].write('t');
                FileLink[no_card].write('a');
                FileLink[no_card].write('b');
                FileLink[no_card].write('l');
                FileLink[no_card].write('e');
                FileLink[no_card].write(' ');
                while ((s=dis.read())!='>'){
                    if (s == 'c'){
                    FileLink[no_card].write(s);
                    s=dis.read();
                        if (s == 'o'){
                            FileLink[no_card].write(s);
                            FileLink[no_card].write('l');
                            FileLink[no_card].write('s');
                            while((s=dis.read())!="");
                            FileLink[no_card].write('=');
                            FileLink[no_card].write("");
                            while((s=dis.read())!=""){
                                FileLink[no_card].write(s);
                    } } }
                    FileLink[no_card].write(s);
                }
                FileLink[no_card].write('>');
                FileLink[no_card].write('\n');
} } } } }
```

```
        else if (s == 'i') {            //tg <timer>
            s = dis.read();
            if (s == 'm'){
                s = dis.read();
                if (s == 'e'){
                    s = dis.read();
                    if (s == 'r'){
                        s = dis.read();
                        if (s == ' '){
                            while ((dis.read())!='>');
    }   }   }   }   }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'C' :  // tag <card>
case 'c' :  {
    if (open_tag ==1){
        s = dis.read();
        if (s == 'a'){
            s = dis.read();
            if (s == 'r'){
                s = dis.read();
                if (s == 'd'){
                    no_card++;
                    count_char=-1;
                    count_link=-1;
                    count_input=-1;
                    while((s=dis.read())!='>'){
                        if (s==' '){
                            s=dis.read();
```

```
                        if(s=='i'){
                            s=dis.read();
                            if(s=='d'){
                                while((s=dis.read())!='');
                                while((s=dis.read())!=''){
                                    Card[no_card][0]=Card[no_card][0]+((char)s);
  } } } } } } } } }
  if (s == '>') {
      FileLink[no_card].write('<');
      FileLink[no_card].write('b');
      FileLink[no_card].write('o');
      FileLink[no_card].write('d');
      FileLink[no_card].write('y');
      FileLink[no_card].write('>');
      tag_card++;
      num_card++;
      num_card1++;
  }
  if ((tag_p!=0)&&(open_tag!=1)){
      count_char++;
      FileLink[no_card].write(s);
  }
  open_tag=0;
  break;
}
case 'P' :  // tag <p>
case 'p' :  {
   if (open_tag ==1){
       s = dis.read();
       if (s == '>'){
           tag_p++;
           FileLink[no_card].write('<');
           FileLink[no_card].write('p');
           FileLink[no_card].write('>');
```

```
}
else if (s == ' ') {      //tag <p ...>
    tag_p++;
    FileLink[no_card].write('<');
    FileLink[no_card].write('p');
    FileLink[no_card].write(' ');
    s = dis.read();
    FileLink[no_card].write(s);
    while ((s=dis.read())!='>'){
        FileLink[no_card].write(s);
    }
    FileLink[no_card].write('>');
    FileLink[no_card].write('\n');
}
else if (s == 'r')   {       // tag <prev/>
    s = dis.read();
    if (s == 'e'){
        s = dis.read();
        if (s == 'v'){
            s = dis.read();
            if (s == '>'){
                tag_prev++;
                if(tag_do!=0){
                    set_back=1;
                    right.setText("   "+Label_do);
            }   }
            else if (s == '/'){
                s = dis.read();
                if (s == '>'){
                    if (tag_do!=0){
                        set_back=1;
                        right.setText("   "+Label_do);
}   }   }   }   }   }
else if (s == 'o')   {       // tag <postfield>
```

```
                s = dis.read();
                if (s == 's'){
                    s = dis.read();
                    if (s == 't'){
                        s = dis.read();
                        if (s == 'f'){
                            s = dis.read();
                            if (s == 'i'){
                                s = dis.read();
                                if (s == 'e'){
                                    s = dis.read();
                                    if (s == 'l'){
                                        s = dis.read();
                                        if (s == 'd'){
                                            s = dis.read();
                                            if (s == ' '){
                                                tag_postfield++;
                                                while((s=dis.read())!='>');
    } } } } } } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'A' :
case 'a' :  {                   ////tag <access>
    if (open_tag ==1){
        s = dis.read();
        if (s == 'c'){
            s = dis.read();
            if (s == 'c'){
                s = dis.read();
```

```
            if (s == 'e'){
                s = dis.read();
                if (s == 's'){
                    s = dis.read();
                    if (s == 's'){
                        while ((dis.read())!='>');
} } } } }
else if (s == 'n'){          // tag <anchor>
    s = dis.read();
    if (s == 'c'){
        s = dis.read();
        if (s == 'h'){
            s = dis.read();
            if (s == 'o'){
                s = dis.read();
                if (s == 'r'){
                    while((s=dis.read())!='>');
                    tag_anchor++;
} } } } }
else if (s ==' '){                ///tag <a href>
    tag_a++;
    has_tag_a++;
    count_link++;
    Link_pos[no_card][count_link][3]=1;
    FileLink[no_card].write('<');
    FileLink[no_card].write('a');
    FileLink[no_card].write(' ');
    while ((s=dis.read())!='>'){
        if (s=='t'){
            FileLink[no_card].write(s);
            while ((s=dis.read())!=' ')FileLink[no_card].write(s);
        }
        else if(s=='h'){
            FileLink[no_card].write(s);
```

```
                    s=dis.read();
                    if(s=='r'){
                        FileLink[no_card].write(s);
                        s=dis.read();
                        if(s=='e'){
                            FileLink[no_card].write(s);
                            s=dis.read();
                            if(s=='f'){
                                FileLink[no_card].write(s);
                                while ((s=dis.read())!="")
                                    FileLink[no_card].write(s);
                                FileLink[no_card].write(s);
                                while ((s=dis.read())!=""){
                                    FileLink[no_card].write(s);
                                    Link_a[no_card][count_link]=Link_a[no_card][count_link]+
                                    ((char)s);
                    }   }   }   }   }
                    FileLink[no_card].write(s);
                }
                FileLink[no_card].write('>');
                FileLink[no_card].write('\n');
                Link_pos[no_card][count_link][0]=count_char+5;
                System.out.println(Link_a[no_card][count_link]+"link name");
    }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'B' :  //tag <big>
case 'b' :  {
    if (open_tag == 1){
```

```
s = dis.read();
if (s == 'i'){
    s = dis.read();
    if (s == 'g'){
        s = dis.read();
        if (s == '>'){
            tag_big++;
            FileLink[no_card].write('<');
            FileLink[no_card].write('b');
            FileLink[no_card].write('i');
            FileLink[no_card].write('g');
            FileLink[no_card].write('>');
} } }
if (s == '>'){        // tag <b>
    tag_b++;
    FileLink[no_card].write('<');
    FileLink[no_card].write('b');
    FileLink[no_card].write('>');
}
else if (s == 'r'){           // tag <br>
    s = dis.read();
    if (s == '>'){
        tag_br++;
        count_char++;
        FileLink[no_card].write('<');
        FileLink[no_card].write('b');
        FileLink[no_card].write('r');
        FileLink[no_card].write('>');
    }
    else if (s == '/'){
        s = dis.read();
        if (s == '>'){
            count_char++;
            FileLink[no_card].write('<');
```

```
                    FileLink[no_card].write('b');
                    FileLink[no_card].write('r');
                    FileLink[no_card].write('>');
    } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag =0;
    break;
}
case 'D' :  //tag <do>
case 'd' :  {
    if (open_tag ==1) {
        s = dis.read();
        if (s == 'o'){
            tag_do++;
            Label_do=" ";
            while((s=dis.read())!='>'){
                if(s==' '){
                    s=dis.read();
                    if (s =='l'){
                        s=dis.read();
                        if(s=='a') {
                            s=dis.read();
                            if(s=='b') {
                                while((s=dis.read())!='"');
                                while((s=dis.read())!='"') {
                                    if(tag_card!=0)
                                        Card[no_card][2]=Card[no_card][2]+((char)s);
                                    else Label_do=Label_do+((char)s);
    } } } } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
```

```
            FileLink[no_card].write(s);
        }
        open_tag =0;
        break;
    }
    case 'E' :  // tag <em>
    case 'e' :  {
        if (open_tag == 1){
            s = dis.read();
            if (s == 'm'){
                s = dis.read();
                if (s == '>'){
                    tag_em++;
                    FileLink[no_card].write('<');
                    FileLink[no_card].write('e');
                    FileLink[no_card].write('m');
                    FileLink[no_card].write('>');
        }  }  }
        if ((tag_p!=0)&&(open_tag!=1)){
            count_char++;
            FileLink[no_card].write(s);
        }
        open_tag = 0;
        break;
    }
    case 'F' :  // tag <fieldset>
    case 'f' : {
        if(open_tag==1){
            if (s == 'i'){
                s = dis.read();
                if (s == 'e'){
                    s = dis.read();
                    if (s == 'l'){
                        s = dis.read();
```

```
                    if (s == 'd'){
                        s = dis.read();
                        if (s == 's'){
                            s = dis.read();
                            if (s == 'e'){
                                s = dis.read();
                                if (s == 't'){
                                    s = dis.read();
                                    if (s == ' '){
                                        tag_postfield++;
                                        FileLink[no_card].write('<');
                                        FileLink[no_card].write('f');
                                        FileLink[no_card].write('i');
                                        FileLink[no_card].write('e');
                                        FileLink[no_card].write('l');
                                        FileLink[no_card].write('d');
                                        FileLink[no_card].write('s');
                                        FileLink[no_card].write('e');
                                        FileLink[no_card].write('t');
                                        FileLink[no_card].write(' ');
                                        while ((s = dis.read())!='>'){
                                            FileLink[no_card].write(s);
                                        }
                                        FileLink[no_card].write('>');
                                        FileLink[no_card].write('\n');
    } } } } } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case 'G' :  //tag <go>
```

```
case 'g' :  {
    if (open_tag == 1){
        Href_go="";
        Method_go="";
        s = dis.read();
        if (s == 'o'){
            while((s = dis.read())!='>'){
                if(s==' '){
                    s = dis.read();
                    if (s == 'h'){
                        s = dis.read();
                        if (s == 'r'){
                            s = dis.read();
                            if (s == 'e'){
                                s = dis.read();
                                if (s == 'f')  {
                                    while((s = dis.read())!='');
                                    while((s = dis.read())!=''){
                                        Href_go=Href_go+((char)s);
                                        if(tag_card!=0)
                                            Card[no_card][1]=Card[no_card][1]+((char)s);
                    }   }   }   }   }
                    else if(s=='m'){
                        s=dis.read();
                        if(s=='e'){
                            s=dis.read();
                            if(s=='t'){
                                s=dis.read();
                                if(s=='h'){
                                    s=dis.read();
                                    if(s=='o'){
                                        s=dis.read();
                                        if(s=='d'){
                                            while((s = dis.read())!='');
```

```
                                        while((s = dis.read())!=''){
                                            Method_go=Method_go+((char)s);
                } } } } } } } } }
                tag_go++;
        } }
        if ((tag_p!=0)&&(open_tag!=1)){
            count_char++;
            FileLink[no_card].write(s);
        }
    open_tag =0;
    break;
}
case 'I' :  // tag <input>
case 'i' :  {
    if (open_tag == 1){
        s = dis.read();
        if (s == 'n'){
            s = dis.read();
            if (s == 'p'){
                s = dis.read();
                if (s == 'u'){
                    s = dis.read();
                    if (s == 't'){
                        count_link++;
                        count_input++;
                        Link_pos[no_card][count_link][3]=2;
                        FileLink[no_card].write('[');
                        count_char++;
                        Link_pos[no_card][count_link][0]=count_char+4;
                        FileLink[no_card].write(' ');
                        count_char++;
                        Link_pos[no_card][count_link][1]=count_char+4;
                        FileLink[no_card].write(']');
                        count_char++;
```

```
                while ((s=dis.read())!='>');
} } } }
else if (s == '>') {          //tag <i>
    tag_i++;
    FileLink[no_card].write('<');
    FileLink[no_card].write('i');
    FileLink[no_card].write('>');
}
else if (s == 'm') {          // tag <img>
    s = dis.read();
    if (s == 'g'){
        Str_pic="";
        load_pic=0;
        FileLink[no_card].write('<');
        FileLink[no_card].write('i');
        FileLink[no_card].write('m');
        FileLink[no_card].write('g');
        FileLink[no_card].write(' ');
        while ((s=dis.read())!='>') {
            if(s!='/')
                FileLink[no_card].write(s);
            if(s=='s'){
                s=dis.read();
                FileLink[no_card].write(s);
                if(s=='r'){
                    s=dis.read();
                    FileLink[no_card].write(s);
                    if(s=='c'){
                        while ((s=dis.read())!='"')FileLink[no_card].write(s);
                        FileLink[no_card].write(s);
                        while ((s=dis.read())!='"'){
                            FileLink[no_card].write(s);
                            Str_pic=Str_pic+((char)s);
                        }
```

```
                        FileLink[no_card].write(s);
                    } } } }
                    FileLink[no_card].write('>');
                }
                Load_Pics(Str_pic);
                tag_img++;
            } }
            if ((tag_p!=0)&&(open_tag!=1)){
                count_char++;
                FileLink[no_card].write(s);
            }
            open_tag =0;
            break;
        }
        case 'M' :  //tag <meta>
        case 'm' :  {
            if (open_tag == 1){
                s = dis.read();
                if (s == 'e'){
                    s = dis.read();
                    if (s == 't'){
                        s = dis.read();
                        if (s == 'a'){
                            s = dis.read();
                            if (s == ' '){
                                for (i=0;i<10;i++){
                                    FileLink[i].write('<');
                                    FileLink[i].write('m');
                                    FileLink[i].write('e');
                                    FileLink[i].write('t');
                                    FileLink[i].write('a');
                                    FileLink[i].write(' ');
                                }
                                while ((s=dis.read())!='>'){
```

```
                        for (i=0;i<10;i++)FileLink[i].write(s);
                    }
                    for (i=0;i<10;i++)FileLink[i].write('>');
    } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag =0;
    break;
}
case 'N' :  //tag <noop>
case 'n' :  {
    if (open_tag ==1){
        s = dis.read();
        if (s == 'o'){
            s = dis.read();
            if (s == 'o'){
                s = dis.read();
                if (s == 'p'){
                    s = dis.read();
                    if (s == '/'){
                        s = dis.read();
                        if (s=='>') ;
    } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag =0;
    break;
}
case 'R' :  // tag <refresh>
case 'r' :  {
```

```
if (open_tag ==1){
    s = dis.read();
    if (s == 'e'){
        s = dis.read();
        if (s == 'f'){
            s = dis.read();
            if (s == 'r'){
                s = dis.read();
                if (s == 'e'){
                    s = dis.read();
                    if (s == 's'){
                        s = dis.read();
                        if (s == 'h'){
                            s = dis.read();
                            if (s == '>'){
                                tag_refresh++;
                                if(tag_do!=0)
                                    show(card);
                                left.setText(Card[card][2]);
  } } } } } } } }
  if ((tag_p!=0)&&(open_tag!=1)){
      count_char++;
      FileLink[no_card].write(s);
  }
  open_tag =0;
  break;
}
case 'O' :  // tag <optgroup>
case 'o' :  {
  if (open_tag ==1){
      s = dis.read();
      if (s == 'p'){
          s = dis.read();
          if (s == 't'){
```

```
            s = dis.read();
            if (s == 'g'){
                s = dis.read();
                if (s == 'r'){
                    s = dis.read();
                    if (s == 'o'){
                        s = dis.read();
                        if (s == 'u'){
                            s = dis.read();
                            if (s == 'p'){
                                s = dis.read();
                                if (s == ' '){
                                    tag_optgroup++;
                                    while ((s = dis.read())!='>');
            }   }   }   }   }   }
            else if (s == 'i') {            //tag <option>
                s = dis.read();
                if (s == 'o'){
                    s = dis.read();
                    if (s == 'n'){
                        s = dis.read();
                        if (s == ' '){
                            tag_option++;
                            while ((s = dis.read())!='>');
        }   }   }   }   }   }
        else if (s == 'n'){                 // tag <onevent>
            s = dis.read();
            if (s == 'e'){
                s = dis.read();
                if (s == 'v'){
                    s = dis.read();
                    if (s == 'e'){
                        s = dis.read();
                        if (s == 'n'){
```

```
                        s = dis.read();
                        if (s == 't'){
                            s = dis.read();
                            if (s == ' '){
                                tag_onevent++;
                                while ((s = dis.read())!='>');
    } } } } } } } }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag =0;
    break;
}
case 'S' :  //tag <select>
case 's' :  {
    if (open_tag ==1){
        s = dis.read();
        if (s == 'e'){
            s = dis.read();
            if (s == 'l'){
                s = dis.read();
                if (s == 'e'){
                    s = dis.read();
                    if (s == 'c'){
                        s = dis.read();
                        if (s == 't'){
                            s = dis.read();
                            if (s == ' '){
                                tag_select++;
                                while ((s = dis.read())!='>');
        } } } } } }
        else if (s == 'm') {          //tag <small>
            s = dis.read();
```

```
        if (s == 'a'){
            s = dis.read();
            if (s == 'l'){
                s = dis.read();
                if (s == 'l'){
                    while ((s = dis.read())!='>');
                    tag_small++;
                    FileLink[no_card].write('<');
                    FileLink[no_card].write('s');
                    FileLink[no_card].write('m');
                    FileLink[no_card].write('a');
                    FileLink[no_card].write('l');
                    FileLink[no_card].write('l');
                    FileLink[no_card].write('>');
}   }   }   }
else if (s == 't')    {            //tag <strong>
    s = dis.read();
    if (s == 'r'){
        s = dis.read();
        if (s == 'o'){
            s = dis.read();
            if (s == 'n'){
                s = dis.read();
                if (s == 'g'){
                    s = dis.read();
                    if (s == '>'){
                        tag_strong++;
                        FileLink[no_card].write('<');
                        FileLink[no_card].write('b');
                        FileLink[no_card].write('>');
}   }   }   }   }   }
else if (s == 'e')    {            //tag <setvar>
    s = dis.read();
    if (s == 't'){
```

```
                s = dis.read();
                if (s == 'v'){
                    s = dis.read();
                    if (s == 'a'){
                        s = dis.read();
                        if (s == 'r'){
                            while((s=dis.read())!='>') ;
    }   }   }   }   }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag =0;
    break;
}
case 'U' :
case 'u' :{
    if (open_tag==1) {
        s=dis.read();
        if (s=='>'){
            FileLink[no_card].write('<');
            FileLink[no_card].write('u');
            FileLink[no_card].write('>');
            tag_u++;
    }   }
    if ((tag_p!=0)&&(open_tag!=1)){
        count_char++;
        FileLink[no_card].write(s);
    }
    open_tag=0;
    break;
}
case ' ':{
    if ((tag_p!=0)&&(open_tag!=1)){
```

```
            s=dis.read();
            if(s!=' '){
                count_char++;
                FileLink[no_card].write(' ');
                count_char++;
                FileLink[no_card].write(s);
            }
            else if(s=='<'){
                open_tag=1;
            }
            else {
                FileLink[no_card].write(' ');
                count_char++;
                FileLink[no_card].write(s);
        }   }
        break;
    }
    case '&':{
        s = dis.read();
        if (s == 'a'){
            s = dis.read();
             if (s == 'm') {     // &amp
                FileLink[no_card].write('&');
                while((s=dis.read())!=';');
            }
            else if (s == 'p') {        // &apos
                FileLink[no_card].write('\'');
                while((s=dis.read())!=';');
        }   }
        else if (s == 's')   {     // &shy
            FileLink[no_card].write('-');
            while((s=dis.read())!=';');
        }
        else if (s == 'g') {      // &gt
```

```
            FileLink[no_card].write('&');
            FileLink[no_card].write(s);
            while((s=dis.read())!=';')FileLink[no_card].write(s);
        }
        else if (s == 'l') {        // &lt
            FileLink[no_card].write('&');
            FileLink[no_card].write(s);
            while((s=dis.read())!=';')FileLink[no_card].write(s);
        }
        else if (s == 'n') {       //  
            FileLink[no_card].write(' ');
            while((s=dis.read())!=';');
        }
        else if (s == 'q') { // &quot
            FileLink[no_card].write("");
            while((s=dis.read())!=';');
        }
        special=1;
    }
    default : {
        if ((tag_p!=0)&&(open_tag!=1)&&(special==0)){
            if (s!='\n'){
                count_char++;
                FileLink[no_card].write(s);
        }  }
        break;
}  }
for (i=0;i<10;i++){
    FileLink[i].close();
}
dis.close();
card=0;
show(card);
left.setText(Card[card][2]);
```

```
            System.out.println(count_char+"words");
        }
        catch (IOException event){
            if(load_pic!=1){
                System.out.println("error get url");
                Alert a = new Alert(" URL Error at "+jTextField1.getText());
                a.setSize(400,300);
                a.show();
} } }
```

บรรณานุกรม

[1] ดร.วีระศักดิ์ ซึงถาวร: *" Fundamental of Java Programming Volume 1. "*, SUM Publishing Department, SUM System Company Limited, 2541.

[2] ดร.วีระศักดิ์ ซึงถาวร: *" Fundamental of Java Programming Volume 2. "*, บริษัท ซีเอ็ดยูเคชั่น จำกัน (มหาชน), 2543.

[3] อภิเนตร อูนากูล: *" Object-Oriented Analysis and Design. "*, แผนกตำรา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543.

[4] Mann, Steve : *" Programming Application with the Wireless Application Protocol: The Complete Developer's Guide."*, John Wiley & sons, Inc., 1999.

[5] Merlin Hughes, Michael Shoffner, Derek Hamner: *" Java Network Programming: Second Edition "*, Manning Publications Co., 1999.

[6] Jason Hunter, William Crawford: " Java Servlet Programming ", O'Reilly & Associates, Inc., 1998.

[7] *" WML Reference Version 1.1 "*, http://www.forum.nokia.com.