ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาพจนานุกรมออกเสียงบนโทรศัพท์เคลื่อนที่ด้วย J2ME

Development of Talking Dictionary on Mobile Phones with J2ME

โดย

สยาม จันชูโต

รหัส 46066333

H002260

วัน เดือน ปี 15 ก.พ. 2550
เลขทะเบียน 02260
เลขเรียกหนังสือ วพ ส319ก 2547
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

อาจารย์ที่ปรึกษา

ผศ. ดร. จันทร์บูรณ์ สถิตวิริยวงศ์

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2547
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การพัฒนาพจนานุกรมออกเสียงบนโทรศัพท์เคลื่อนที่ด้วย J2ME
นักศึกษา	นายสยาม จันชูโต
อาจารย์ที่ปรึกษา	ผศ. ดร. จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ในปัจจุบันมีการใช้งานโทรศัพท์มือถือกันอย่างแพร่หลาย เนื่องจากสามารถพกพาได้ และสะดวกในการติดต่อสื่อสาร อีกทั้งโทรศัพท์เคลื่อนที่รุ่นใหม่ๆ สามารถใช้งานฟังก์ชันต่างๆได้มากขึ้น ได้แก่ ส่งข้อความมัลติมีเดีย ถ่ายรูป เล่นอินเตอร์เน็ตและยังสนับสนุนเทคโนโลยีจาวา และซิมเบียน ทำให้พัฒนาแอพลิเคชันต่างๆขึ้นมาใช้งานบนโทรศัพท์เคลื่อนที่ได้ จึงมีความสนใจศึกษาและพัฒนาโปรแกรมพจนุกรมที่สามารถออกเสียงได้ โดยใช้หลักการสังเคราะห์เสียงในการอ่านคำต่างๆ เพื่อแนวทางในการพัฒนาโปรแกรมในลักษณะนี้ให้เป็นประโยชน์มากขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Development of Talking Dictionary on Mobile Phones with J2ME
Student	Mr. Siam janchooto
Advisor	Asst. Prof. Dr. Chanboon Sathitwiriyawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic year	2004

Abstract

Nowadays mobile phone is general equipment that takes to everywhere, ease to communicate with others. The new generations of mobile phones have many additional functions such as taking a photograph, supporting MMS, connecting to the internet. Including java and symbian technology supported. We can develop any applications on mobile phones. So I am interested in the development of talking dictionary on mobile phone with J2ME and speech synthesis principle. These are lead to develop other useful applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ III

สารบัญตาราง V

สารบัญรูป VI

บทที่

1. บทนำ 1

    1.1 ความเป็นมาของปัญหา 1

    1.2 ประโยชน์ที่คาดว่าจะได้รับ 1

    1.3 ขอบเขต 2

2. ทฤษฎีที่เกี่ยวข้อง 3

    2.1 J2ME 3

    2.2 Text to Speech 8

3. การวิเคราะห์และออกแบบระบบ 15

    3.1 สิ่งที่ควรคำนึงถึงในการพัฒนาระบบบนมือถือ 15

    3.2 องค์ประกอบของระบบ 15

    3.3 Activity Diagram 17

    3.4 Sequence Diagram 19

    3.5 Class Diagram 23

4. การพัฒนาระบบ 27

    4.1 ขั้นตอนการพัฒนาระบบ 27

    4.2 ฐานข้อมูล 27

    4.3 ส่วนพจนานุกรม 29

    4.4 TTS Engine 30

    4.5 เครื่องมือที่ใช้ในการพัฒนา 31

    4.6 หน้าจอระบบ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

5. สรุป และข้อเสนอแนะ.....40
5.1 สรุป.....40
5.2 ปัญหา และข้อเสนอแนะ.....40
5.3 งานที่จะทำต่อไป.....41
บรรณานุกรม.....42
ประวัติผู้เขียน.....43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 ตัวอย่างโพรไฟล์สำหรับอุปกรณ์ประเภทต่างๆ....4
2.2 เปรียบเทียบคุณสมบัติของ MIDP 1.0 และ MIDP 2.0....5
4.1 ชื่อไฟล์ที่สัมพันธ์กันระหว่างไฟล์คำศัพท์และคำแปลแบบไทย-อังกฤษ....28
4.2 ชื่อไฟล์ที่สัมพันธ์กันระหว่างไฟล์คำศัพท์และคำแปลแบบอังกฤษ-ไทย....28
4.3 ความสัมพันธ์ของข้อมูลในไฟล์คำศัพท์และคำแปล....29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 โครงสร้างของ J2ME....3
2.2 สถานะการทำงานของ MIDlet....5
2.3 RecordStore ในแต่ละ MIDlet Suite....6
2.4 โครงสร้างภายในของ RecordStore....7
2.5 องค์ประกอบของ TTS Engine....9
3.1 Use-Case Diagram ของระบบ....16
3.2 Activity Diagram ของระบบ....17
3.3 Sequence Diagram ของการแปลความหมาย....19
3.4 Sequence Diagram ของ Talk....20
3.5 Sequence Diagram ของการค้นหาคำศัพท์....21
3.6 Sequence Diagram ของการเปลี่ยนโหมดพจนานุกรม....22
3.7 Class Diagram ของระบบ....23
4.1 หน้าแรกของระบบ MTalkingDict....32
4.2 การป้อนคำลงใน textbox....33
4.3 การ clear คำที่ป้อนไว้....34
4.4 การการอ่านคำศัพท์....35
4.5 การแปลความหมายคำศัพท์....36
4.6 การค้นหาคำศัพท์ด้วย Prefix....37
4.7 การ switch dictionary....38
4.8 การแปลศัพท์ภาษาไทยเป็นอังกฤษ....39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	การพัฒนาพจนานุกรมออกเสียงบนโทรศัพท์เคลื่อนที่ด้วย J2ME
นักศึกษา	นายสยาม จันชูโต
อาจารย์ที่ปรึกษา	ผศ. ดร. จันทร์บูรณ์ สถิตวิริยวงศ์
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ในปัจจุบันมีการใช้งานโทรศัพท์มือถือกันอย่างแพร่หลาย เนื่องจากสามารถพกพาได้ และสะดวกในการติดต่อสื่อสาร อีกทั้งโทรศัพท์เคลื่อนที่รุ่นใหม่ๆ สามารถใช้งานฟังก์ชันต่างๆได้มากขึ้น ได้แก่ ส่งข้อความมัลติมีเดีย ถ่ายรูป เล่นอินเตอร์เน็ตและยังสนับสนุนเทคโนโลยีจาวา และซิมเบียน ทำให้พัฒนาแอพลิเคชันต่างๆขึ้นมาใช้งานบนโทรศัพท์เคลื่อนที่ได้ จึงมีความสนใจศึกษาและพัฒนาโปรแกรมพจนุกรมที่สามารถออกเสียงได้ โดยใช้หลักการสังเคราะห์เสียงในการอ่านคำต่างๆ เพื่อแนวทางในการพัฒนาโปรแกรมในลักษณะนี้ให้เป็นประโยชน์มากขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Development of Talking Dictionary on Mobile Phones with J2ME
Student	Mr. Siam janchooto
Advisor	Asst. Prof. Dr. Chanboon Sathitwiriyawong
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic year	2004

Abstract

Nowadays mobile phone is general equipment that takes to everywhere, ease to communicate with others. The new generations of mobile phones have many additional functions such as taking a photograph, supporting MMS, connecting to the internet. Including java and symbian technology supported. We can develop any applications on mobile phones. So I am interested in the development of talking dictionary on mobile phone with J2ME and speech synthesis principle. These are lead to develop other useful applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

ปัจจุบันมีการใช้งานโทรศัพท์มือถือกันอย่างแพร่หลาย จนเป็นสิ่งจำเป็นในการติดต่อสื่อสาร และต้องพกติดตัวอยู่เสมอ มีการใช้งานในฟังก์ชันต่างๆเพิ่มขึ้นไม่ว่าจะเป็นการเล่นอินเตอร์เน็ต ส่งอีเมล์ เล่นเกมส์ และอีกมากมาย ซึ่งการใช้งานเหล่านี้ต้องมีการพัฒนาแอพลิชันขึ้นมารองรับ เทคโนโลยีจาวาเป็นแนวทางหนึ่งที่จะช่วยในการพัฒนาแอพลิชัน โดยผู้พัฒนาระบบส่วนใหญ่มีความคุ้นเคยกับภาษาจาวาอยู่แล้วเพียงแต่ปรับเปลี่ยนบางส่วน ที่มีความแตกต่างกันอันเนื่องมาจากข้อจำกัดของทรัพยากรที่มีน้อยกว่า

ในความเป็นจริงเทคโนโลยีจาวา และ การสังเคราะห์เสียงพูดไม่ใช่เรื่องใหม่ แต่เมื่อนำทั้งสองมาผนวกกันเพื่อพัฒนาแอพลิชันบนโทรศัพท์มือถือ ทำให้ประโยชน์ในการใช้งานโทรศัพท์มือถือเพิ่มขึ้น อีกทั้งยังเป็นแนวทางในการพัฒนาระบบอื่นๆที่ประโยชน์มากขึ้น อาทิเช่น ระบบช่วยคนตาบอดในการใช้งานโทรศัพท์เคลื่อนที่ หรือ อุปกรณ์ขนาดเล็กอื่นๆให้สามารถใช้งานได้สะดวกมากขึ้น

จะเห็นว่าในแง่ของการใช้งานพจนานุกรมออกเสียงบนโทรศัพท์เคลื่อนที่นั้น มีความสามารถไม่เท่าบนคอมพิวเตอร์ หรือ บนอุปกรณ์เฉพาะสำหรับพจนานุกรมออกเสียง แต่ถือเป็นการเพิ่มประโยชน์ใช้สอยให้มากขึ้น

1.2 ประโยชน์ที่คาดว่าจะได้รับ

แนวคิดในการพัฒนาพจนานุกรมออกเสียงได้นั้น เพื่อเป็นการศึกษาเทคโนโลยีในการพัฒนาระบบบนโทรศัพท์เคลื่อนที่ ซึ่งถือว่าเป็นเรื่องค่อนข้างใหม่ และมีการใช้หลักของการอ่านคำโดยการสังเคราะห์เสียงซึ่งจะเพิ่มประโยชน์ในการใช้งานพจนานุกรมมากขึ้น อีกทั้งจากการศึกษาในเรื่องการสังเคราะห์เสียงบนโทรศัพท์บนเคลื่อนที่จะเป็นแนวทางในการพัฒนาแอพลิเคชันอื่นๆที่มีประโยชน์มากขึ้น อาธิเช่น การอ่านออกเสียงเมนูบนโทรศัพท์เคลื่อนที่ เพื่อประโยชน์สำหรับการใช้งานโทรศัพท์ของคนตาบอด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขต

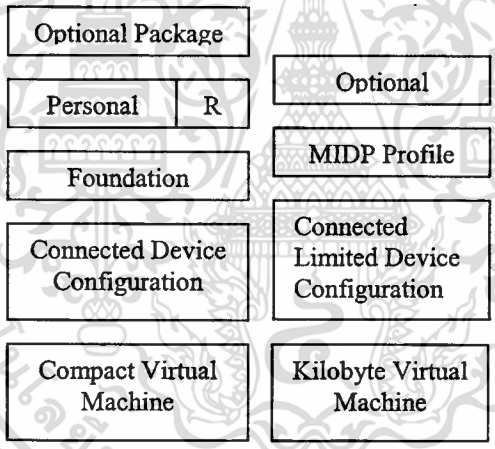
- ในการพัฒนาระบบพจนานุกรมบนเคลื่อนที่เป็นการพัฒนาด้วยเทคโนโลยีจาวา ซึ่งจะสนับสนุนการใช้งานบนโทรศัพท์เคลื่อนที่บางรุ่นเท่านั้น
- เนื่องด้วยข้อจำกัดของโทรศัพท์มือถือที่มีทรัพยากรจำกัด ทำให้เนื้อที่สำหรับเก็บคำศัพท์มีจำกัด ไม่สามารถเพิ่มเติมคำศัพท์ได้ครบทุกคำ
- การอ่านคำออกเสียง อ่านได้เฉพาะคำภาษาอังกฤษเท่านั้น
- การแปลความหมายจะเป็นแบบ อังกฤษ-ไทย และ ไทย-อังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2. 1 J2ME (Java 2 Platform, Micro Edition)

เป็นเทคโนโลยีจาวาที่ใช้สำหรับเขียนโปรแกรมบนอุปกรณ์ขนาดเล็กที่มีทรัพยากรจำกัด ได้แก่ โทรศัพท์มือถือและ PDA เป็นต้น ซึ่งอุปกรณ์เหล่านี้มีขนาดหน่วยความจำ การแสดงผล และความสามารถในการประมวลผลจำกัด โครงสร้างของ J2ME ปันดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของ J2ME

จากโครงสร้างของ J2ME จะเห็นว่าต้องมี Virtual Machine (VM) เหมือนโปรแกรมจาวาโดยทั่วไป เพื่อทำหน้าที่แปลงไบต์โค้ดเป็นภาษาเครื่องให้สามารถทำงานได้ แต่จะมีการแบ่งอุปกรณ์ออกเป็น 2 ประเภท ตามคุณสมบัติของหน่วยความจำ การแสดงผล และความสามารถในการประมวลผล

1) CDC (Connected Device Configuration) คุณสมบัติของอุปกรณ์ในกลุ่มนี้ได้แก่ มีหน่วยความจำตั้งแต่ 2-16 เมกะไบต์ มีหน่วยประมวลผลขนาด 32 บิต เป็นอย่างน้อย โดยมีความเร็วในการเชื่อมต่อเครือข่ายค่อนข้างสูง ตัวอย่างของอุปกรณ์ในกลุ่มนี้ ได้แก่ Pocket PC และ Set-Top Box ของเคเบิลทีวี เป็นต้น (กาญจนา ตันวิสุทธิ์. 2547)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) CLDC (Connected Limited Device Configuration) คุณสมบัติของอุปกรณ์ในกลุ่มนี้ ได้แก่ มีหน่วยความจำ 160 - 512 กิโลไบต์ โดยควรมีหน่วยความจำแบบถาวรอย่างน้อย 128 กิโลไบต์ เพื่อเก็บไลบารี และ VM ของ CLDC และควรมีหน่วยความจำแบบชั่วคราว อย่างน้อย 32 กิโลไบต์ สำหรับ VM ใช้ในการรันโปรแกรม ส่วนหน่วยประมวลผลควรเป็น 16-32 บิต ซึ่งมีความเร็วอย่างน้อย 25 เมกะเฮิร์ต แต่จะมีข้อจำกัดในการแสดงผล และ ความเร็วในการเชื่อมต่อเครือข่ายค่อนข้างต่ำ ตัวอย่างอุปกรณ์ที่จัดอยู่ในกลุ่มนี้ ได้แก่ โทรศัพท์มือถือ เพจเจอร์ เป็นต้น (กาญจนา ตันวิสุทธิ์. 2547)

ฟังก์ชันต่างๆของ CDC และ CLDC ส่วนใหญ่จะสืบทอดมาจาก J2SE แต่ปรับเปลี่ยนและเพิ่มเติมขึ้นมาเพื่อให้เหมาะสมกับการทำงานบนอุปกรณ์ขนาดเล็กที่มีทรัพยากรจำกัด

โดยจะใช้โพรไฟล์เป็นตัวกำหนดกลุ่มของไลบรารีที่เพิ่มเติมมาจาก Configuration เดิม เพื่อรองรับข้อแตกต่างของอุปกรณ์แต่ละชนิด เพราะคุณลักษณะของอุปกรณ์แต่ละชนิดไม่เหมือนกันเช่น ส่วนแสดงผล ส่วนรับอินพุทจากผู้ใช้ หรือ ส่วนติดต่อกับเครือข่าย เป็นต้น

ตารางที่ 2.1 ตัวอย่างโพรไฟล์สำหรับอุปกรณ์ประเภทต่างๆ

Profile	Configuration	ตัวอย่างอุปกรณ์
MIDP	CLDC	โทรศัพท์มือถือ, เพจเจอร์
PDAP	CLDC	PDA
Personal	CDC	Pocket PC
RMI	CDC	อุปกรณ์ใดๆ

จากตารางที่2.1 จะเห็นว่าอุปกรณ์แต่ละชนิดมีโพรไฟล์ที่แตกต่างกัน แต่ส่วนที่เราสนใจคือ MIDP (Mobile Information Profile) ซึ่งเป็นไลบรารีที่รองรับการเขียนโปรแกรมบนโทรศัพท์มือถือ ซึ่งจะต้องมีไลบรารีที่สนับสนุนการพัฒนาโปรแกรมจำพวกการสังเคราะห์เสียงพูดที่เรียกว่า JSAPI (Java Speech API)

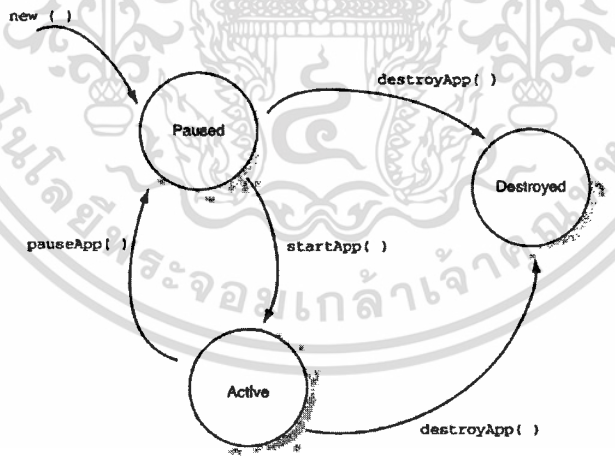
MIDP 2.0 ซึ่งเป็นเวอรชันใหม่มีข้อแตกต่างจาก MIDP 1.0 โดยมีการปรับปรุงและเพิ่มเติมในบางส่วนโดยมีรายละเอียดข้อเปรียบเทียบดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 เปรียบเทียบคุณสมบัติของ MIDP 1.0 และ MIDP 2.0

คุณสมบัติ	MIDP 1.0	MIDP 2.0
การเชื่อต่อเครือข่าย	HTTP	HTTP และ HTTPS
Media API	ไม่มี	มี
Game API	ไม่มี	มี
User Interface	มี	ได้มีการปรับปรุงในส่วนของ - การจัดวางส่วนต่างในฟอร์ม - การจัดการคำสั่ง - สามารถสร้างไอเท็มใหม่ได้เอง
RGB Image	ไม่มี	มี

ในการสร้างแอพลิเคชั่นบนอุปกรณ์ประเภท MIDP จะเรียกว่า MIDlet ซึ่งการทำงานจะคล้ายๆ กับโปรแกรมที่ทำงานบนบราวเซอร์ โดยมีสถานะการทำงานดังนี้



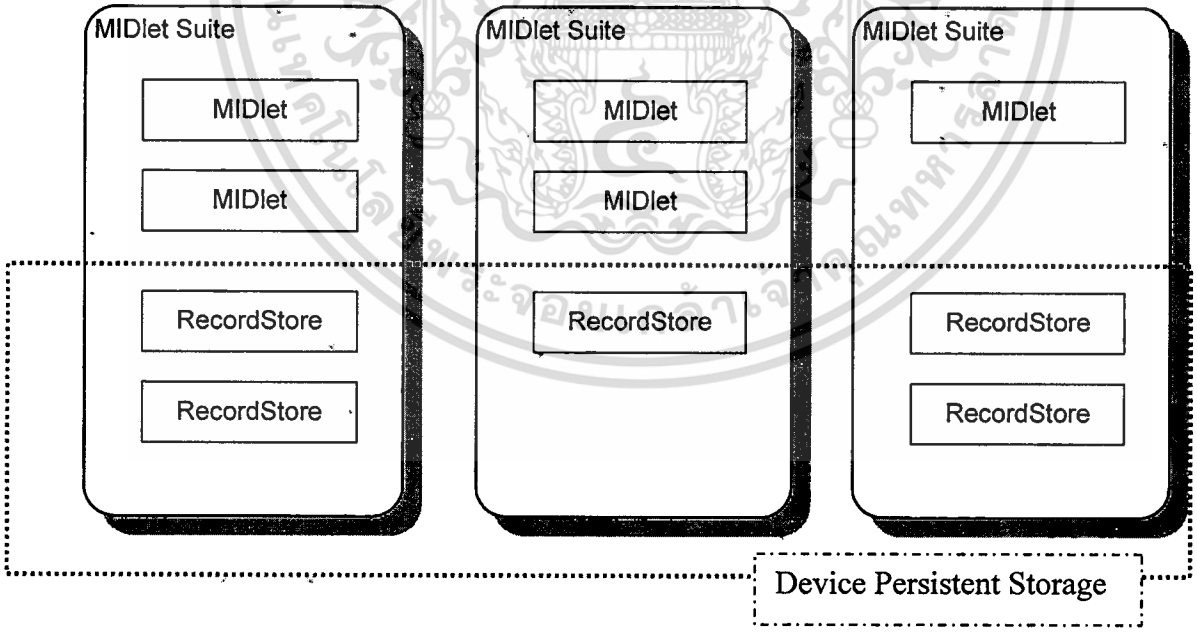
รูปที่ 2.2 สถานะการทำงานของ MIDlet

จากรูปที่ 2.2 มีการทำงาน 3 ลักษณะด้วยกันคือ startApp( ) เป็นการจัดหาทรัพยากรในระบบเพื่อกำหนดการเริ่มต้นทำงาน pauseApp( ) เป็นการหยุดการทำงานชั่วคราวเพื่อย้ายการทำงานไปยังส่วนอื่นๆ destroyApp( ) เป็นการยกเลิกการใช้งานทรัพยากรที่จัดหามาและหยุดการทำงานของ MIDlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการพัฒนาระบบบนโทรศัพท์เคลื่อนที่ในบางครั้งจำเป็นต้องเก็บข้อมูลถาวรไว้ใช้ในครั้งต่อไป เช่น ถ้าเป็นเกมข้อมูลที่มักจะเก็บเอาไว้ก็คือ คะแนนสูงสุดที่ผู้เล่นเคยทำได้ ถ้าเป็นโปรแกรมจดบันทึกจะต้องเก็บข้อมูลที่ป้องเอาไว้ หรือโปรแกรมที่ต้องใช้ฐานข้อมูลอย่างเช่น พจนานุกรมต้องเก็บข้อมูลด้วย ในโทรศัพท์เคลื่อนที่จะมีหน่วยความจำอยู่ 2 ประเภท ประเภทแรกใช้ในการทำงานของแอพลิเคชัน และอีกประเภทใช้สำหรับเก็บข้อมูลซึ่งจะเก็บได้ถาวรกว่าหน่วยความจำประเภทแรก เรามักจะเรียกหน่วยความจำส่วนนี้ว่า ฐานข้อมูล โดยใน MIDP มีระบบการจัดการฐานข้อมูลแบบเร็คคอร์ดง่ายๆ ที่เรียกว่า Record Management System (RMS)

โครงสร้างของ Record Management System หรือ RMS ประกอบด้วย 2 ส่วนคือ RecordStore และ เร็คคอร์ด (Record) โดยตัวข้อมูลจริงๆจะถูกเก็บแยกกันอยู่คนละเร็คคอร์ด และเร็คคอร์ดที่มีข้อมูลเกี่ยวข้องกันจะเก็บรวมกันอยู่ใน RecordStore ซึ่งถ้าเทียบกับฐานข้อมูลที่ใช้กันบนเครื่องคอมพิวเตอร์แล้ว RecordStore จะเทียบได้กับตาราง ส่วน Record คือข้อมูลแต่ละแถวในตารางนั่นเอง โดยที่ MIDlet จะสามารถใช้งาน RecordStore ที่อยู่ภายใน MIDlet Suite เดียวกันได้เท่านั้น (Sun Microsystems. 2005a)

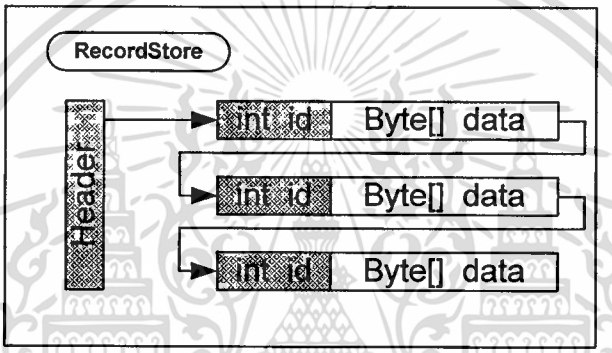


รูปที่ 2.3 RecordStore ในแต่ละ MIDlet Suite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จำนวนเร็คคอร์ดทั้งหมดเริ่มต้นจาก 0
- หมายเลขเวอร์ชั่น โดยจะเป็นจำนวนเต็มที่มากกว่า 0 และจะเพิ่มขึ้นทุกครั้งที่มีการแก้ไข RecordStore
- เวลาครั้งล่าสุดที่มีการเปลี่ยนแปลงแก้ไข RecordStore
- ค่า RecordID ต่อไปสำหรับการเพิ่มเร็คคอร์ดเข้ามาใน RecordStore ซึ่งจะเพิ่มขึ้นครั้งละ 1

ส่วนข้อมูลใน RecordStore จะเก็บเป็นเร็คคอร์ด ซึ่งแต่ละเร็คคอร์ดจะเชื่อมโยงเข้าหากันโดยมีพอยน์เตอร์ชี้ไปยังเร็คคอร์ดต่อไป และเร็คคอร์ดแรกจะถูกเชื่อมโยงกับข้อมูลส่วนหัวดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างภายในของ RecordStore

อีกส่วนหนึ่งคือการใช้งาน Media API ของ J2ME โดย Media API เป็นกลุ่มย่อยของ Mobile Media API(MMAPI) ซึ่งไม่ได้รองรับการทำงานกับ Multimedia ทั้งหมด แต่จรองรับการทำงานกับ Audio เท่านั้น ตัวอย่างการทำงานที่รองรับโดย Media API ได้แก่

- การสร้างเมโลดี้
- การเล่นไฟล์ Audio
- พื้นฐานการควบคุมทั่วไป เช่น เล่น หยุด หรือหยุดชั่วคราว
- การควบคุมระดับความดังของเสียง

โครงสร้างของ Media API ประกอบด้วย Manager, Control และ Player โดย Player จะเป็นตัวเล่นไฟล์เสียง ส่วน Manager เป็นตัวจัดการ Player แต่ละตัว และสุดท้ายส่วน Control แบ่งออกเป็น 2 ประเภทคือ ToneControl ใช้ในการควบคุมการเล่นของโน้ตเสียง และVolumeControl ใช้ในการควบคุมความดังของเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Text to Speech (TTS)

Thierry Dutoid (1997) ได้อธิบายไว้ว่า TTS เป็นกระบวนการที่เกี่ยวข้องกับการแปลงคำ หรือตัวอักษรให้อยู่ในรูปแบบเสียง แล้วเปล่งเสียงออกมาให้ผู้ใช้ได้ยิน ซึ่งส่วนมากจะใช้เทคนิคการผสมเสียงๆเข้าด้วยกันให้สามารถฟังแล้วเข้าใจได้ โดยสามารถแบ่งเป็น 2 ประเภทหลักๆดังนี้

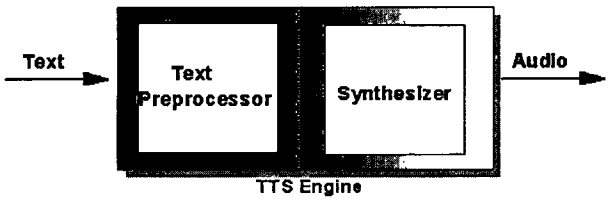
1) Concatenated TTS

เป็นกระบวนการที่ทำให้เครื่องคอมพิวเตอร์พูดได้อีกแบบหนึ่ง แต่มิใช่เป็นกระบวนการสังเคราะห์เสียงอย่างแท้จริง แต่จะใช้เทคนิคขั้นพื้นฐานของระบบ TTS โดยที่ Concatenated Engine จะเป็นการออกแบบ Application โดยจะใช้การบันทึกคำพูดทั้งคำลงใน ฐานข้อมูล เมื่อมีการเรียกใช้ Application นี้ ตัว Engine จะทำ การดึงข้อมูลเสียงจากฐานข้อมูล ออกมาแล้วนำแต่ละตัวมาต่อกันเพื่อออกเสียงให้เป็นประโยค ตัวอย่างเช่น Voice mail เมื่อมี mail ใหม่เข้ามาแล้วเราเข้าไปเช็ค เราจะได้ยินว่า [You have][three][new message] ซึ่งตัวโปรแกรมที่ใช้นี้ จะต้องบันทึกกลุ่มคำซึ่งได้แก่ "You have" , คำพูดที่แสดงตัวเลขทั้งหมด,และ "new massage"

2) Synthesis TTS

ใน TTS Engine จะใช้กระบวนการสังเคราะห์เสียงในการผลิตเสียงเพื่อที่จะ พยายามให้เหมือนเสียง มนุษย์ให้มากที่สุด เช่นการขยับของลิ้น ปากหรือแม้กระทั่งภายในช่องปาก เสียงที่ถูกสร้างขึ้นโดยกระบวนการสังเคราะห์ เสียงนี้จะได้คุณภาพของเสียงน้อยกว่ากระ บวนการในแบบแรก แต่ไม่มากนัก - Subword Segment. กระบวนการนี้ TTS Engine จะทำการ ปะติดปะต่อเสียงที่อยู่ในรูปแบบ short digital-audio แล้วทำการสร้างส่วนของเสียงขึ้นมาใหม่ให้ มีความนุ่มนวลของเสียงมากยิ่งขึ้น ตัวอย่างเช่น ในแต่ละ segment ของเสียงจะประกอบไปด้วย phonemes 2 ตัว ตัวแรกจะเป็นตัวเริ่มต้นของเสียง ส่วนอีกตัวจะเป็นตัวสิ้นสุด ดังเช่นคำว่า "hello" จะประกอบไปด้วย phonemes ทั้งหมด 4 ตัว คือ h eh l oe แล้วจะถูกนำไปรวมกันเป็น subword segment ที่ประกอบด้วย silence-h h-eh eh-l l-oe oe-silence ซึ่งจะทำให้เสียงที่ได้มามีความต่อเนื่องกันยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 องค์ประกอบของ TTS Engine

โดยมีกระบวนการทำงาน 5 ขั้นตอนดังนี้

1) Text Normalization

Text Normalization เป็นขั้นตอนที่ช่วยในการลดความซับซ้อนของโครงสร้างของ Text ให้เป็นโครงสร้างที่ง่าย ไม่ซับซ้อน โดยทำการแยก string เช่น "John rode home" ออกให้เป็นคำๆ คือ "John","rode", "home" อย่างไรก็ตาม เทคนิคนี้ค่อนข้างจุกจิกและยุ่งยากเมื่อ string มีความซับซ้อนมากๆ เช่น "John rode home at 23.5 mph" ส่วนของ "23.5 mph" จะถูกเปลี่ยนให้เป็น "twenty three point five mile per hour" โดยขั้นตอนนี้จะทำงานดังนี้

ขั้นแรก จะทำการแยกคำออกและค้นหาว่าคำที่แยกออกมานั้นมีสัญลักษณ์พิเศษ คือ apostrophe หรือ hyphen อยู่ด้วยหรือไม่ หากมีระบบจะไม่แยกคำที่อยู่หน้าและหลังสัญลักษณ์พิเศษนี้ออกจากกัน แต่จะถือว่าเป็นคำเดียวกัน จากนั้นจะทำการค้นหาตัวเลข , วันที่ , เวลา และเครื่องหมายพิเศษต่างๆ แล้วนำมาเปลี่ยนให้เป็นคำ เช่น "$54.32" จะถูกเปลี่ยนให้เป็น "fifty four dollars and thirty two cents" ซึ่งคำและสัญลักษณ์ต่างๆจะแตกต่างกันไปขึ้นอยู่กับภาษาที่ใช้และคำที่แวดล้อมมันอยู่

ขั้นต่อไป ระบบจะทำการค้นหาคำย่อต่างๆ แล้วเปลี่ยนคำเหล่านั้นให้เป็นคำเต็ม เช่น "in." จะเปลี่ยนให้เป็น "inches" , "St." จะเปลี่ยนให้เป็น "street" หรือ "saint" ขึ้นอยู่กับคำที่อยู่รอบข้างตัวมัน นอกจากที่กล่าวมาข้างต้นแล้ว Text Normalization ยังสามารถทำการเปลี่ยน text ในรูปแบบอื่นๆได้อีก เช่น Internet Address "http://www.Microsoft.com" จะอ่านเป็น "w w w dot Microsoft dot com" ซึ่งผู้ทำการ Normalize จะพิจารณาสัญลักษณ์ที่เป็นตัวคั่นระหว่างคำว่าควรจะออกเสียงหรือไม่ เช่น จุด (.) หากวางอยู่ที่ท้ายประโยคจะไม่ออกเสียง แต่ถ้าหากวางอยู่ระหว่างคำใน Internet Address จะอ่านออกเสียงเป็น "dot" เป็นต้น เมื่อทำการ Normalize แล้ว ขั้นตอนต่อไปคือ การลดความซับซ้อนของคำที่เขียนเหมือนกันแต่ออกเสียงต่างกัน หรือที่เรียกว่า Homograph Disambiguation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Homograph Disambiguation

ในภาษาอังกฤษหรือหลายๆภาษา มีคำไม่ต่ำกว่าร้อยคำที่เขียนเหมือนกัน แต่อ่านออกเสียงต่างๆกันไป ดังนั้น คำว่า "Homograph" จึงหมายถึงคำที่เขียนเหมือนกัน แต่อ่านออกเสียงต่างกัน ซึ่งหมายความรวมถึงคำย่อและตัวเลขต่างๆด้วย เช่นตัวย่อ "Ft." จะอ่านออกเสียงต่างกัน ระหว่าง "Ft. Wayne" กับ "100 ft." ในทางเดียวกัน ตัวเลขเช่น "1997" อาจจะอ่านเป็น "nineteen ninety seven" หรือ "one thousand nine hundred and ninety seven" ในกรณีที่หมายถึงปีก็ได้

ขั้นตอน Homograph Disambiguation จะใช้หลายเทคนิคเข้ามาช่วยเพื่อลดความสับสนในการอ่านออกเสียง ซึ่งมีขั้นตอนการทำงาน คือ จะดูภาพรวมก่อนว่าประโยคนี้พยายามจะกล่าวถึงอะไร แล้วจึงมาพิจารณาคำที่ต้องการว่าควรจะมีความหมายอย่างไรจึงจะสอดคล้องกับข้อความทั้งหมด เมื่อทราบความหมายแล้วจะสามารถเดาเสียงที่จะอ่านออกมาได้

สรุปการทำงานของขั้นตอน Homograph Disambiguation ได้คือ จะดูความหมายของคำรอบๆข้างก่อน จากนั้นจึงแปลความหมายของคำที่ต้องการ แล้วตัดสินใจว่าจะอ่านออกเสียงอย่างไรนั่นเอง

เมื่อทำการลดความซับซ้อนในการอ่านออกเสียงคำที่เป็น Homograph แล้ว คำเหล่านั้นจะถูกส่งไปยังขั้นตอนถัดไป

3) Word Pronunciation

ข้อมูลที่ถูกนำเข้าในขั้นตอนนี้จะเป็น text และ output ที่ออกมาจะเป็นกลุ่มของ phonemes เหมือนที่เป็นคำอ่านที่เราเห็นใน dictionary

ในกระบวนการแปลงตัวอักษรให้เป็นคำอ่านออกเสียง อันดับแรก TTS Engine จะทำการตรวจสอบดูว่าคำที่จะทำการแปลงเป็นคำศัพท์เฉพาะที่อยู่ในฐานข้อมูลของ TSS Engine หรือไม่ ถ้าไม่เป็นก็จะทำการแปลงโดยใช้วิธีการ "letter to sound"

Letter-to-sound เป็นกระบวนการที่จะทำการแปลง text ให้อ่านออกเสียงในรูปแบบของคำแต่ละคำ ซึ่งจะมีเทคนิควิธีการในการสะกดเหมือนกับที่เราเรียนในภาษาอังกฤษ มีอัลกอริทึมหลายอัลกอริทึมที่ช่วยในการทำ แต่ในที่นี้จะขอยกตัวอย่างอัลกอริทึมอย่างง่าย ดังนี้

วิธีการทำงานของ letter-to-sound คือเริ่มต้นจากการสร้าง lexicon หรือพจนานุกรม ซึ่งจะใช้เป็นฐานข้อมูลคำศัพท์และคำอ่าน การบันทึกข้อมูลจะทำได้โดยผู้สร้างฐานข้อมูลเอง เช่น ถ้าเราต้องการใส่ข้อมูลคำว่า hello ข้อมูลที่ถูกบันทึกก็จะเป็น ดังนี้

Hello h eh l oe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมนี้จะถูกใช้ในการแยกส่วนประกอบของคำศัพท์ แล้วทำการประมวลผลว่าตัวอักษรตัวใดจะออกเสียงอย่างไร ซึ่งอธิบายได้จากคำว่า hello

ตัวอักษร “h” ใน hello จะกลายเป็นตัวสร้างพยางค์เสียง “h”

ตัวอักษร “e” จะสร้างพยางค์เสียงที่ออกเสียงเป็น “eh”

ตัวอักษร “l” ตัวแรกจะสร้างพยางค์เสียงที่ออกเสียงเป็น “l”

ส่วนตัวอักษร “l” ที่สองไม่ต้องทำการสร้างพยางค์เสียง

สุดท้ายตัวอักษร “o” จะใช้ในการสร้างพยางค์เสียง “oe”

สำหรับในคำศัพท์ที่ต่างกัน การสร้างพยางค์เสียงก็จะไม่เหมือนกันด้วย เช่น “he” ตัวอักษร “e” จะสร้างพยางค์เสียงที่ออกเสียงเป็น “ee” ซึ่งต่างจาก “e” ใน “hello”

คำศัพท์ 1 คำจะประกอบไปด้วยหลายพยางค์เสียง ซึ่งอัลกอริทึมต่อไปนี้จะเป็นการตรวจสอบว่าอักษรหรือกลุ่มของอักษรใดควรที่จะใช้พยางค์เสียงตัวไหน ในขั้นแรกจะใช้การแทนตัวอักษรด้วยพยางค์เสียงที่เป็นพื้นฐานก่อน ตัวอย่างเช่น “H” ส่วนใหญ่จะใช้พยางค์เสียงเป็น “h” หรือ “O” ส่วนมากจะใช้พยางค์เสียงเป็น “ow” ขั้นตอนต่อไปต้องไปทำการตรวจสอบอักษรที่อยู่รอบข้าง เช่นเมื่อตรวจสอบได้ว่า “O” ปรากฏอยู่ในลำดับสุดท้ายของคำศัพท์และอักษรตัวที่อยู่ข้างหน้าเป็น “L” พยางค์เสียงของ “O” นี้ก็จะแทนด้วย “oe” แทนที่จะเป็น “ow” ตามปกติ

ตัวอย่างของอัลกอริทึม Letter-to-sound สามารถอธิบายได้จากตัวอย่างต่อไปนี้คือ เมื่อต้องการแปลงคำศัพท์ว่า “hello” เป็นเสียง อันดับแรกต้องทำการสร้างพยางค์เสียงของ “h” แต่ก่อนที่จะแปลงต้องไปตรวจสอบดูว่าข้อมูลในฐานข้อมูล Lexicon มีชุดของตัวอักษร “he” หรือไม่ ถ้าไม่มีก็จะออกเสียงตามปกติคือใช้พยางค์เสียง “h” ต่อมาก็จะทำการตรวจสอบอักษร “e” ในฐานข้อมูล Lexicon ว่ามี “e” ที่ถูกนำหน้าด้วย “h” และตามหลังด้วย “l” หรือไม่ถ้ามีจะต้องใช้พยางค์เสียงตัวไหน ซึ่งจะได้พยางค์เสียงเป็น “eh” แล้วก็ทำการตรวจสอบตัวต่อ ๆ ไปโดยใช้เทคนิควิธีการเดียวกัน

จะเห็นได้ว่าในอัลกอริทึมนี้สามารถที่ทำการแปลงอักษรเป็นพยางค์เสียงได้ทุกคำศัพท์ถึงแม้ว่าจะเป็นศัพท์ที่ไม่เคยเห็นมาก่อนก็ตาม แต่วิธีการนี้ก็มีข้อจำกัดเหมือนกันโดยเฉพาะการอ่านออกเสียงสำหรับชื่อของคน เนื่องจากชื่อคนส่วนมากจะไม่ได้มาจากภาษาอังกฤษแท้ ๆ

เมื่อทำการวิเคราะห์และสร้างพยางค์เสียงเสร็จเรียบร้อยแล้วก็จะส่งผลลัพธ์ที่ได้ไปในขั้นตอนต่อไป ก็คือขั้น Prosody

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Prosody

Prosody หรือฉันทลักษณ์ คือ ระดับเสียงสูงต่ำ, ความเร็วในการออกเสียงรวมไปถึงความหนักเบาของการออกเสียง พยางค์ คำหรือแม้กระทั่งประโยค กระบวนการนี้จึงเป็นส่วนประกอบที่สำคัญของ TTS engine เนื่องจากถ้าขาดกระบวนการนี้ไปเสียงที่ส่งออกมาก็จะมีลักษณะที่ราบเรียบไม่มีการเน้นเสียงสูงต่ำ

มีหลายเทคนิควิธีการที่ใช้ในการสังเคราะห์ฉันทลักษณ์ ในที่นี้จะอธิบายถึงเทคนิคที่สามารถเข้าใจได้ง่าย ๆ ในการช่วยในกระบวนการผลิตนี้ซึ่งแบ่งออกเป็น 3 ขั้นตอนดังนี้

Sentence level: เป็นขั้นตอนแรกสุดโดย TTS engine จะทำการตรวจสอบจุดเริ่มต้นและจุดสิ้นสุดของประโยค ซึ่งในประโยคภาษาอังกฤษส่วนมากมักจะใช้เสียงที่ต่ำในการลงท้ายประโยคและจะใช้เสียงสูงในกรณีที่เป็นประโยคคำถาม เช่นเดียวกับความเร็วในการพูดออกเสียงในตอนต้นประโยคความเร็วจะมากแต่จะค่อย ๆ ช้าลงตอนสิ้นสุดประโยคและระหว่างประโยคก็จะมีการเว้นช่วงเวลาเล็กน้อย

Phrase level: TTS engine ต้องทำการตรวจสอบชนิดของถ้อยคำ เช่นถ้อยคำที่เป็นคำกริยาหรือถ้อยคำที่อยู่ในรูปคำนาม เช่นเดียวกันกับการออกเสียงของประโยคแต่จะไม่ชัดเท่า TTS engine สามารถที่จะตรวจสอบชนิดของถ้อยคำได้โดยอาศัย Part-of-speech information ที่ได้มาจากขั้นตอน Homograph disambiguation เช่นการออกเสียงจะหยุดชั่วขณะหนึ่งเมื่อตรวจพบเครื่องหมายคอมมา (,) ระหว่างคำเป็นต้น

ขั้นตอนต่อไป TTS engine ก็ต้องทำการตรวจสอบดูว่าคำที่อยู่ในประโยคคำไหนที่ต้องให้ความสำคัญในการเน้นเสียงเมื่อพบแล้วก็จะทำการกำหนดว่าคำ ๆ นั้นต้องทำการเน้นเสียง การเน้นเสียงคือการออกเสียงด้วยเสียงที่ดังขึ้นหรือยาวขึ้นหรืออาจจะใช้เสียงที่สูงขึ้นเป็นต้นส่วนคำที่ต้องให้ความสำคัญน้อยกว่าก็จะไม่มีการเน้นเสียง ดังตัวอย่างเช่นในประโยค

"John and Bill walked to the store"

เมื่อทำการเน้นเสียงก็จะกลายเป็น

"JOHN and BILL walked to the STORE"

Word level: มาถึงขั้นตอนนี้ก็เป็นการตรวจสอบฉันทลักษณ์ภายในคำศัพท์ซึ่งจะออกเสียงหนักเบาสูงต่ำตามพยางค์ตามหลักของการพูดภาษาอังกฤษ

เมื่อทำเสร็จทุกขั้นตอนแล้ว ตัวบ่งบอกระดับเสียง,ความยาวในการออกเสียงและความหนักเบาของเสียงที่จากการวิเคราะห์ในแต่ละขั้นตอนก็จะถูกนำมารวมกันเพื่อสร้าง Output สุดท้ายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Play Audio

ในกระบวนการสังเคราะห์เสียงจะสิ้นสุดที่ขั้นตอนนี้ซึ่ง TTS engine ส่วนใหญ่ต้องทำการเปลี่ยนข้อมูลของพยางค์เสียงและคุณสมบัติต่าง ๆ ของตัวพยางค์เสียงที่ได้มาจากขั้นตอนต่าง ๆ ให้อยู่ในรูป Digital audio

ซึ่งในกระบวนการผลิต digital audio นี้มีอยู่หลายเทคนิควิธีการ แต่โดยมาก TTS engine จะทำการผลิตด้วยเทคนิคการต่อพยางค์เสียงจากพยางค์เสียงที่เก็บอยู่ในฐานข้อมูล ซึ่งฐานข้อมูลเสียงเหล่านี้ได้มาจากเสียงที่ผลิตโดยมนุษย์จริง ๆ อธิบายอย่างง่าย ๆ ก็คือ TTS engine จะรับพยางค์เสียงมาจากขั้นตอนก่อนหน้านี้แล้วทำการดึงข้อมูลเสียงออกมาจากฐานข้อมูลเสียง ซึ่งรวมถึงข้อมูลที่เป็นข้อกำหนดถึงลักษณะการออกเสียงต่าง ๆ ด้วย

แต่ก็มีหลายเหตุผลเลยทีเดียวที่ทำให้กระบวนการดังกล่าวไม่ง่ายอย่างที่คิดจะเห็นได้ชัดจากข้อมูลของพยางค์พยางค์เสียงเมื่อทำการแปลงเป็นข้อมูลเสียงแล้วนำมาต่อกันจะเกิดความไม่ต่อเนื่องของเสียงระหว่างพยางค์เสียง 2 ตัว ซึ่ง TTS engine สามารถแก้ไขปัญหาเหล่านี้ด้วยการทำการผสมผสานขอบของเสียงระหว่าง 2 พยางค์เสียงโดยทำการอินเตอร์เซกชั่นให้ขอบของพยางค์เสียงทั้งสองมีระดับเสียงที่อยู่ในระดับเดียวกัน

เสียงที่ถูกสร้างขึ้นโดยมนุษย์นั้นเวลาที่เราพูดตัวพยางค์เสียงที่เปล่งออกมาจะแตกต่างกันทั้งนี้ขึ้นอยู่กับพยางค์เสียงที่อยู่รอบ ๆ ข้าง เช่นถ้าเราทำการบันทึกเสียงลงในเครื่องบันทึกเสียงธรรมดาเมื่อเราทำการฟังเสียงแบบ Reverse เสียงที่ได้ยินจะไม่เหมือนกับการสลับพยางค์เสียงใน TTS engine กล่าวก็คือถ้าเราบันทึกเสียงคำว่า “cat” ลงเครื่องบันทึกเสียงธรรมดาเมื่อเราฟังแบบ Reverse เสียงที่ได้ยินจะไม่ได้ยินเป็น “tak”

สาเหตุที่ใช้ Text-to-Speech

สามารถที่จะใช้ในการติดต่อสื่อสารระหว่างคนกับเครื่องโดยใช้เสียงเป็นตัวกลางในการติดต่อโดยที่ TTS จะใช้ได้ดีกว่าการ บันทึกเสียงลงฐานข้อมูลในกรณีที่

- เสียงที่ต้องทำการบันทึกมีขนาดใหญ่มากๆ ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ
- ไม่สามารถใช้กระบวนการบันทึกเสียงลงฐานข้อมูลได้ เนื่องจากตัวเครื่องคอมพิวเตอร์ไม่สามารถคาดเดาเหตุการณ์ที่จะเกิดขึ้นในอนาคตว่าจะต้องพูดอะไร ออกมาบ้าง TTS มีประโยชน์อย่างมากในกรณีที่มีการออกเสียงด้วยถ้อยคำสั้นๆหรือเมื่อไม่สามารถใช้เทคนิคแบบ Concatened ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ของ Text-to-speech

- ใช้ในการอ่าน Dynamic text ตัว TTS จะมีประโยชน์สูงเนื่องจากการอ่าน Dynamic text จะต้อง สามารถออก เสียงได้หลากหลายรูปแบบ ซึ่งวิธีการแบบ Concatened ไม่สามารถทำได้ เช่น การอ่านออก เสียงเวลา จะใช้ TTS เพราะ ถ้าเราใช้วิธี Concatened จะยุ่งยากมาก เนื่องจากจะต้องเก็บเสียงทุกเสียง ตามที่เข็มบนนาฬิกา ชี้ทั้งหมด
- ใช้ในงานพิสูจน์อักษร การใช้เสียงในการพิสูจน์อักษรและตัวเลขจะช่วยให้ผู้ใช้สามารถที่จะตรวจสอบข้อผิด พลาดที่เหลือจากการตรวจสอบด้วยตา
- ช่วยในการประหยัดเนื้อที่จัดเก็บในหน่วยความจำ เมื่อเราใช้ตัว TTS ร่วมกับไฟล์ Text จะช่วยประหยัดหน่วยความจำได้มากกว่าที่จะเก็บเป็น digital-audio ไฟล์ทั้งหมด
- ช่วยในการแจ้งสถานะของเครื่องให้ user ทราบ ตัวอย่างเช่น เมื่อผู้ใช้ทำการพิมพ์ค้างไว้แล้วไปทำงาน อย่างอื่น เมื่องานที่พิมพ์เสร็จ TTS จะช่วยแจ้ง user โดยการส่งเสียงเตือนว่า "Printing Complete" เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 สิ่งที่ควรคำนึงถึงในการพัฒนาระบบบนมือถือ

ประเด็นหลักๆ ที่ควรคำนึงถึงในการพัฒนาระบบบนมือถือ คือ ข้อจำกัดของทรัพยากร โดยแบ่งเป็นหัวข้อย่อยๆดังต่อไปนี้

- ขนาดของหน่วยความจำ ถึงแม้ว่าโทรศัพท์มือถือรุ่นใหม่ๆ มีขนาดของหน่วยความจำเพิ่มขึ้นจากเดิมมาก แต่ยังไม่สามารถรองรับการทำงานของระบบที่ซับซ้อนได้ อีกทั้งต้องคำนึงถึงขนาดข้อมูลที่เก็บ โดยเฉพาะอย่างยิ่งระบบพจนานุกรมเป็นระบบที่ต้องทำงานโดยใช้ฐานข้อมูลเป็นหลัก จำเป็นต้องจัดสรรหน่วยความจำให้เพียงพอ และทำให้ระบบสามารถทำงานได้ถูกต้อง ซึ่งในระบบนี้จะเก็บฐานข้อมูลเป็นไฟล์ข้อความแต่จะมีการบีบอัดให้มีขนาดเล็กลง

- อีกส่วนที่เกี่ยวข้องกับข้อจำกัดของทรัพยากร คือ คอมไพเลอร์ หรือ ตัวแปลภาษา ซึ่งในการพัฒนาระบบนี้ใช้เทคโนโลยีจาวา สิ่งที่ต้องคำนึงคือ จาวาบนมือถือสามารถรองรับการทำงานเกี่ยวกับระบบออกเสียงพูดเหมือนภาษามนุษย์ได้หรือไม่ จากการศึกษาองค์ประกอบของจาวาบนมือถือสำหรับ JSAPI 1.0 ยังไม่สนับสนุนการเขียนโปรแกรมเกี่ยวกับออกเสียงพูด แต่จะสนับสนุนใน JSAPI 2.0

- ส่วนการแสดงผลของหน้าจอโทรศัพท์เคลื่อนที่มีเนื้อที่จำกัด การวางตำแหน่งต่างๆควรจัดสรรให้ลงตัวพอดี สามารถใช้งานได้ง่ายและสะดวก

สุดท้ายที่ต้องคำนึงถึง คือ ระบบมีความยืดหยุ่นในการรองรับการเปลี่ยนแปลงได้มากน้อยเพียงใด ต้องออกแบบระบบฐานข้อมูลให้สามารถเพิ่ม ลด หรือ แก้ไขคำศัพท์ ได้สะดวก และเมื่อมีการเปลี่ยนแปลงฐานข้อมูลต้องไม่กระทบกับการทำงานของระบบพจนานุกรมออกเสียง

3.2 องค์ประกอบของระบบ

ในการพัฒนาระบบพจนานุกรมออกเสียงได้ประกอบด้วยส่วนหลักๆ 3 ส่วนด้วยกันคือ

1) ส่วนติดต่อต่อกับผู้ใช้ ที่เป็นส่วนรับอินพุทในการการป้อนคำ หรือ กดเมนูต่างๆ และส่วนแสดงผลออกหน้าจอ

2) ส่วนประมวลผลข้อความ เป็นส่วนที่ทำการค้นหาคำศัพท์ และคำแปลจากฐานข้อมูล

3) ส่วน TTS Engine เป็นส่วนแปลงข้อความเป็นเสียงพูด

4) ส่วนของฐานข้อมูลคำศัพท์ และคำแปล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันการทำงานของพจนานุกรมออกเสียงมีดังนี้

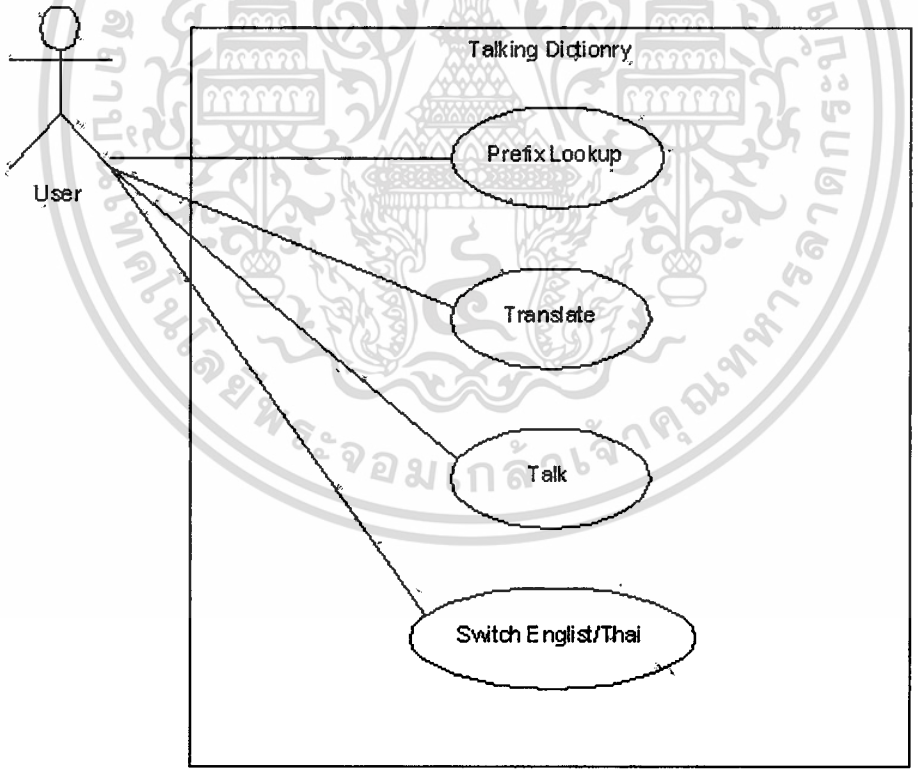
1) ค้นหาข้อมูลคำศัพท์ ผู้ใช้สามารถป้อนคำศัพท์แล้วเลือกค้นหา ระบบทำการค้นหาคำศัพท์นั้นจากฐานข้อมูล โดยการค้นหาสามารถเลือกได้ว่าเป็นการค้นหาทั้งคำ หรือ เฉพาะส่วนหน้าของคำ

2) แปลคำศัพท์ ผู้ใช้ป้อนคำศัพท์แล้วคำการเลือกแปลคำศัพท์ ระบบจะทำการค้นหาคำแปลของคำศัพท์นั้นจากฐานข้อมูล และแสดงผลออกทางหน้าจอ

3) อ่านคำศัพท์ ผู้ใช้เลือกอ่านคำศัพท์ ระบบจะทำการประมวลผลผ่าน TTS Engine ได้ผลเป็นเสียงอ่านของคำศัพท์นั้น โดยอ่านได้เฉพาะคำศัพท์ภาษาอังกฤษเท่านั้น

4) สลับรูปแบบพจนานุกรม ผู้ใช้สามารถเลือกสลับจากการแปล อังกฤษ-ไทย เป็น ไทย-อังกฤษ ได้

จากฟังก์ชันที่กล่าวข้างต้นสามารถเขียนเป็น Use Case ได้ดังนี้



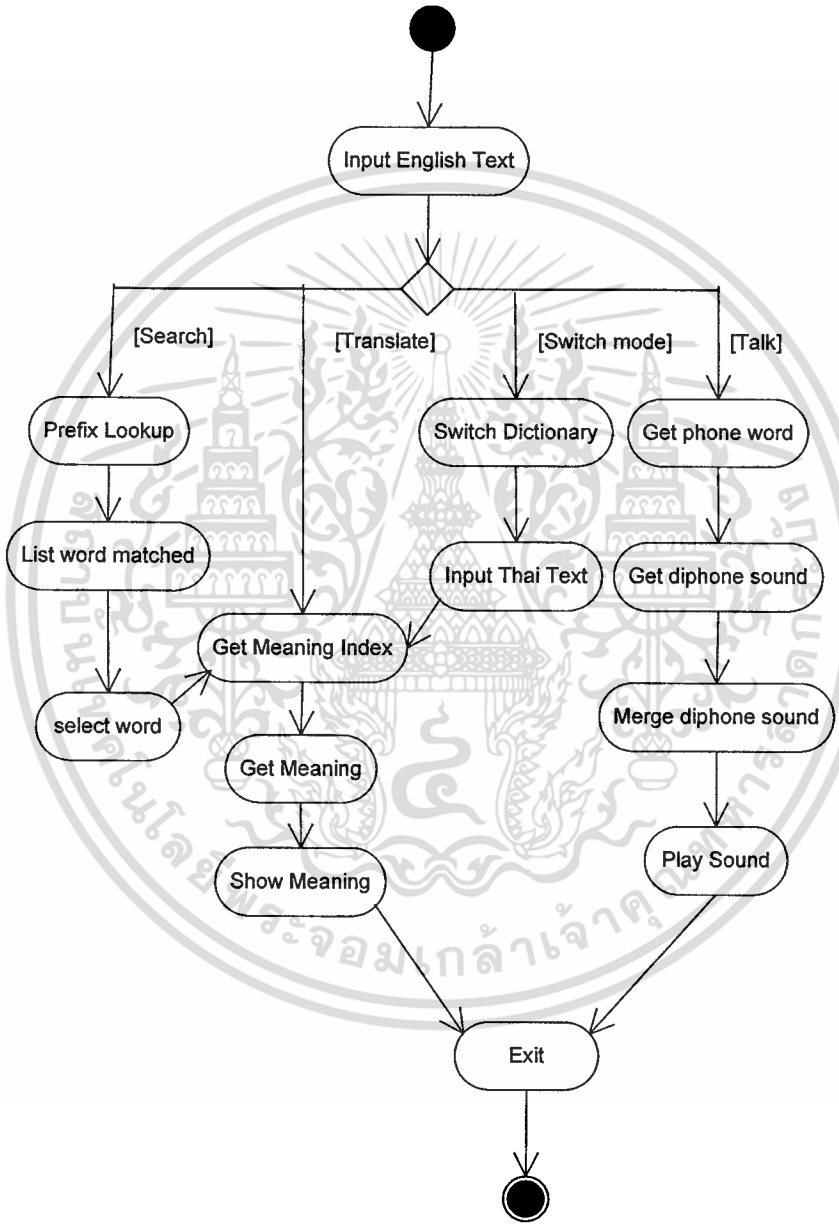
รูปที่ 3.1 Use-Case Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

3.3 Activity Diagram

เพื่ออธิบายให้เข้าใจการทำงานของระบบ Talking Dictionary เราสามารถเขียนขั้นตอนและกิจกรรมที่เกิดขึ้นภายในระบบได้ดังนี้



รูปที่ 3.2 Activity Diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 สามารถอธิบายการทำงานของระบบได้ดังนี้

1) Input English Text เป็นการป้อนคลงในช่องกรอกข้อความ ซึ่งค่าเริ่มต้นจะเป็นการป้อนคำภาษาอังกฤษ

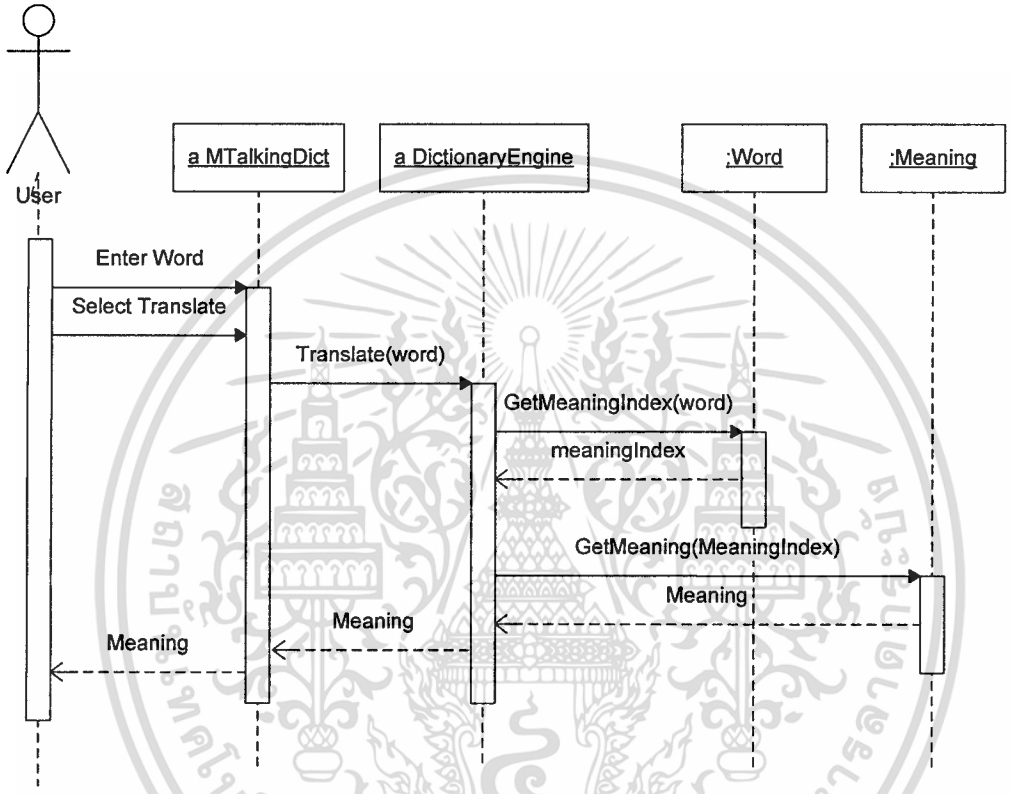
2) เมื่อป้อนคำเสร็จแล้วสามารถเลือกทำงานได้หลายแบบได้แก่

- Translate เป็นการเลือกแปลความหมายของคำที่ป้อน โดยจะเริ่มจาการหาค่าอินเด็กซ์ของคำศัพท์นั้นก่อน จากนั้นใช้อินเด็กซ์ไปค้นหาความหมาย เมื่อได้ความหมายจะทำการแสดงความหมายของคำๆนั้นออกมาทางหน้าจอ
- Prefix Lookup เมื่อมีการป้อนคำและเลือกการทำงานส่วนนี้จะมีการแสดงรายการคำที่ตรงตามเงื่อนไขที่ต้องการค้นหา จากนั้นเมื่อมีการเลือกคำในรายการที่ต้องการหาความหมาย นั้นคือไปยังขั้นตอนของการทำงานของส่วน Translate นั้นเอง
- Switch Dictionary ส่วนนี้เป็นการสลับโหมดการทำงานไปมาระหว่างการแปลอังกฤษ-ไทย และ ไทย-อังกฤษ ซึ่งจะมีการทำงานต่างกันคือ โหมดอังกฤษ-ไทย เป็นการป้อนคำภาษาอังกฤษเหมือนกับตอนเข้าโปรแกรมมาตอนแรก แต่การทำงานของโหมด ไทย-อังกฤษ จะเป็นการเลือกตัวอักษรครั้งละตัวจากตาราง มาประกอบเป็นคำ จากนั้นสามารถแปลงเป็นคำภาษาอังกฤษในลักษณะกลับกันกับโหมด อังฤษ-ไทย
- Talk เป็นการออกเสียงอ่านคำที่ป้อนไว้ โดยในส่วนการทำงานนี้จะสามารถทำได้เฉพาะคำภาษาอังกฤษเท่านั้น โดยเริ่มจากแปลคำศัพท์ปกติเป็นคำพยางค์เสียง ซึ่งแต่ละพยางค์เสียงจะมีไฟล์เสียงก็บไว้แล้ว ทำการไฟล์เสียงแต่ละพยางค์มารวมกันให้เสียงมีความราบเรียบมากขึ้น จากนั้นจึงเปล่งเสียงเป็นคำอ่านศัพท์นั้นๆออกมา

3) Exit สามารถออกจากการทำงานได้เกือบตลอดเวลา ไม่เฉพาะว่าต้องอยู่ในการทำงานส่วนใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

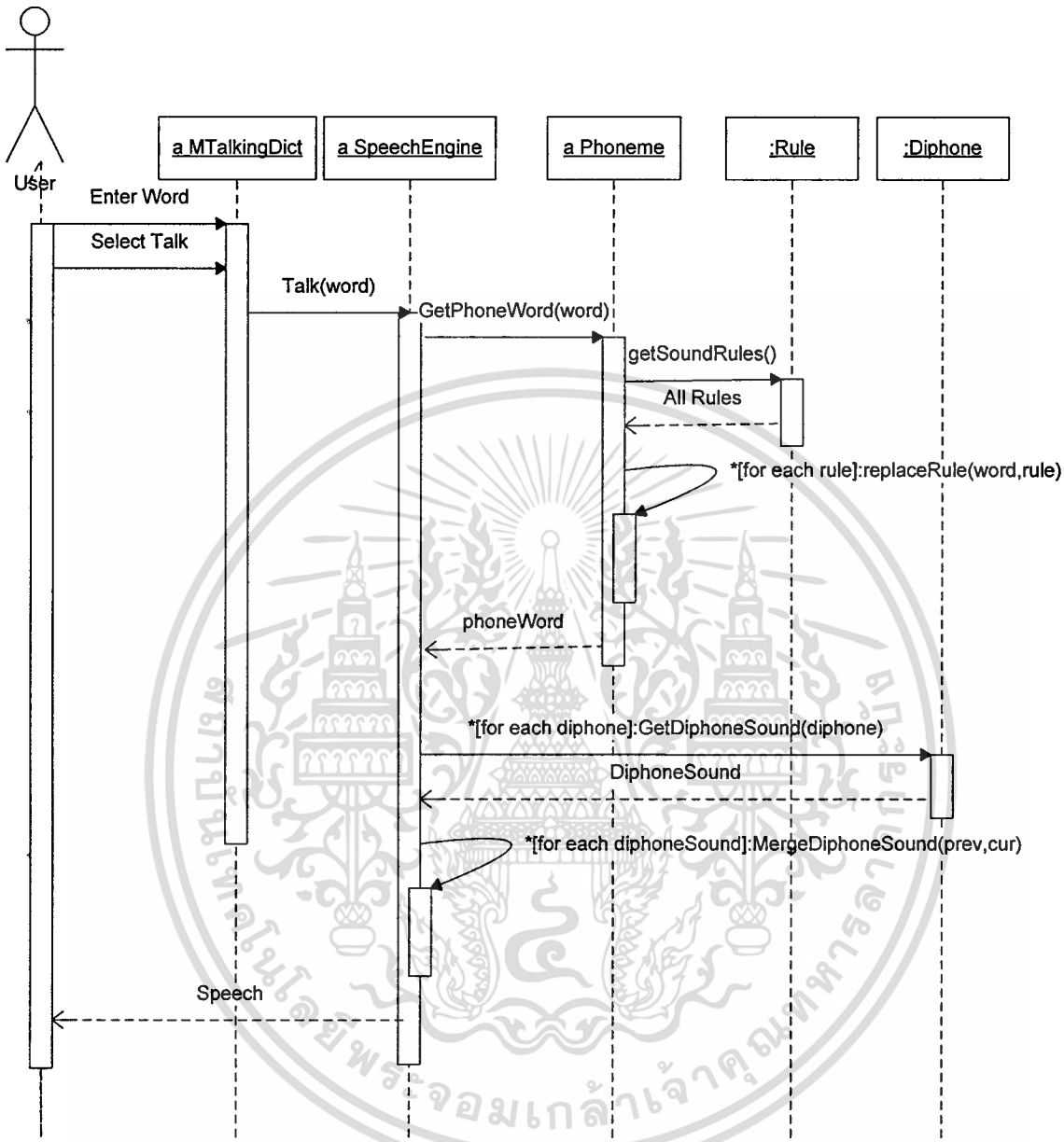
3.4 Sequence Diagram



รูปที่ 3.3 Sequence diagram ของการแปลความหมาย

จากรูปที่ 3.3 เป็นการทำงานในส่วนของการแปลความหมายคำศัพท์ เมื่อผู้ใช้ป้อนคำศัพท์และเลือกเมนู Translate คำศัพท์จะถูกส่งไปยัง DictionaryEngine เพื่อทำการค้นหาคำศัพท์จากฐานข้อมูล โดยจะได้ค่าอินเด็กซ์มาจากไฟล์คำศัพท์ก่อน จากนั้นนำค่าอินเด็กซ์ไปค้นหาความหมายจากไฟล์คำแปลอีกครั้ง เมื่อพบความหมายจากส่งคืนกลับมาเพื่อแสดงผลให้ผู้ใช้เห็น

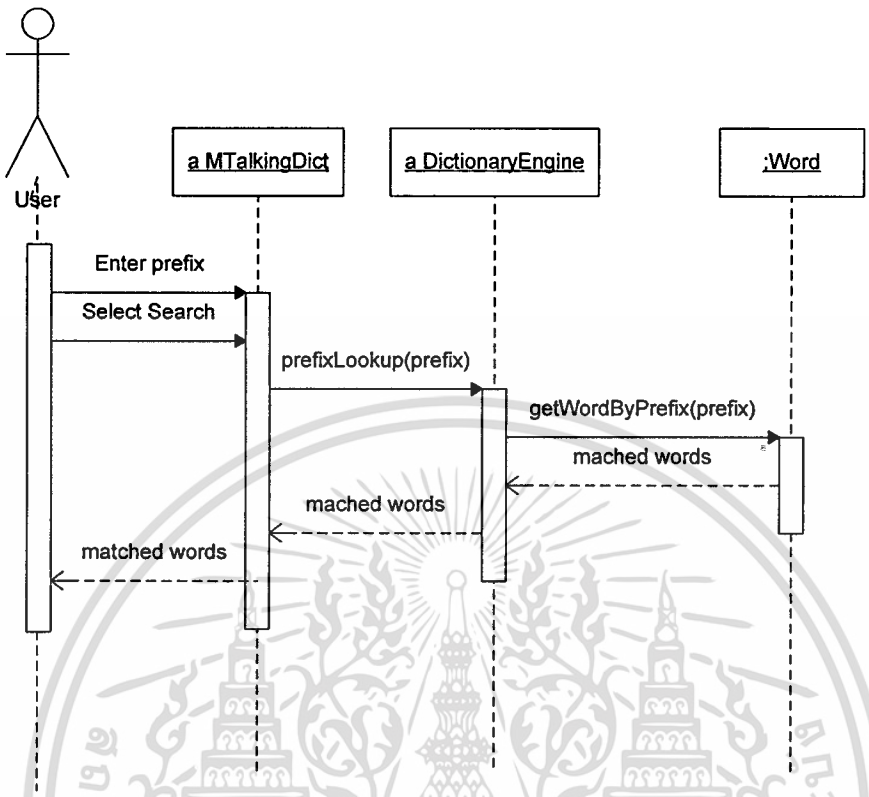
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Sequence Diagram ของ Talk

จากรูปที่ 3.4 เป็นการทำงานในส่วนของการอ่านคำศัพท์ โดยเมื่อผู้ใช้ป้อนคำศัพท์และเลือก talk จะมีการเรียกใช้งานฟังก์ชัน Talk() จาก SpeechEngine ซึ่งเริ่มต้นแปลงคำศัพท์เป็นข้อมูลพยางค์เสียงจากนั้นจะดึงเสียงของแต่ละพยางค์มาทำการรวมเสียงระหว่างพยางค์เพื่อให้เสียงมีความราบเรียบมากขึ้น เมื่อทำการรวมทุกพยางค์เข้าด้วยกันแล้ว จะทำการเปล่งเสียงออกมาเป็นคำอ่านของคำศัพท์นั้นๆ

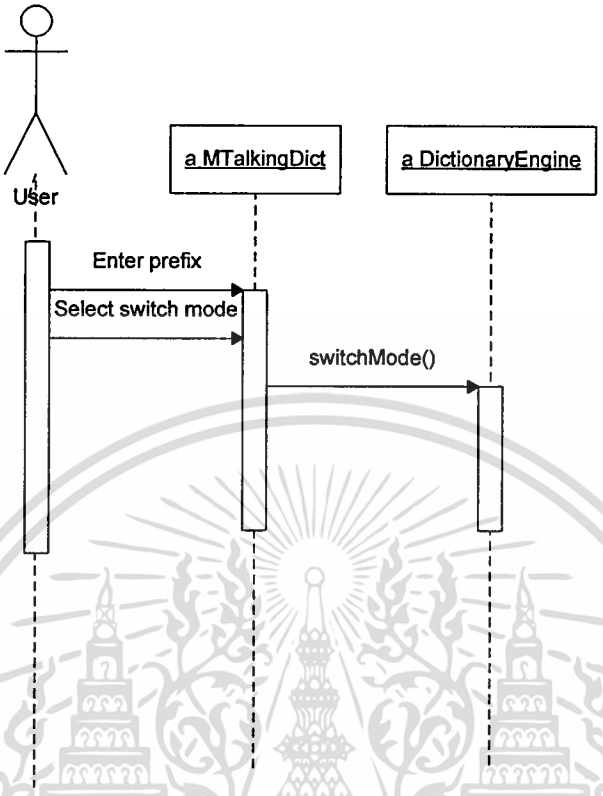
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Sequence Diagram ของการค้นหาคำศัพท์

จากรูปที่ 3.5 เป็นการค้นหาคำศัพท์ด้วย Prefix เมื่อผู้ใช้ป้อนคำศัพท์และเลือกเมนู Prefix Lookup จะมีการเรียนกใช้งาน prefixLookup() ของคลาส DictionaryEngine โดยจะทำการแสดงรายการทั้งหมดที่ตรงตามเงื่อนไขที่ผู้ใช้ป้อนเข้ามาให้ผู้ใช้ได้ทำการเลือกเพื่อทำงานในฟังก์ชั่นอื่นๆต่อไป เช่น เลือก translate เพื่อแปลความหมาย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

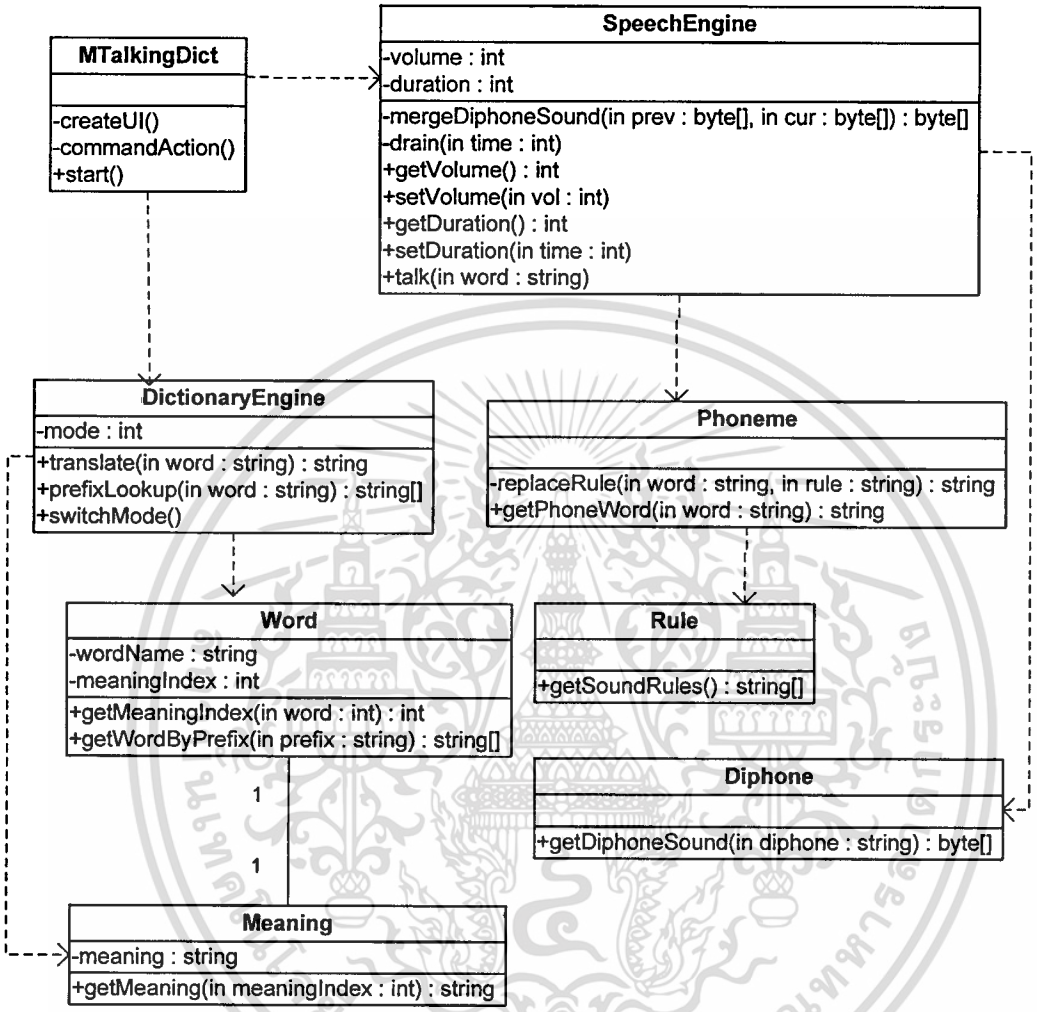


รูปที่ 3.6 Sequence Diagram ของการเปลี่ยนโหมดพจนานุกรม

จากรูปที่ 3.6 เป็นการทำงานในส่วนการเปลี่ยนโหมดของพจนานุกรม โดยเริ่มต้นพจนานุกรมจะทำงานในโหมดอังกฤษ-ไทย เมื่อผู้ใช้เลือกเปลี่ยนโหมด DictionaryEngine จะการสลับเป็นโหมดไทย-อังฤษแทน การเปลี่ยนโหมดสามารถทำสลับไปมาได้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Class Diagram



รูปที่ 3.7 Class Diagram ของระบบ

Class: MTalkingDict

เป็นคลาสหลักที่ควบคุมการทำงานของคลาสอื่นๆ โดยมีโอเปอเรชั่นหลักๆคือ

- Start() การเริ่มต้นทำงานของโปรแกรมจะเริ่มจากตรงนี้ ซึ่งจะมีการกำหนดค่าเริ่มต้นของระบบ และจะเรียกใช้งาน CreateUI()
- CreateUI() เป็นส่วนที่สร้างส่วนติดต่อกับผู้ใช้ ไม่ว่าจะเป็นส่วนของการรับอินพุทคำศัพท์ เมนูต่างๆให้ผู้ใช้เลือกใช้งาน และเตรียมในส่วนของการแสดงผล
- CommandAction() เป็นโอเปอเรชันที่สำคัญอีกส่วนหนึ่ง เพราะจะคอยควบคุมการทำงานของโปรแกรม เมื่อผู้ใช้มีการเลือกใช้งานเมนูใด ส่วนนี้จะคอยรับคำสั่ง และเรียกใช้งานคลาสอื่นๆภายในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class: DictionaryEngine

เป็นคลาสที่ทำงานในส่วนของพจนานุกรมเป็นหลักโดยมีแอตริบิวและโอเปอเรชันดังต่อไปนี้

- Mode เป็นแอตริบิวที่เก็บค่าโหมดการทำงานของส่วนพจนานุกรม โดยค่าเริ่มต้นจะเป็น 0 หมายถึงทำงานในหมด อังกฤษ-ไทย และจะมีการทำงานอีกโหมดคือ ไทย-อังกฤษ ซึ่งจะมีค่าเป็น 1
- Translate() เป็นโอเปอเรชันการแปลความหมายโดยรับพารามิเตอร์เป็นคำศัพท์และให้ค่ากลับมาเป็นคำแปล โดยจะมีการเรียกใช้งาน getMeaningIndex() ของคลาส word และ getMeaning() ของคลาส Meaning
- PrefixLookup() เป็นโอเปอเรชั่นในการค้นหาคำโดยรับพารามิเตอร์เป็นส่วนหนึ่งของคำและให้ค่ากลับมาเป็นกลุ่มของคำตรงตามเงื่อนไข โดยจะมีการเรียกใช้งาน getWordByPrefix() ของคลาส word

Class: Word

เป็นคลาสที่เกี่ยวข้องกับคำศัพท์โดยมีแอตริบิวและโอเปอเรชั่นดังต่อไปนี้

- WordName เป็นคำศัพท์ทั้งภาษาไทย และภาษาอังกฤษ
- MeaningIndex เป็นอินเด็กซ์ซึ่งเป็นค่าตัวเลขสำหรับชี้ไปยังตำแหน่งของคำแปล
- GetMeaningIndex() รับพารามิเตอร์เป็นคำศัพท์และให้ค่ากลับมาเป็นค่าอินเด็กซ์ที่ชี้ไปยังคำแปล
- GetWordByPrefix() รับพารามิเตอร์เป็นกลุ่มตัวอักษร และให้ค่ากลับเป็นกลุ่มของคำที่มีคำนำหน้าตรงกับกลุ่มของตัวอักษรที่ใส่มาในพารามิเตอร์

Class: Meaning

เป็นคลาสที่เกี่ยวข้อกับคำแปลโดยมีแอตริบิวแลโอเปอเรชั่นดังต่อไปนี้

- Meaning คือ คำแปลของคำศัพท์แต่ละคำ
- getMeaning() เป็นโอเปอเรชั่นดึงค่าคำแปลโดยรับพารามิเตอร์เป็นค่าอินเด็กซ์และรีเทิร์นค่าเป็นคำแปล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class: SpeechEngine

เป็นคลาสหลักที่ทำงานในการแปลคำศัพท์ออกมาเป็นเสียงอ่านโดยมีแอตริบิวและโอเปอเรชั่นดังต่อไปนี้

- Volume คือดังของการออกเสียง
- Duration คือความเร็วในการออกเสียงในแต่ละพยางค์
- getVolume() เป็นการอ่านค่าความดังของการออกเสียง
- setVolume() เป็นการกำหนดค่าความดังของการออกเสียง
- getDuration() เป็นการอ่านค่าความเร็วในการออกเสียงแต่ละพยางค์
- setDuration() เป็นการกำหนดค่าความเร็วในการออกเสียงแตละพยางค์
- drain() เป็นการกำหนดช่วงเงียบซึ่งใช้ในการตัดเสียงระหว่างพยางค์โดยรับพารามิเตอร์เป็นวินาที
- MergeDiphoneSound() เป็นการรวมเสียงของแต่ละพยางค์ให้มีความรับเรียบมากยิ่งขึ้นโดยการนำพยางค์เสียงก่อนหน้าบางส่วนมารวมกับพยางค์เสียงปัจจุบันบางส่วน
- Talk() เป็นโอเปอเรเตอร์ที่ควบคุมการทำงานให้ออกเสียงมาเป็นคำอ่านของคำศัพท์หนึ่งคำ โดยรับพารามิเตอร์เป็นคำศัพท์ จากนั้นจะแปลงคำศัพท์เป็นคำพยางค์เสียงโดยการเรียกใช้ getPhoneWord() ของคลาส Phoneme จากนั้นนำแต่ละพยางค์เสียงมาทำการดึงไบต์เสียงโดยเรียกใช้ getDiphoneSound() ของคลาส Diphone เมื่อได้ไบต์เสียงแต่ละพยางค์มาแล้วทำการรวมเสียงระหว่างพยางค์ด้วยการเรียกใช้ MergeDiphoneSound() จากนั้นทำการเล่นไฟล์เสียงที่ได้จากการรวม ออกมาเป็นเสียงอ่านคำศัพท์

Class: Phoneme

เป็นคลาสที่เกียวกับพยางค์เสียงของคำศัพท์โดยมีโอเปอเรชั่นคือ

- ReplaceRule() เป็นโอเปอเรชั่นที่ใช้สำหรับแปลคำศัพท์ให้เป็นพยางค์เสียงโดยกฎต่างๆในการแปลง
- getPhoneWord() เป็นโอเปอเรชั่นที่ใช้ในการคำศัพท์เป็นคำที่เป็นพยางค์เสียงซึ่งจะคั่นพยางค์เสียงด้วยตัวคั่นที่ชัดเจน โดยรับพารามิเตอร์เป็นคำศัพท์ และรีเทิร์นค่าเป็นคำอ่านโดยจะมีการเรียกใช้ GetSoundRule() จากคลาส Rule เพื่อนำกฎมาใช้ในการแปลงคำศัพท์โดยการเรียกใช้ ReplaceRule()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Class: Rule

เป็นคลาสที่เกียวกับกฎของการออกเสียงในการอ่านคำต่างๆ โดยมีโอเปอเรชั่นคือ

- GetSoundRule() เป็นโอเปอเรชั่นที่จะรีเทิร์นกฎทั้งหมดในการออกเสียงซึ่งใช้สำหรับการแปลคำศัพท์ออกเป็นคำอ่าน

Class: Diphone

เป็นคลาสที่เกียวกับไฟล์เสียงของพยางค์ต่างๆที่ใช้ในการอ่านคำ โดยมีโอเปอเรชั่น คือ

- GetDiphoneSound() เป็นโอเปอเรชั่นที่ใช้ในการดึงค่าของไฟล์เสียงโดยจะรับพารามิเตอร์เป็นพยางค์เสียงเดี่ยว (diphone) ซึ่งพยางค์เสียงแต่ละตัวจะมีไฟล์เสียงที่เก็บไว้ในฐานข้อมูล ทำการอ่านไฟล์เสียงนั้นออกมาเป็นไบต์ข้อมูลตามขนาดของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การพัฒนาระบบ

4.1 ขั้นตอนการพัฒนาระบบ

1. เตรียมส่วนฐานข้อมูลที่จะเป็นไฟล์คำศัพท์ และ คำแปล ซึ่งเป็นส่วนสำคัญในการทำงานของระบบพจนานุกรม ทำการจัดเป็นหมวดหมู่ให้เหมาะสม และสะดวกกับการใช้งาน
2. เตรียมเครื่องมือที่ใช้ในการพัฒนาระบบบนโทรศัพท์เคลื่อนที่ด้วย J2ME
3. พัฒนาในส่วนของการแสดงผล และรับข้อมูลจากผู้ใช้ผ่านทางหน้าจอโทรศัพท์เคลื่อนที่
4. พัฒนาส่วนติดต่อกับฐานข้อมูล
5. พัฒนาส่วน TTS Engine
6. พัฒนาส่วนพจนานุกรมในส่วนการค้นหาคำศัพท์ และ คำแปล
7. ทดสอบระบบ

4.2 ฐานข้อมูล

ฐานข้อมูลของพจนานุกรมเป็นคำศัพท์จากพจนานุกรม Lexitron ที่พัฒนาโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ นำมาจัดให้สามารถใช้งานได้สะดวกโดยแบ่งไฟล์เป็นสองประเภท คือ

1. ไฟล์คำศัพท์ เป็นไฟล์ที่เก็บคำศัพท์ต่างๆเก็บเรียงคำละบรรทัด แต่ละบรรทัดประกอบด้วยคำศัพท์ตามด้วยตำแหน่งของคำแปลในไฟล์คำแปล
2. ไฟล์คำแปล เป็นไฟล์ที่เก็บความหมายของคำศัพท์ซึ่งจะอยู่ในตำแหน่งที่สอดคล้องกับค่าที่ใส่ไว้ในไฟล์คำศัพท์

เพื่อความสะดวกในการใช้งานจะแยกไฟล์สำหรับแปลอังกฤษ-ไทย และ ไทย-อังกฤษออกจากกัน โดยใช้ชื่อไฟล์ที่แตกต่างกันคือ et และ te ตามลำดับ โดยไฟล์คำศัพท์ซึ่งเป็นไฟล์อินเด็กซ์ชี้ไปยังคำแปลจะใช้เป็น eti หรือ tei ไฟล์คำแปลจะใช้เป็น etd หรือ ted และเพื่อเพิ่มประสิทธิภาพในการค้นหาคำศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำแปล จึงได้แยกไฟล์คำศัพท์ และไฟล์คำแปลออกเป็นไฟล์ย่อยๆ โดยใช้ตัวเลขกำกับตามรหัส ascii ของอักษรตัวแรกของคำศัพท์แต่ละตัว ดังตัวอย่าง

ตารางที่ 4.1 ชื่อไฟล์ที่สัมพันธ์กันระหว่างไฟล์คำศัพท์และคำแปลแบบไทย-อังกฤษ

ไฟล์คำศัพท์	ไฟล์คำแปล
tei161.t	ted161.t
tei162.t	ted162.t
tei163.t	ted163.t
…	…
tei228.t	ted228.t

ตารางที่ 4.2 ชื่อไฟล์ที่สัมพันธ์กันระหว่างไฟล์คำศัพท์และคำแปลแบบอังกฤษ-ไทย

ไฟล์คำศัพท์	ไฟล์คำแปล
eti97.t	etd97.t
eti98.t	etd98.t
eti99.t	etd99.t
…	…
eti122	etd122.t

จากตารางที่ 4.1 จะเห็นว่าไฟล์ไทย- อังกฤษจะเริ่มต้นจาก eti161.t – eti228.t นั่นคือรหัส ascii ของตัวอักษร 'ก' – 'ไ' เช่นเดียวกัน ตารางที่ 4.2 ไฟล์อังฤษ-ไทยเริ่มต้นจาก eti97.t – eti122.t นั่นคือรหัส ascii ของอักษณ 'a' – 'z' ในการจับคู่กันระหว่างคำศัพท์และคำแปลนั้นจะใช้ชื่อไฟล์ที่สัมพันธ์กันโดยตัวเลขที่ระบุอยู่ท้ายชื่อไฟล์คำศัพท์และไฟล์คำแปลจะตรงกัน โดยข้อมูลภายในไฟล์คำศัพท์ และไฟล์คำแปลจะเป็นดังตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ความสัมพันธ์ของข้อมูลในไฟล์คำศัพท์ และ คำแปล

ตัวอย่างคำศัพท์ในไฟล์ eti97.t	ตัวอย่างคำแปลในไฟล์ etd97.t
a|0	(DET) หนึ่ง (คำนำหน้าคำนามเพื่อแสดงว่าคำนามนั้นๆ ไม่ชี้เฉพาะ)
a|66	(N) อักษรตัวแรกในภาษาอังกฤษ
a-|94	(PRF) ของ
a-|104	(PRF) จาก
a-|114	(PRF) ใช้เป็นคำเติมหน้าเพื่อเน้นความหมาย
...	...
azure|88818	(N) ฟ้าเหมือนท้องฟ้า

จากตาราง 4.3 จะเห็นว่าคำศัพท์ 'a' ในไฟล์ eti97.t คั่นด้วย '|' แล้วตามด้วยตัวเลข 0 หมายถึงมีความหมายอยู่ในตำแหน่งที่ 0 ของไฟล์คำแปล โดยความหมายของ a มีขนาด 65 ตัวอักษร ทำให้คำศัพท์ตัวถัดมาซึ่งเป็น 'a' เช่นกันจะมีตัวเลขชี้ตำแหน่งคำแปลเป็นตัวอักษรที่ 66 ซึ่งต่อจากความหมายของคำแรกนั่นเอง คำถัดๆมาจะเป็นลักษณะนี้จนกระทั่งจบไฟล์ที่คำศัพท์สุดท้ายของตัวอักษร 'a' คือคำว่า 'azure'

4.3 ส่วนพจนานุกรม

ส่วนนี้เป็นส่วนสำคัญที่ทำงานโดยใช้ฐานข้อมูลเป็นหลัก โดยสามารถแบ่งเป็นส่วนย่อยๆได้ดังนี้

- ส่วนติดต่อฐานข้อมูล ส่วนนี้จะเป็นตัวจัดการในการค้นหาคำศัพท์จากฐานข้อมูล และทำการดึงข้อมูลที่ได้มาใช้งาน เนื่องจากระบบนี้ไม่มีการบันทึกข้อมูลลงฐานข้อมูลจึงเป็นการอ่านข้อมูลเพียงอย่างดียว แต่จะมีรายละเอียดในการเข้าถึงข้อมูลคือต้องมีค้นหาตัวอักษรตามรหัส ascii เพื่อได้ชุดข้อมูลที่ต้องการ จากนั้นจะเข้าไปค้นหาคำศัพท์ภายในชุดข้อมูลนั้นอีกครั้ง ก็จะได้คำศัพท์ที่ต้องการ ในกรณีที่เป็นการดึงข้อมูลคำแปลง จะใช้ตัวชี้ที่อยู่หลังคำศัพท์ เพื่อไปดึงข้อมูลคำแปล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนแสดงผล ส่วนนี้อาจมองแยกเป็นอินพุท และเอาพุท ซึ่งเป็นการทำงานกับหน้าจอของโทรศัพท์เคลื่อนที่ จะต้องมีการจัดตำแหน่งให้พอเหมาะและใช้งานง่าย เนื่องจากหน้าจอการแสดงผลของโทรศัพท์เคลื่อนที่มีข้อจำกัดพอสมควร ลำดับการทำงานจึงควรจัดให้สามารถใช้งานได้สะดวก ในส่วนของอินพุทที่ใช้งานเช่น การป้อนคำศัพท์ ซึ่งสามารถรองรับการป้อนคำศัพท์ได้ทั้งภาษาไทยและภาษาอังกฤษ สำหรับภาษาอังกฤษเป็นการป้องคำลงในช่องรับข้อมูลธรรมดา ส่วนการป้อนภาษาไทย จำเป็นต้องใช้ตารางตัวอักษรให้ผู้ใช้เลือกป้อนทีละตัว ในการแสดงผลเป็นคำแสดงผลคำแปล หรือ แสดงผลรายการคำในกรณีที่มีการค้นหาคำด้วยคำขึ้นต้น
- ส่วนการควบคุมการทำงานของโปรแกรม เป็นส่วนสำคัญอีกส่วนหนึ่งในการควบคุมขั้นตอนการทำงานเมื่อได้รับอินพุท หรือ คำสั่งจากผู้ใช้จะต้องไปทำงานในส่วนใด เปรียบเสมือนเป็นตัวเชื่อมการทำงานของแต่ละส่วนเข้าด้วยกัน ตั้งแต่เริ่มใช้งานระบบ จนกระทั่งสิ้นสุดการใช้งานระบบ เช่น เมื่อมีการเปลี่ยนจาก อังกฤษ-ไทย เป็น ไทย-อังกฤษ จะต้องมีการเปลี่ยนวิธีการอินพุทจากการป้อนตัวอักษรภาษาอังกฤษเป็นการป้อนตัวอักษรภาษาไทยแทน หรือหากผู้ใช้งานเลือกคำสั่ง 'talk' จะต้องไปเรียก TTS Engine ให้อ่านออกเสียงคำศัพท์นั้นๆ เป็นต้น

4.4 TTS Engine

การพัฒนา TTS Engine แบ่งเป็นสองส่วนหลักๆ คือ

1. Text Processor

ทำการแบ่งคำศัพท์ออกเป็นพยางค์เสียง เพื่อให้สามารถอ่านออกเสียงได้ โดยใช้อัลกอริทึมในการแยกพยางค์หลายๆแบบ ยกตัวอย่างเช่น คำศัพท์ 'hello' ตัวอักษร "h" ใน hello จะกลายเป็นตัวสร้างพยางค์เสียง "h" ตัวอักษร "e" จะสร้างพยางค์เสียงที่ออกเสียงเป็น "eh" ตัวอักษร "l" ตัวแรกจะสร้างพยางค์เสียงที่ออกเสียงเป็น "l" ส่วนตัวอักษร "l" ที่สองไม่ต้องทำการสร้างพยางค์เสียง สุดท้ายตัวอักษร "o" จะใช้ในการสร้างพยางค์เสียง "oe" ซึ่งจะได้พยางค์เสียงทั้งคำเป็น "h | eh | l | oe" และจะทำการส่งพยางค์เสียงนี้ไปยัง ส่วน Speech Synthesis เพื่ออ่านออกเสียงอีกต่อหนึ่ง

2. Speech Synthesis

ในส่วนนี้จะทำการแปลงพยางค์ที่แบ่งไว้ออกเป็นเสียง เมื่อออกเสียงครบทุกๆพยางค์จะได้เป็นเสียงอ่านของคำศัพท์นั้นๆ เนื่องจากการออกเสียงแยกแต่ละพยางค์ทำให้เสียงที่ออกมาไม่ราบเรียบต่อเนื่องเท่าเสียงมนุษย์ จึงใช้เทคนิคการต่อพยางค์ให้มีการกลมกลืนมากขึ้น ด้วยการอินเตอร์เซกชันขอบเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสองพยางค์ที่ติดกันให้มีระดับเสียงเดียวกัน เช่นคำว่า "hello" ที่แยกออกเป็นพยางค์เสียง "h|eh|l|oe" ตามกฎของการออกเสียง จะมีไฟล์เสียงค์ที่ตรงตามพยางค์คือ h.wav eh.wav l.wav และ oe.wav จากนั้นจะทำการอ่านไฟล์เสียงแต่ละพยางค์ออกมาเป็นไบต์ข้อมูล ทำการรวมพยางค์เสียงที่ติดกันโดยเอาไบต์ข้อมูลของ h.wav ส่วนท้ายมาต่อกับไบต์ข้อมูลส่วนหน้าของ eh.wav จะได้เป็นไบต์ข้อมูลใหม่สำหรับพยางค์เสียง h-eh เมื่อทำการรวมเสียงครบทั้งคำ จะได้ไบต์ข้อมูลของพยางค์เสียง h h-eh eh eh-l l l-oe oe และทำการเล่นเสียงทุกพยางค์ออกมาเป็นคำอ่านคำศัพท์ทั้งคำ

4.5 เครื่องมือที่ใช้ในการพัฒนา

การพัฒนาระบบบนโทรศัพท์เคลื่อนที่มีข้อแตกต่างจากการพัฒนาระบบบนคอมพิวเตอร์ทั่วไป เพื่อความสะดวกจึงใช้เครื่องมือเข้ามาช่วยในการพัฒนา โดยในการพัฒนาระบบนี้มีการใช้เครื่องมือ 2 ตัวด้วยกัน คือ

1. J2ME Wireless Toolkit เป็นเครื่องมือของ Sun Microsystems ที่สามารถดาวน์โหลดมาใช้งานได้จากอินเตอร์เน็ต ใช้สำหรับคอมไพล์จาวาโค้ดเป็น .class และยังสามารถรันโปรแกรมบนโทรศัพท์เคลื่อนที่จำลองที่มีให้มากับเครื่องมือตัวนี้ได้
2. Nokia Developer Suite เป็นเครื่องมือทางโนเกียซึ่งเป็นบริษัทผลิตมือถือยักษ์ใหญ่ของโลกบริษัทหนึ่ง สามารถดาวน์โหลดมาใช้งานได้จากอินเตอร์เน็ต เช่น สามารถทำงานได้เช่นเดียวกัน J2ME Wireless Toolkit แต่จะมีข้อได้เปรียบถือ สามารถรันใช้งานบนโทรศัพท์เคลื่อนที่จำลองได้หลายรุ่น โดยเฉพาะโทรศัพท์เคลื่อนที่ของโนเกียเอง ทำให้สะดวกในการทดสอบโปรแกรมก่อนจะโหลดนำเข้าไปใช้งานบนโทรศัพท์เคลื่อนที่จริงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 หน้าจอระบบ

รูปที่ 4.1 หน้าแรกของระบบ MtalkingDict

จากรูปที่ 4.1 เป็นหน้าจอแรกก่อนเข้าใช้งานระบบ MtalkingDict ซึ่งเป็นชื่อของแอพลิเคชั่นที่พัฒนาขึ้นมาใช้งานบนโทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 การป้อนคำลงใน Textbox

จากรูปที่ 4.2 เมื่อเข้ามาในโปรแกรมในส่วนแรกจะให้ผู้ใช้ป้อนคำศัพท์ที่ต้องการลงใน Textbox โดยเริ่มต้นจะเป็นการป้อนคำภาษาอังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 การ clear คำที่ป้อนไว้

จากรูปที่ 4.3 หากต้องการลบคำที่ป้อนไว้ให้เลือกที่ option จะปรากฏเมนูขึ้นมาเลือกตรงเมนู 'clear' ข้อความที่เราป้อนไว้จะถูกลบเพื่อให้เราสามารถป้อนข้อความใหม่ลงไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 การอ่านคำศัพท์

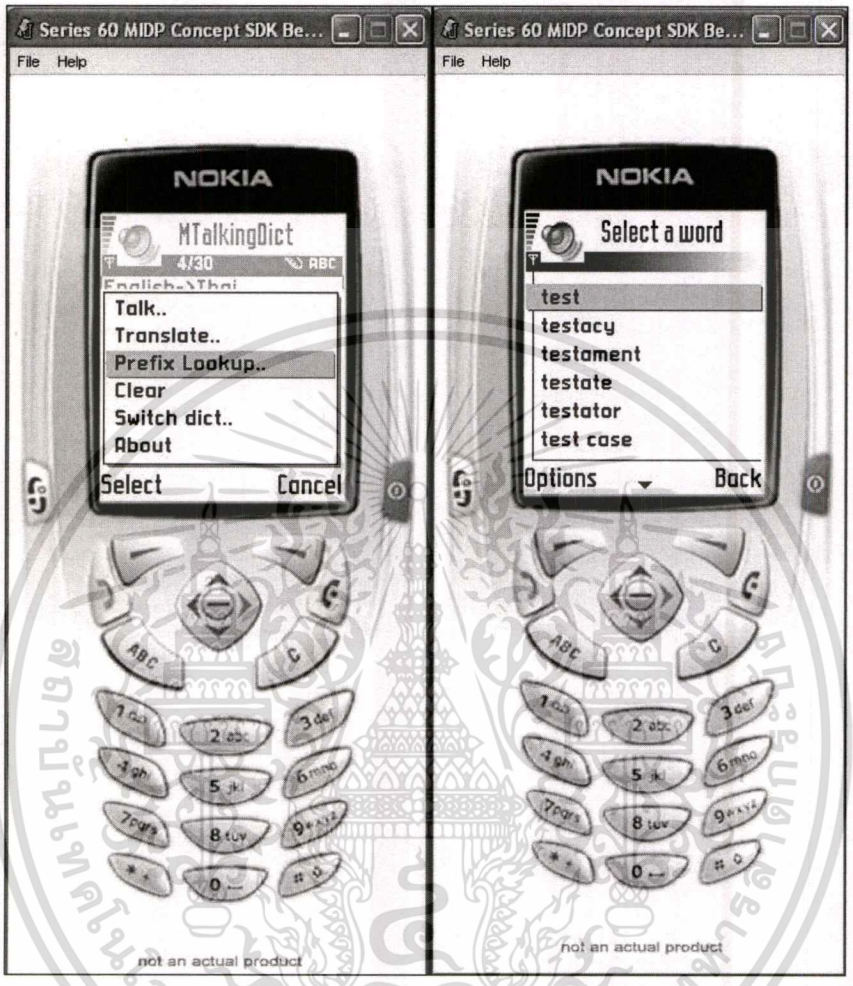
จากรูปที่ 4.4 เมื่อป้อนคำศัพท์ที่ต้องการเรียบร้อยแล้วทำการกดปุ่ม option และเลือกเมนู 'Talk' ระบบจะทำการอ่านออกเสียงคำที่เราป้อนไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 การแปลความหมายคำศัพท์

จากรูปที่ 4.5 เมื่อเราป้อนคำศัพท์คำว่า 'test' แล้วต้องการหาความหมายให้เลือก option แล้วเลือกเมนู 'Translate' ระบบจะทำการค้นหาความหมายและแสดงผลออกมาทางหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การค้นหาคำด้วย prefix

จากรูปที่ 4.6 เป็นการค้นหาคำจาก prefix ที่ใส่ โดยใส่เป็น 'test' เมื่อเลือก 'prefix lookup …' จะปรากฏคำที่ขึ้นต้นด้วยคำว่า 'test' ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 การ switch dictionary

จากรูปที่ 4.7 เป็นการสลับโหมดของพจนานุกรมจากอังกฤษ-ไทย เป็น ไทย-อังกฤษซึ่งจะปรากฏตารางตัวอักษรภาษาไทยให้เลือกป้อนคำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.8 การแปลศัพท์ภาษาไทยเป็นอังกฤษ

จากรูปที่ 4.8 เป็นการแปลศัพท์ภาษาไทยเป็นภาษาอังกฤษ ซึ่งการใช้งานเป็นลักษณะเดียวการภาษาอังกฤษเป็นภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป และข้อเสนอแนะ

5.1 สรุป

การพัฒนาระบบต่างๆบนโทรศัพท์เคลื่อนที่ด้วยภาษาจาวา จำเป็นต้องเข้าใจถึงสถาปัตยกรรมของอุปกรณ์ขนาดเล็กที่มีทรัพยากรที่จำกัด ไม่สามารถสนับสนุนการทำงานได้ครบถ้วนทั้งหมด แต่โทรศัพท์เคลื่อนที่เป็นอุปกรณ์อิเล็คทรอนิกส์ที่มีการเปลี่ยนแปลงเทคโนโลยีใหม่ๆเข้ามาเพิ่มเติมอยู่เสมอ จึงควรติดตามศึกษาอย่างใกล้ชิด ในการจะพัฒนาระบบใหม่บนโทรศัพท์เคลื่อนที่ใหม่ๆขึ้นมาได้นั้นต้องอาศัยความร่วมมือของทั้งผู้ผลิตตัวอุปกรณ์ และผู้สนับสนุนด้านภาษาโปรแกรมที่ใช้พัฒนาบนอุปกรณ์นั้นช่วยสนับสนุน อย่างในเอกสารฉบับนี้เป็นการพัฒนาระบบพจนานุกรมออกเสียงได้บนโทรศัพท์เคลื่อนที่ ต้องอาศัยเทคโนโลยีในหลักๆสองส่วนด้วยกัน คือ ส่วนแรกใช้วิธีการเขียนภาษาจาวาบนโทรศัพท์เคลื่อนที่ หรือ ที่เรียกว่า J2ME ซึ่งสามารถรองรับการเขียนโปรแกรมบนอุปกรณ์ขนาดเล็ก อีกส่วนหนึ่งต้องอาศัยเทคโนโลยีการแปลงข้อความเป็นเสียงพูด หรือที่เรียนว่า Text to speech ซึ่งมีการพัฒนาภาษาจาวาให้สามารถสนับการเขียนโปรแกรมเกี่ยวกับเสียงพูดได้ เมื่ออาศัยเทคโนโลยีในสองส่วนคือ J2ME และ Text to Speech ทำให้โปรเจคนี้สำเร็จขึ้นมาได้

การพัฒนาระบบนี้ขึ้นมาไม่ได้หวังแค่เพียงประโยชน์จากพจนานุกรมออกเสียงได้เท่านั้น แต่หวังเป็นอย่างยิ่งว่าการพัฒนาโปรเจ็คนี้จะเป็นแนวทางในการพัฒนาระบบอื่นๆที่เป็นประโยชน์มากยิ่งๆขึ้นไปทั้งทางตรงและทางอ้อม

5.2 ปัญหา และ ข้อเสนอแนะ

1. ในการพัฒนาระบบบนโทรศัพท์เคลื่อนที่ซึ่งเป็นเทคโนโลยีที่เติบโตค่อนข้างรวดเร็ว มีการแข่งขันกันทั้งทางด้านฮาร์ดแวร์ และซอฟต์แวร์สูง การพัฒนาระบบใหม่ๆขึ้นมาใช้งานบนโทรศัพท์เคลื่อนจึงส่วนเสริมให้สามารถใช้ประโยชน์จากโทรศัพท์เคลื่อนที่ได้มากขึ้น
2. การพัฒนาระบบบนโทรศัพท์เคลื่อนที่ยังมีข้อจำกัดในเรื่องทรัพยากรของอุปกรณ์ ทำให้ไม่สามารถพัฒนานาระบบที่ต้องใช้เนื้อที่ในการเก็บข้อมูลจำนวนมากได้ แต่สามารถเลี่ยงได้โดยการใช้งานข้อมูลต่างๆผ่านทางระบบเครือข่ายแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การพัฒนาระบบซึ่งเกี่ยวข้องกับเทคโนโลยีโดยตรง จำเป็นต้องอาศัยการค้นคว้า หาข้อมูลจากอินเตอร์เน็ตซึ่งมีข้อมูล หรือ เทคโนโลยีใหม่ๆเปลี่ยนแปลงอยู่เสมอ ช่วยให้เราสามารถพัฒนาระบบได้ง่าย และรวดเร็วขึ้น
4. ส่วนหนึ่งที่เป็นปัญหาในการพัฒนาระบบภายในประเทศไทยคือ แหล่งความรู้ที่เป็นภาษาไทยที่เผยแพร่ให้บุคคลทั่วไปได้เข้ามาศึกษาค้นคว้ามีอยู่น้อยมาก ทำให้เป็นข้อจำกัดให้เฉพาะบุคคลที่เข้าใจภาษาอังกฤษสามารถเรียนรู้ได้ และไม่ได้กระจายถึงผู้พัฒนาระบบทั่วไปในวงกว้าง ทำให้การพัฒนาระบบต่างๆภายในประเทศไทยไม่สามารถเติบโตเท่าที่ควร

5.3 งานที่จะทำต่อไป

นำความรู้ที่ได้จากการศึกษาและพัฒนาระบบพจนานุกรมออกเสียงได้บนโทรศัพท์เคลื่อนที่ ไปประยุกต์ใช้ในการพัฒนาระบบอื่นๆที่มีประโยชน์มากขึ้น อาธิ ระบบช่วยเหลือการใช้งานโทรศัพท์ของคนตาบอด ซึ่งในโทรศัพท์เคลื่อนที่บางรุ่นสามารถทำได้แล้ว แต่อาจจะทำให้สามารถมีเสียงพูดเป็นภาษาไทยได้ ซึ่งจะเป็นประโยชน์ต่อผู้ใช้งานคนไทยโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กาญจนา ตันวิสุทธิ์. 2547. **เขียนเกมและโปรแกรมบนมือถือ J2ME.** กรุงเทพฯ: อินโฟ ดิศทริบิวเตอร์ เซ็นเตอร์.

Dutoid, Thierry. 1997. **An Introduction to Text-To Speech Synthesis.** Kluwer Academic.

Nokia. 2005. **Forum Nokia – Developer Resource.** [Online]. Available: http://www.forum.nokia.com/main.html

Ronald, Ashri. 2001. **Professional Java Mobile Programming.** Wrox Press.

Sun Microsystems. 2005a. **Java 2 Platform, Micro Edition.** [Online]. Available: http://java.sun.com/j2me/index.jsp

Sun Microsystems. 2005b. **JSAPI Documentation.** [Online]. Available: http://www.java-channel.org/display.jsp?id=c_7772

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสยาม จันชูโต
สถานที่เกิด	จ.สงขลา
ปริญญาตรี	วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมศาสตร์คอมพิวเตอร์
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
ปีที่สำเร็จการศึกษา	ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้