สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบสั่งงานด้วยเสียง 2

Voice-to-Command Agent 2

โดย

นาย เปรมมินทร์ เดือนอุ่ม

อาจารย์ที่ปรึกษา

อาจารย์สมเกียรติ วังศิริพิทักษ์

เลขหมู่.....
เลขทะเบียน 61369
วัน,เดือน,ปี 17 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิศวกรรมศาสตร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง

เรื่อง ระบบสั่งงานด้วยเสียง 2

Voice-to-Command Agent 2

คณะผู้จัดทำ นายเปรมมินทร์ เดือนอุ่ม รหัส 45015375

.....อาจารย์ที่ปรึกษา

(อ.สมเกียรติ วังศิริพิทักษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสั่งงานด้วยเสียง 2

นาย เปรมมินทร์ เดือนอุ่ม 45015375
อ. สมเกียรติ วังศิริพิทักษ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

การนำเอาเสียงพูดของมนุษย์เข้ามาใช้ในการติดต่อสื่อสารกับเครื่องคอมพิวเตอร์นั้น ได้รับความสนใจศึกษามานานแล้ว ด้วยวัตถุประสงค์เพื่อช่วยอำนวยความสะดวกในการใช้งานคอมพิวเตอร์ แต่เนื่องจากการทำงานที่หลากหลายและขั้นตอนที่ซับซ้อน ซึ่งขึ้นอยู่กับสภาพแวดล้อมในแต่ละระบบปฏิบัติการ ในโครงงานนี้จึงมีการออกแบบการรับคำสั่งที่เป็นเสียงพูดให้สามารถครอบคลุมการใช้งานทั่วๆ ไปของระบบปฏิบัติการวินโดวส์ โดยมีการรับคำสั่งที่เป็นลำดับชั้น ซึ่งมีขั้นตอนและการจัดกลุ่มคำสั่งที่เหมาะสม ส่วนของกลไกที่ใช้ในการรู้จำเสียงในโครงงานนี้ ใช้ SAPI ของไมโครซอฟท์ ซึ่งมีหลักการทำงานแบบฮิดเดนมาร์คอฟโมเดล มีการโมเดลในระดับหน่วยเสียง และมีการรู้จำเสียงที่ต่อเนื่อง โดยการทดลองพบว่ามีการรู้จำคำสั่งที่ค่อนข้างแม่นยำ และมีเปอร์เซ็นต์ความถูกต้องสูงในระดับที่น่าพอใจ ในส่วนของการสั่งงานไปยังระบบปฏิบัติการวินโดวส์นั้น ใช้วิธีการส่งเมสเสจไปยังเมสเสจคิวของระบบเพื่อสลับหน้าต่างไปยังโปรแกรมที่เราต้องการใช้งาน ปรับขนาดของหน้าต่าง และเรียกเมนูของหน้าต่างโปรแกรมที่เรากำลังใช้งานอยู่ หรือเมนูสตาร์ทโปรแกรมขึ้นมา แล้วส่งเมสเสจเพื่อจำลองการกดลูกศร ขึ้น, ลง, ซ้าย, ขวา ที่คีย์บอร์ดเลือกคำสั่งจากเมนู และในหน้าต่างย่อยที่รอการกดปุ่มเพื่อยืนยัน เช่น "Yes", "No", "OK" และ "Cancel" สามารถออกคำสั่งเพื่อกดปุ่มดังกล่าวได้ โดยต้องมีการกำหนดข้อความบนปุ่มนั้นๆ ไว้ในโปรแกรมก่อน ซึ่งในโครงงานนี้ได้กำหนดไว้ตั้งแต่ขั้นตอนการพัฒนาโปรแกรม ผู้ใช้จึงไม่สามารถเพิ่มเติมได้ และนอกจากนี้ในโปรแกรมยังติดปัญหาการเข้าถึงคอนโทรลบางตัว เช่น รีบาร์เมนู(แถวเมนูที่สามารถย้ายตำแหน่งไปยังจุดต่างๆ บนหน้าต่างโปรแกรมได้) และโฟลเดอร์วิว(ส่วนหน้าต่างสำหรับเลือกรายการที่เป็นไฟล์และโฟลเดอร์ในเชลล์ของวินโดวส์) จึงยังไม่ครอบคลุมการทำงานทั้งหมด ตามที่ได้คาดการณ์ไว้ แต่สามารถทำงานได้กับบางโปรแกรม ที่เมนูไม่เป็นรีบาร์คอนโทรล และไม่มีโฟลเดอร์วิวเป็นส่วนประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Voice-to-Command Agent 2

Mr. Premmin Dounaum 45015375
Mr. Somkiat Wangsiripitak Advisor
Academic Year 2003

ABSTRACT

Bringing the voice of human to use communication with the computer is interested for long time .Object is comfortable to use with the computer. But through complex of working depend on environment in many performant system. This project designed for on order by voice to can cover several performant window system. It's on order step by step and set order group to suitable. SAPI of Microsoft is Speech Recognition engine in this project by use Hidden Markov Modeling .It is model in phone level unit and continuous to recognition the voice. This research is discover the remember of voice that acculate and high percentage to correct in satisfactority level when order.In the part of order to Windows system is send message to system message qeue so switch the window to other program that want to use.Resize of the window and call menu program of using window or start menu and send message to representation arrow-key at the keyboard. Choose the order from menu and the part of window to comferm such as Yes, No, OK and Cancel. It can be order to press bottom but before that must assign the label on the bottom in program.This project is assign step begin step of development.User can not add other the order.The problem of this program is access of any control such as Rebar Menu(the position to other position on window of this program) and folder view(the part of window to choice files and floders of list in shell window). It can not cover every features of Windows system .Follow the expect, it can be work whit some program that menu is not Rebar Control and have not Folder View Control in the combine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยคำแนะนำ ความช่วยเหลือ และ การให้โอกาสที่ดี จากอาจารย์หลายท่าน โดยเฉพาะอาจารย์สมเกียรติ วังศิริพิทักษ์ ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้า ขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชา และอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และจริยธรรมอันดี ตลอดระยะเวลาที่ได้ศึกษาอยู่ ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแห่งนี้

ท้ายนี้ขอขอบคุณบุคคลสำคัญในครอบครัวทุกท่านที่เข้าใจ คอยดูแล และให้กำลังใจตลอดมา ถึงแม้ว่าเกิดปัญหา ก็ให้โอกาสในการแก้ปัญหาอย่างเต็มที่ ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

เปรมมินทร์ เดือนอุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VII
บทที่ 1 หลักการและเหตุผล	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	2
1.3 ผลที่คาดว่าจะได้รับ	2
1.4 ขอบเขตโครงงาน	2
1.5 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎี	3
2.1 ศาสตร์ด้านการรู้จำ (Cognitive sciences)	3
2.1.1 แบบจำลองฮิดเด็นมาร์คอฟ (Hidden Markov Modeling)	3
2.1.2 เครื่องมือในการทำงานกับเสียงพูด ของไมโครซอฟท์ (Microsoft Speech SDK)	7
2.2 การทำงานร่วมกันของโปรแกรมบน Microsoft Windows	11
2.2.1 การสร้างและการประมวลผลเมสเสจ	11
บทที่ 3 โครงสร้างและการออกแบบ	13
3.1 ส่วนออกคำสั่งควบคุมไปยังไมโครซอฟท์วินโดว์	14
3.1.1 การเปลี่ยนหน้าจอการทำงาน	14
3.1.2 การเรียกเมนูของแต่ละหน้าต่าง	15
3.1.3 การเรียก Start Menu เพื่อสั่งรันโปรแกรม	16
3.1.4 การออกคำสั่งแทนการกดปุ่ม Button	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้าที่
3.1.5 การนำทุกส่วนมารวมกัน	17
3.2 ส่วนวิเคราะห์ และ การรู้จำคำพูด	18
บทที่ 4 การทดสอบการทำงานของโปรแกรม	21
4.1 การทดสอบการรู้จำ	21
4.2 การใช้งานกับโปรแกรมต่างๆ	30
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	31
5.1 สรุปผลการทดลอง	31
5.2 บทวิจารณ์และแนวทางพัฒนา	31
ภาคผนวก ก การติดตั้งโปรแกรม และชุดเครื่องมือ Speech SDK	33
ภาคผนวก ข XML Grammar ที่ใช้ในโครงงานนี้	41
บรรณานุกรม	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้าที่
รูปที่ 2.1 กระบวนการของการรู้จำเสียงแบบอัตโนมัติ	3
รูปที่ 2.2 การโมเดลประโยคตัวอย่างในชั้นต่างๆ	4
รูปที่ 2.3 ตัวอย่างโมเดลฮิดเด็นมาร์คอฟ 2 สถานะ 2 เครื่องหมายผลลัพธ์	5
รูปที่ 2.4 ตัวอย่างโมเดลฮิดเด็นมาร์คอฟ 5 สถานะ	6
รูปที่ 2.5 ภาพรวมของการติดต่อกันระหว่าง Applications กับส่วนของ Engines	7
รูปที่ 3.1 ภาพรวมของโปรแกรม	13
รูปที่ 3.2 หน้าต่างของโปรแกรมบนวินโดวส์	14
รูปที่ 3.3 หน้าต่างช่วยเหลือสำหรับการเลือกหน้าต่างขึ้นมาทำงาน	14
รูปที่ 3.4 การอ้างอิงถึงเมนูบาร์	15
รูปที่ 3.5 หน้าต่างช่วยเหลือสำหรับการเลือกเมนู	15
รูปที่ 3.6 Start Menu ของวินโดวส์	16
รูปที่ 3.7 ปุ่มบน Dialog box	17
รูปที่ 3.8 ภาพรวมของโปรแกรมเมื่อนำส่วนควบคุมมารวมกัน	19
รูปที่ 4.1 กราฟแสดงผลการรู้จำจากการเทรนที่ต่างกัน	24
รูปที่ 4.2 กราฟแสดงผลการรู้จำจากการเทรนที่ต่างกันในสภาพแวดล้อมต่างกัน	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้าที่
ตารางที่ 2.1 ฟังก์ชันที่ใช้กับเมสเสจในระบบปฏิบัติการวินโดวส์	12
ตารางที่ 4.1 ผลการทดสอบประสิทธิภาพการรู้จำของ SR engine ของ Microsoft	23
ตารางที่ 4.2 ผลการทดสอบการรู้จำในห้องที่มีเสียงรบกวน	25
ตารางที่ 4.3 ผลการทดสอบความถูกต้องของการรู้จำเมื่อจัดคำสั่งเป็นกลุ่ม	27
ตารางที่ 4.4 ผลการทดสอบโปรแกรมโดยเทรนจากห้องที่ไม่มีเสียงรบกวน	28
ตารางที่ 4.5 ผลการทดสอบการรู้จำโดยเทรนจากห้องที่มีเสียงรบกวน	29
ตารางที่ 4.6 ผลการทดลองใช้งานกับโปรแกรมต่าง ๆ	30

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

การนำเอาเสียงพูดของมนุษย์เข้ามาใช้ติดต่อสื่อสารกับคอมพิวเตอร์นั้นได้รับความสนใจศึกษาด้วยวัตถุประสงค์เพื่อความสะดวกสบายในการใช้งานโดยเฉพาะ จากภาพยนต์แนววิทยาศาสตร์บางเรื่องจะเห็นว่าคอมพิวเตอร์ที่ติดไว้กับยานพาหนะจะสามารถสื่อสารกับผู้ใช้ด้วยเสียงพูดและคอยให้คำแนะนำต่างๆ

โดยโครงงานเดิม(ระบบสั่งงานด้วยเสียง จัดทำโดย รวินทร์ วิรัชพินทุ, สุวารี เหลือเพิ่มพร ปริญญานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2546) ได้ทดลองบันทึกคำสั่งผู้ใช้ 10 คำสั่ง โดยเสียงคำสั่งของผู้ใช้ถูกเก็บในรูปแบบไฟล์นามสกุล WAV ก่อนนำเสียงไปวิเคราะห์โดยการแทนค่าเอ็มเอฟซีซี (MFCC parameter) และสร้างรูปแบบอ้างอิงด้วยการทำเวคเตอร์ควอนไตเซชัน (Vector Quantization) ในส่วนควบคุมการทำงานของคอมพิวเตอร์นั้น ใช้หลักการบันทึกชุดของ Windows Messages ที่เกิดขึ้นจากการทำงานแต่ละอย่างของผู้ใช้ด้วย Windows Hook Filter โดยจับคู่ชุดของ Windows Messages กับคำสั่งหนึ่งคำสั่งเพื่อสั่งให้คอมพิวเตอร์ทำงานตามที่ได้บันทึกเอาไว้ เช่น การเลื่อนตำแหน่งของเมาส์ไปยังตำแหน่งของไอคอน (Icon) ของโปรแกรมเพื่อเรียกขึ้นมา ใช้งาน ในกรณีนี้จึงไม่สามารถย้ายตำแหน่งของปุ่มหรือ Shortcut ดังกล่าวไปไว้ตำแหน่งอื่นได้

ในปริญญานิพนธ์นี้จึงได้ศึกษาเพื่อหาวิธีควบคุมหรือสั่งงานคอมพิวเตอร์ในรูปแบบที่ยืดหยุ่นขึ้น และเพื่อรองรับคำสั่งที่หลากหลายและหลีกเลี่ยงความสับสนในคำสั่ง จึงจำเป็นต้องแบ่งคำสั่งออกเป็นลำดับชั้น และให้แต่ละคำสั่งมีความหมายเฉพาะตัว หากใช้วิธีการรู้จำคำสั่งจากผู้ใช้แบบเดิมซึ่งเป็นการรู้จำคำพูดแบบวิเคราะห์คำเสียงจะต้องใช้เวลาในการรู้จำ (Recognition) ที่นานพอสมควร เพราะทุกคำสั่งต้องถูกแทนด้วยหนึ่งคำเสียงทำให้มีคำศัพท์อยู่ในระบบเพื่อการเปรียบเทียบเป็นจำนวนมาก การใช้การรู้จำคำพูดแบบวิเคราะห์หน่วยเสียงจึงเหมาะสมกว่าเพราะมีการใช้หน่วยเสียงร่วมกันในบางคำสั่งทำให้มีคำศัพท์อยู่ในระบบไม่มากเมื่อเทียบกับแบบวิเคราะห์คำเสียง โครงงานนี้จึงเลือก Microsoft Speech SDK 5.1 ซึ่งพัฒนาขึ้นตามทฤษฎีการรู้จำคำพูดแบบวิเคราะห์หน่วยเสียง โดยผู้พัฒนาได้ออกแบบประโยคคำสั่งในรูปแบบของไฟล์เอกสาร XML(XML grammar) ซึ่งมีลักษณะเฉพาะเพื่อนำมาใช้กับ Speech API ของไมโครซอฟท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาวิธีการที่นำมาใช้ในการรู้จำเสียงพูดของมนุษย์แบบต่างๆ

1.2.2 เพื่อศึกษาการพัฒนาโปรแกรมประยุกต์ ที่สามารถควบคุมการทำงานของ Microsoft Windows ได้

1.2.3 เพื่อสร้างโปรแกรมที่สามารถรับคำสั่งจากผู้ใช้ด้วยคำพูดได้

1.3 ผลที่คาดว่าจะได้รับ

1.3.1 ได้รับความรู้เกี่ยวกับการรู้จำเสียงพูดของมนุษย์

1.3.2 ได้รับความรู้เกี่ยวกับการสั่งงานคอมพิวเตอร์โดยไม่ต้องใช้เมาส์ และ คีย์บอร์ด

1.3.3 โปรแกรมที่พัฒนาสามารถทำงานได้ถูกต้องตามความต้องการของผู้ใช้

1.4 ขอบเขตของโครงงาน

สร้างโปรแกรมประยุกต์ที่สามารถรับคำสั่งของผู้ใช้ด้วยคำพูด โดยแบ่งโครงงานออกเป็นสองส่วนคือส่วนของการรู้จำคำสั่งเสียงพูดของผู้ใช้ และส่วนที่เข้าไปควบคุมการใช้งานทั่วไปภายใต้ระบบปฏิบัติการวินโดว์โดยใช้ Microsoft Speech SDK 5.1 ในการรู้จำเสียงพูด และ ออกแบบรูปแบบการสั่ง เช่น การเรียกใช้คำสั่งในเมนู การปรับขนาดหน้าต่าง และการเปลี่ยนหน้าต่างทำงาน ให้ง่ายต่อการใช้งานมากขึ้น

1.5 วิธีการดำเนินงาน

1.5.1 ศึกษาทฤษฎีที่เกี่ยวกับการรู้จำเสียงพูดเพื่อการเลือกใช้ โดยพิจารณาความเหมาะสม

1.5.2 ศึกษาการทำงาน และวิธีใช้ Microsoft Speech SDK 5.1

1.5.3 พัฒนาโปรแกรมประยุกต์ เพื่อควบคุมการทำงานของระบบปฏิบัติการวินโดว์

1.5.4 รวมโปรแกรมทั้งสองส่วนเข้าด้วยกันเพื่อให้สามารถทำงานร่วมกันได้

1.5.5 ทำการทดสอบและบันทึกผลการใช้งานโปรแกรม

1.5.6 แก้ไขส่วนที่ผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

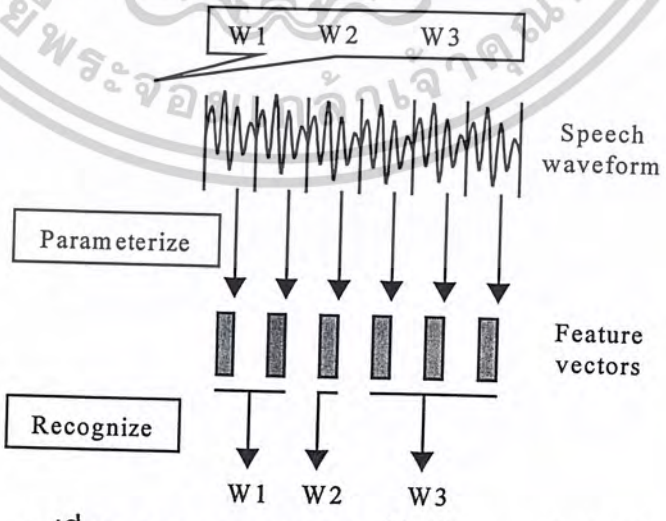
บทที่ 2
ทฤษฎี

การรู้จำคำพูด(speech recognition) และ การรู้จำเสียงพูด(voice recognition) เป็นการแปลงสัญญาณเสียงตามธรรมชาติ(acoustic signal) ในที่นี้คือเสียงที่เกิดจากการพูดของมนุษย์ผ่านอุปกรณ์รับเสียง เช่น ไมโครโฟน เข้าไปในระบบ โดยผลลัพธ์สุดท้ายของการรู้จำจะถูกใช้เพื่อออกคำสั่งควบคุม หรือเพื่อกรอกข้อมูลบนเอกสาร โดยผลลัพธ์ของการรู้จำเสียงจะนำไปใช้เป็นเหมือนอินพุต (input) ของการทำงานส่วนอื่นๆ ที่เกี่ยวข้อง

2.1 ศาสตร์ด้านการรู้จำ (Cognitive sciences)

หมายถึง การให้คอมพิวเตอร์รู้จักรูปแบบของสิ่งใดสิ่งหนึ่ง โดยเก็บบันทึกคุณลักษณะบางอย่างเพื่อให้สามารถบอกได้ว่าสิ่งนั้นๆ คืออะไร ซึ่งแท้จริงแล้วมนุษย์เองไม่ได้ทราบถึงคุณลักษณะพิเศษเหล่านี้ในการแยกแยะตามธรรมชาติ ศาสตร์ด้านการรู้จำ(Cognitive sciences) ในช่วงแรกๆ นั้นจะนำหลักสถิติมาใช้ แต่พบว่า มีขีดจำกัดค่อนข้างมากในการเลียนแบบการรู้จำของมนุษย์ เพราะสมองปัญญา และความฉลาดของมนุษย์ ไม่ใช่เรื่องของสถิติเพียงอย่าง

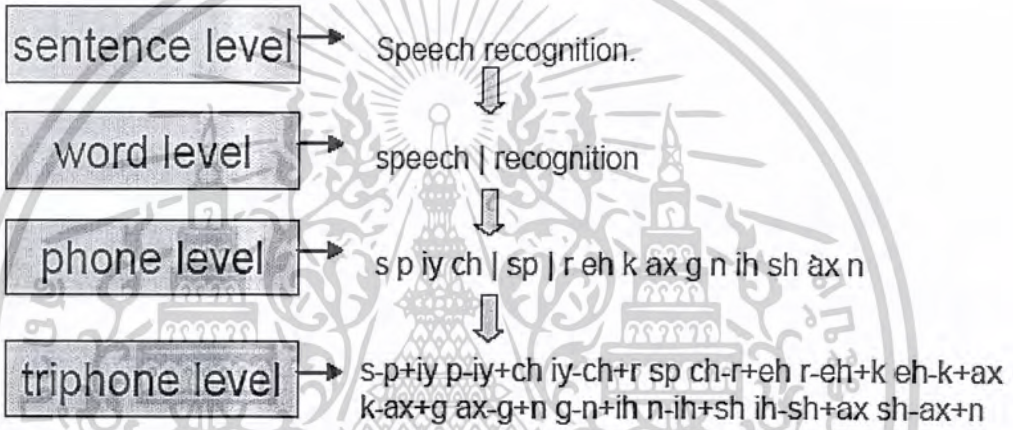
2.1.1 แบบจำลองฮิดเด็นมาร์คอฟ(Hidden Markov Modeling)



รูปที่ 2.1 กระบวนการของการรู้จำเสียงแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปคือวิธีการในยุคแรกๆ ที่ใช้แปลงคำพูดที่ได้ยินหรือสัญญาณเสียงให้เป็นเครื่องหมายที่เหมาะสม เครื่องหมายที่เหมาะสมในที่นี้คือลักษณะพิเศษที่สามารถหาได้จากสัญญาณเสียง เช่น ความถี่มูลหลัก(fundamental frequency) สัญญาณความถี่(spectral envelop) เป็นต้น วิธีการหนึ่งที่นิยมนำมาใช้และให้ผลดีที่สุดขณะนี้ในการแปลงคำพูดให้เป็นเครื่องหมายคือการใช้วิธีการที่เรียกว่าแบบจำลองฮิดเด็นมาร์คอฟ (Hidden Markov Modeling หรือ HMM)



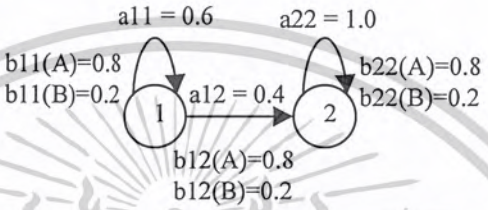
รูปที่ 2.2 การโมเดลประโยคตัวอย่างในชั้นต่าง ๆ

แบบจำลองฮิดเด็นมาร์คอฟ สามารถเลือกโมเดลเป็นแบบ ***คำเสียง***หรือ***หน่วยเสียง*** ก็ได้ขึ้นอยู่กับการใช้งาน โดยแบบคำเสียงเหมาะกับระบบที่มีการรู้จำคำศัพท์ที่น้อย และรู้จำคำพูดที่ไม่ติดต่อกัน ส่วนแบบหน่วยเสียงนั้น เหมาะสำหรับระบบที่มีคำศัพท์มาก(มากกว่า 1000 คำ) และรู้จำเสียงแบบพูดติดต่อกัน ซึ่งต้องมีการโมเดลแต่ละหน่วยเสียงก่อน และรู้จำหน่วยเสียงนั้นๆ ต่อจากนั้นก็มีวิธีการสร้างคำจากหน่วยเสียง ถ้ามีการรู้จำหน่วยเสียงผิดไปหนึ่งหน่วยเสียง จะทำให้คำนั้นรู้จำผิดไปด้วย

ตามทฤษฎีแบบจำลองฮิดเด็นมาร์คอฟ คือ การเก็บรวบรวมของสถานะหลายๆ สถานะที่ถูกเชื่อมโยงโดยการเปลี่ยนสถานะในที่นี้คือ ลักษณะของสัญญาณเสียงหรือเครื่องหมายที่หมายถึงสัญญาณเสียงนั้นที่เวลาใดๆ ส่วนการเปลี่ยนสถานะ คือ การเลื่อนหรือเปลี่ยนแปลงของสัญญาณเสียงจากเวลาหนึ่งไปหาอีกเวลาหนึ่ง แบบจำลองฮิดเด็นมาร์คอฟ มีค่าความน่าจะเป็นอยู่สองชนิดคือ ค่าความน่าจะเป็นที่การเปลี่ยนสถานะหนึ่งที่จะเกิดขึ้น (transition probability) และ ค่าความน่าจะเป็นของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ (output symbol) เมื่อมีการเปลี่ยนสถานะหนึ่งเกิดขึ้น ผลลัพธ์ที่ว่านี้จะนำไปใช้ในการตัดสินใจว่าเสียงที่ได้ยินนั้นเป็นเสียงอะไร (output probability) รูปต่อไปนี้แสดงถึงตัวอย่างของแบบจำลองฮิดเด็นมาร์คอฟที่มี 2 เอาท์พุตคือ A และ B และ 2 สถานะ คือ 1 และ 2



รูปที่ 2.3 ตัวอย่างโมเดลฮิดเด็นมาร์คอฟ ที่มี 2 สถานะ และมี 2 เครื่องหมายผลลัพธ์

จากรูปเป็นตัวอย่างโมเดลแบบง่ายๆ ที่ผ่านการเทรนหรือมีการกำหนดค่าความน่าจะเป็นของ a และ b มาแล้ว ค่า a12=0.4 หมายความว่า เมื่อสถานะปัจจุบันเป็น 1 สถานะ ต่อไปจะเป็นสถานะ 2 มีความน่าจะเป็นอยู่ 0.4 ส่วน b12(A)=0.8 คือ เมื่อมีการเปลี่ยนสถานะจากสถานะ1 ไปหาสถานะ 2 ความน่าจะเป็นที่จะมีผลลัพธ์เป็น A คือ 0.8 (เมื่อมีการเปลี่ยนสถานะจะมีการสร้างผลลัพธ์หนึ่งออกมาเท่านั้น) หรือ อธิบายได้ดังนี้

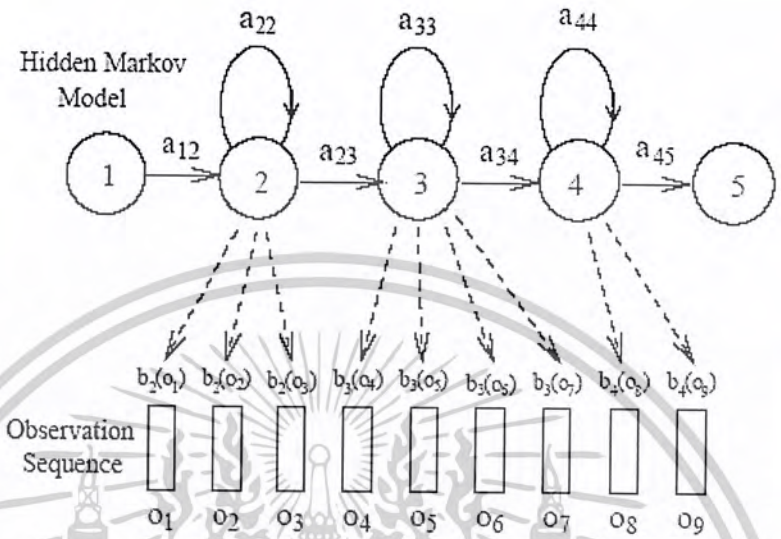
$$a_{ij} = P(X_{t+1} = j \mid X_t = i)$$

คือ ความน่าจะเป็นของสถานะถัดไปจะเป็น j เมื่อสถานะปัจจุบันเป็น i

$$b_{ij}(k) = P(Y_t = k \mid X_t = i, X_{t+1} = j)$$

คือ ความน่าจะเป็นของผลลัพธ์เป็น k เมื่อมีการเปลี่ยนสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างโมเดลฮิดเด็นมาร์คอฟ ที่มี 5 สถานะ

จึงสรุปได้ว่าแบบจำลองฮิดเด็นมาร์คอฟในสถานะแรกมีการตั้งสมมุติฐานอยู่สองประการคือ

1. ความน่าจะเป็นของสถานะถัดไปขึ้นอยู่กับสถานะปัจจุบันเท่านั้น
2. ความน่าจะเป็นของเครื่องหมายผลลัพธ์(symbol output) ถัดไปขึ้นอยู่กับการเปลี่ยนสถานะปัจจุบันเท่านั้นไม่ใช่การเปลี่ยนสถานะในอดีต

ปัญหาของ แบบจำลองฮิดเด็นมาร์คอฟ มีอยู่ 3 ข้อดังนี้ คือ

1. Evaluation problem : แบบจำลองสามารถสร้างหรือเข้าใจเสียงที่ได้ยินดีแค่ไหน
2. Decoding Problem : ลำดับของสถานะในแบบจำลองสร้างเสียงที่ได้ยินดีแค่ไห
3. Learning problem : ตัวแปร (parameter) ของแบบจำลองควรเป็นอย่างไรเพื่อที่จะทำให้มีความผิดพลาดในการสร้างเสียงหรือเข้าใจเสียงที่ได้ยินน้อยที่สุด

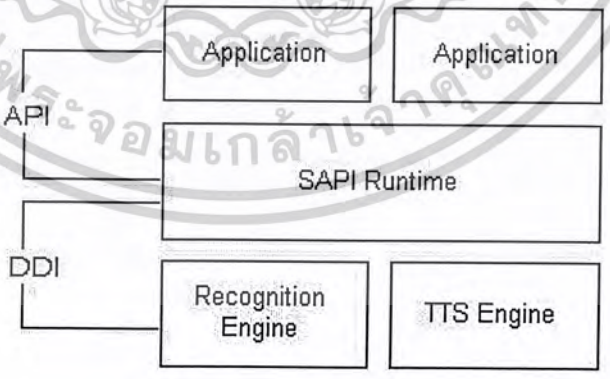
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 **เครื่องมือในการทำงานกับเสียงพูด ของไมโครซอฟท์**(Microsoft Speech SDK)

Microsoft Speech SDK คือเครื่องมือสำหรับช่วยสร้างกลไกการทำงาน(engines) และโปรแกรมประยุกต์ที่เกี่ยวกับการพูด สำหรับทำงานบนระบบปฏิบัติการไมโครซอฟท์วินโดวส์ ซึ่งภายในประกอบด้วย

- Microsoft® Win32®-compatible speech application programming interface (หรือ SAPI)
- Microsoft continuous speech recognition engine
- Microsoft concatenated speech synthesis (หรือ text-to-speech) engine
- ชุดเครื่องมือพัฒนาช่วย compiling source code และ executing commands
- ตัวอย่างโปรแกรม และ การสาธิตการใช้ Speech กับเทคโนโลยีของ engine อื่นๆ
- ตัวอย่าง speech recognition และ speech synthesis engines เพื่อทดสอบกับ applications ที่สามารถสื่อสารกับผู้ใช้ด้วยเสียงพูด
- เอกสารประกอบ(documentation) ของ SDK

ในส่วนของ SAPI นั้นมีการติดต่อระดับสูงระหว่างโปรแกรมประยุกต์และกลไกการทำงานที่เกี่ยวกับการพูด ซึ่งใน SAPI นั้นจะมีการทำงานกับรายละเอียดระดับล่างด้วยตัวเองซึ่งเป็นการดีสำหรับการทำงานแบบ real-time ของระบบ



รูปที่ 2.5 ภาพรวมของการติดต่อกันระหว่าง Applications **กับส่วนของ** Engines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สองส่วนประกอบพื้นฐานของ SAPI engines คือ text-to-speech (TTS) system และ speech recognizers โดยใน TTS systems นั้นเป็นการสังเคราะห์เสียงจากข้อความหรือไฟล์โดยใช้การประกอบกันขึ้นของเสียง ส่วน recognizers นั้นเป็นตัวแปลงเสียงพูดของมนุษย์ไปสู่ข้อความหรือไฟล์ในคอมพิวเตอร์ ซึ่งในส่วนนี้จะมีการทำงานโดยใช้หลักการฮิดเดนมาร์คอฟแบบสร้างโมเดลจากหน่วยเสียง และเป็นส่วนที่ถูกนำมาใช้ในโครงงานนี้

ใน SAPI เวอร์ชัน 5.1 มีการสนับสนุน OLE automation ทำให้ภาษาการเขียนโปรแกรมอื่นๆ นอกจาก C/C++ บางภาษาสามารถนำ SAPI ไปใช้ร่วมในการพัฒนาโปรแกรมประยุกต์ได้ ซึ่ง automation ก็คือความสามารถของออบเจ็คท์ ที่จะติดต่อกับออบเจ็คท์อื่นๆ โดยที่พวกมันเหล่านี้อาจจะเป็น applications หรือ ActiveX® controls ก็ได้ ซึ่งในที่สุดมันก็กลายมาเป็น COM (component object model) ในปัจจุบัน ส่วนประกอบย่อยๆ ที่ซับซ้อนจะถูกตัดออกไปทำให้โปรแกรมประยุกต์นั้นสามารถดึงการทำงานและข้อมูลของโปรแกรมอื่นๆ มาใช้ได้ เช่น หากเราเขียนตัวตรวจคำผิดด้วย Visual Basic เราต้องใช้เวลาและความพยายามอย่างมากในการทำ แต่เราสามารถใช้งานตัวตรวจคำผิดของ Microsoft Word ได้ซึ่งง่ายกว่าโดยมาตรฐานของการใช้ร่วมกันที่เรียกว่า Visual Basic for Applications (VBA)

การใช้แบบที่สองของ automation ที่ทำให้มันถูกเรียกว่า components ซึ่งมีรูปแบบเป็น GUI (Graphics User Interface) ตัวอย่างเช่น text box ใน Visual Basic ที่ข้อความข้างในจะเป็นตัวหนาทั้งหมด หรือตัวอักษรมีขนาดเดียวกันทั้งหมดแต่เราสามารถใช้ rich text edit box ที่มีการทำงานกับข้อความได้ดีกว่า ซึ่งถูกเพิ่มเข้าไปใน Visual Basic Toolbox โดยการเพิ่มจาก Components menu ซึ่ง rich text edit box ก็คือ COM object หรือ automation ตัวนึงที่มีความยืดหยุ่นและสะดวกสบายในการเขียนโปรแกรมนั่นเอง

การใช้แบบที่สามของ automation คือ การเข้าถึงจาก dynamic link libraries (DLLs) โดยในแบบนี้ automation จะเป็น applications ที่ไม่มี user interface เช่นเดียวกับใน SAPI ที่ตัวของ instance จะไม่มี GUI โดยตรง ขณะที่ผู้ใช้สามารถรวม SAPI functions ไว้ใน applications ของตัวเองได้โดยผู้พัฒนาสามารถเข้าถึง SAPI's functions ทั้งหมดโดยการเชื่อมโยงกับไฟล์ DLLs โดยตรง เช่น sapi.dll จะถูกโหลดเข้ากับระบบปฏิบัติการเมื่อผู้พัฒนาติดตั้ง SAPI

จากการใช้งานทั้งหมดที่แตกต่างกันของ automation ที่อธิบายไว้ด้านบนนั้นก็ถือว่าเป็นรากฐานที่สำคัญเพื่อการพัฒนาโปรแกรมแบบ COM โดยตัวมันเองแล้ว automation เป็นเทคโนโลยีที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซับซ้อนซึ่งได้ถูกคิดขึ้นมาหลายปี มีการเปลี่ยนแปลงเทคโนโลยีและเพิ่มความต้องการที่ผิดแปลกออกไปจากเดิม โปรแกรมเมอร์เก่าๆ อาจจะรู้จัก OLE (object linking and embedding) ซึ่งทำให้เกิดเป็น ActiveX, COM และ COM+ ทั้งหมดนั้นคือวิวัฒนาการของสิ่งที่ถูกเรียกว่า automation

ต่อไปนี้คือ COM interfaces ใช้เพื่อจัดการและควบคุมการรู้จำเสียงพูด เพื่อนำเข้าไปรวมกับโปรแกรมประยุกต์ของเรา ส่วนดังกล่าวนี้ถูกออกแบบมาให้มีการทำงานอยู่ในชั้น API หรือที่เรียกว่า application level แต่อาจมีบางส่วนที่เข้าไปยุ่งเกี่ยวกับส่วน Engine-Level Interfaces บ้าง แต่อย่างไรก็ตามในที่นี้จะกล่าวถึงแต่ application level ของ Speech Recognition interface ส่วนที่จะนำมาใช้ในโครงงานนี้เท่านั้น ซึ่งมีดังนี้

- *ISpRecoContext*
- *ISpRecoGrammar*
- *ISpRecognizer*
- *ISpPhrase*

ISpRecoContext

เป็น interface ที่ยอมให้ applications สร้าง ข้อความที่มีความหมายเพื่อใช้กับ SR engine โดยในออบเจ็คท์ ISpRecoContext แต่ละอันสามารถรับ SR events ที่ต่างกัน (ISpEventSource และ SPEVENTENUM) และสามารถใช้ recognition grammars ที่ต่างกัน (ISpRecoGrammar) โดยใน 1 applications ควรจะมี ISpRecoContext อย่างน้อย 1 instance เพื่อรับการ recognitions เข้าไปใน applications หากมีหลายๆ instances จะทำให้สามารถแยกประเภทของการ recognition ที่ต่างกันได้ เช่น multiple-document-interface (MDI) application สามารถจะทำงานร่วมกับออบเจ็คท์ ISpRecoContext คนละตัวกันในแต่ละหน้าต่าง document เพื่อการใช้ grammar และ event - processing เฉพาะในขอบเขตที่จำกัดของตนเอง การสร้างออบเจ็คท์ ISpRecoContext ใหม่นั้นสามารถทำได้โดยเรียกใช้ method ที่ชื่อ CreateRecoContext ในออบเจ็คท์ของ ISpRecognizer (ISpRecognizer::CreateRecoContext) และ ประกาศให้ใช้ recognizer ร่วมกัน(shared) โดยการเรียก method ที่ชื่อ CoCreateInstance ในออบเจ็คท์ของ IspRecognizer ด้วยค่าพารามิเตอร์เป็น CLSID_SpSharedRecoContext

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISpRecoGrammar

เป็น interface ที่ยอมให้ applications สามารถจัดการกับคำ(word) หรือ วลี(phrases) ที่ SR engine จะทำการ recognize ซึ่งในออบเจ็คท์ SpRecognizer ตัวเดียวสามารถมีได้หลายออบเจ็คท์ ของ SpRecoContext ที่มาทำงานร่วมกัน และทำนองเดียวกันที่ออบเจ็คท์ SpRecoContext ตัวเดียว สามารถมีได้หลายออบเจ็คท์ SpRecoGrammar ทำให้ applications สามารถแยกชนิดของ วลี และ ข้อความภายในเพื่อสร้าง application logic ที่เข้าใจง่าย แต่ละออบเจ็คท์ SpRecoGrammar ยัง สามารถมี context-free grammar (CFG) และการโหลด dictation grammar ในเวลาเดียวกันได้ การสร้าง context-free grammars นั้นดูได้จาก ภาคผนวก ข.

ISpRecognizer

เป็น interface ที่ยอมให้ applications สามารถควบคุมการทำงานของ speech recognition (SR) engine แต่ละ ISpRecognizer interface นั้นทำหน้าที่แทน SR engine หนึ่งตัว โดย application สามารถที่จะติดต่อ recognizer object แต่ละตัวด้วย recognition contexts อย่างน้อย 1 object จาก ส่วนนี้ทำให้ application สามารถควบคุม recognition grammars ที่ใช้อยู่ให้เริ่มต้นและหยุดการ

recognition หรือรับ events และผลของการ recognition โดยตัว ISpRecognizer interface นั้นมีส่วน เสริมการควบคุมบางอย่างของ SR engine และ audio input โดยทั่วไป application มาตรฐานอาจจะ ไม่ต้องการเรียกใช้ methods อะไรมากมายจากส่วนนี้เลยนอกจากการทำงานพื้นฐานที่จำเป็นเท่านั้น

การใช้งานสองอย่างของ ISpRecognizer และ ISpRecoContext ใน SAPI คือ สำหรับการ recognition แบบ "in-process" (InProc) คือ SR engine ถูกสร้างใน process เดียวกันกับส่วนของ application มีเพียง application นี้เท่านั้นที่สามารถติดต่อกับ recognizer นี้ได้ อีกแบบคือ "shared-recognizer" คือ SR engine ถูกสร้างแยกกันคนละ process ซึ่งพวกมันเหล่านั้นสามารถที่จะมีแค่ shared engine เพียงตัวเดียวอยู่ในระบบ และทุก applications ใช้ shared engine นี้เพื่อติดต่อกับ recognizer เดียวกัน จึงทำให้ speech applications หลายๆ ตัวสามารถทำงานพร้อมกันได้ในเวลา เดียวกัน และในแบบนี้ยังยอมให้ผู้ใช้พูดไปยังแต่ละ application พร้อมกัน ดังนั้นการ recognition จะ ถูกกระทำจาก grammars ของ applications ทุกตัว ซึ่งใน desktop-based speech applications ได้ แนะนำให้ใช้ shared recognizer เพราะว่าวิธีนี้ SAPI applications หลายๆ ตัว ทำงานแค่ครั้งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ISpPhrase

เป็น interface หลักที่ถูกใช้เพื่อเข้าถึงข้อมูลที่บรรจุอยู่ในวลี การใช้ interface นี้ applications สามารถเรียกดู recognition informations เช่น ข้อความ recognized (หรือ hyphthesized) กฎ(rule) ของการ recognized และ semantic tag หรือ property information ใน application ยังสามารถส่งต่อ ข้อมูลวลีไปที่แถวของข้อมูลที่เคลื่อนที่เข้าสู่การ recognitions ใน disk, network หรือ memory

2.2 การทำงานร่วมกันของโปรแกรมบน Microsoft Windows

ในวินโดวส์นั้นเมสเสจถือเป็นตัวแทนต่อทุกๆ เหตุการณ์ที่เกิดขึ้นทั้งหมด ซึ่งต้องการการตอบสนอง โดยแมสเสจนั้นเป็นตัวแปรโครงสร้างชนิดหนึ่งที่ประกอบด้วยส่วนค่าอ้างอิงของเมสเสจนั้น กับส่วนพารามิเตอร์ของเมสเสจ โดยที่พารามิเตอร์จะขึ้นกับเมสเสจนั้นๆ โดยจะมีความเกี่ยวเนื่องกัน

2.2.1 การสร้างและการประมวลผลเมสเสจ

วินโดวส์จะมีการสร้างเมสเสจทุกครั้งที่มีเหตุการณ์ใดๆ เกิดขึ้น โดยวินโดวส์จะจะนำเมสเสจที่สร้างขึ้นเหล่านี้ใส่เข้าไปในคิวของระบบ และจะส่งผ่านไปยังคิวของโปรแกรมที่เหมาะสม โดยคิวเป็นแบบเข้าก่อนออกก่อน โปรแกรมหรือแอพพลิเคชันจะเป็นตัวดึงเมสเสจออกไปเองโดยใช้ฟังก์ชัน GetMessage โดยจะเริ่มดูว่าในคิวของตัวเองมีเมสเสจหรือไม่ ถ้ามีก็จะดึงเมสเสจแรกสุดไป แต่ถ้าไม่มีก็จะคืนการควบคุมให้กับวินโดวส์ เมื่อดึงเมสเสจออกจากคิวแล้วต้องใช้ฟังก์ชัน DispatchMessage เพื่อแจกแจงเมสเสจแล้วส่งเมสเสจนั้นๆ ไปให้ฟังก์ชันประจำวินโดว์ต่างๆ ของมัน

นอกจากนั้นแล้วยังมีบางเมสเสจที่วินโดวส์ต้องการส่งไปยังฟังก์ชันประจำวินโดว์โดยตรง เราเรียกเมสเสจเหล่านี้ว่า เมสเสจลัดคิว โดยวินโดวส์จะส่งเมสเสจเหล่านี้ด้วยฟังก์ชัน SendMessage และฟังก์ชันประจำวินโดว์นั้นๆ ยังสามารถส่งค่าการประมวลผลเมสเสจกลับมาได้อีกด้วย แต่ก็ไม่ได้หมายความว่ามีเพียงวินโดวส์เท่านั้นที่สร้างเมสเสจขึ้นมาได้ แอพพลิเคชันเองก็สามารถสร้างเมสเสจขึ้นมาเพื่อส่งให้กับคิวของตัวเองหรือคิวของแอพพลิเคชันอื่นๆ ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการตีความของเมสเสจโดยทั่วไปแล้วใช้ฟังก์ชัน TranslateMessage โดยตัวอย่างต่อไปนี้จะเป็นโค้ดของส่วนเมสเสจลูปที่อยู่ในฟังก์ชันหลักของแอพพลิเคชัน

```
while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

ฟังก์ชันที่ใช้กับเมสเสจในวินโดวส์มีดังตารางต่อไปนี้

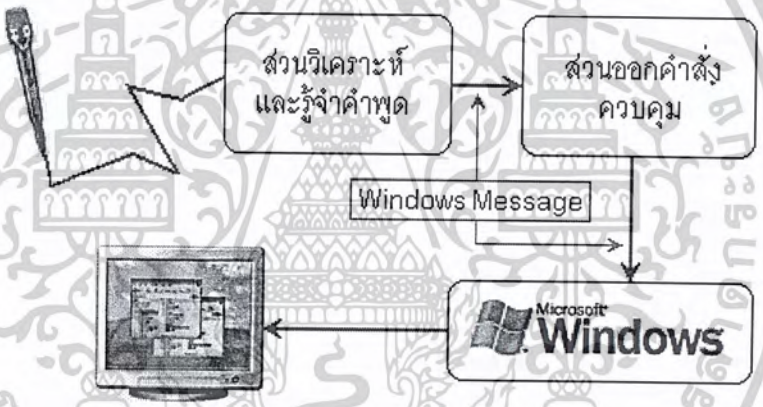
ฟังก์ชัน	ลักษณะการทำงาน
CallWindowProc	ส่งผ่านข้อมูลของเมสเสจไปยังฟังก์ชัน
DispatchMessage	แจกแจงเมสเสจและส่งไปยังฟังก์ชันประจำวินโดว์
GetMessage	รับเมสเสจในช่วงที่กำหนด
GetMessagePost	ให้ค่าตำแหน่งของเมาส์ล่าสุดขณะที่รับเมสเสจ
GetMessageTime	ให้ค่าเวลาล่าสุดขณะมีการรับเมสเสจ
PeekMessage	ตรวจสอบว่ามีเมสเสจในคิวหรือไม่โดยไม่ต้องดึงออกมา
PostAppMessage	ส่งเมสเสจไปยังแอพพลิเคชัน
PostMessage	ส่งเมสเสจไปยังคิวของแอพพลิเคชัน
PostQuitMessage	ส่ง WM_QUIT ไปให้แอพพลิเคชัน
ReplyMessage	ตอบรับเมสเสจ
SendMessage	ส่งเมสเสจไปยังวินโดว์ต่างๆ
TranslateAccelerator	ประมวลผลคีย์ลัดของเมนูแล้วส่งเป็นเมสเสจให้วินโดว์
WaitMessage	ส่งการควบคุมไปให้แอพพลิเคชันอื่น

ตารางที่ 2.1 ฟังก์ชันที่ใช้กับเมสเสจในระบบปฏิบัติการวินโดวส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการออกแบบ

ในโครงงานนี้แบ่งการออกแบบโปรแกรมออกเป็นสองส่วนซึ่งเป็นอิสระต่อกัน คือ ส่วนของการวิเคราะห์และรู้จำเสียง และ ส่วนออกคำสั่งควบคุมไปยังระบบไมโครซอฟท์วินโดว์ ซึ่งมีภาพรวมดังนี้



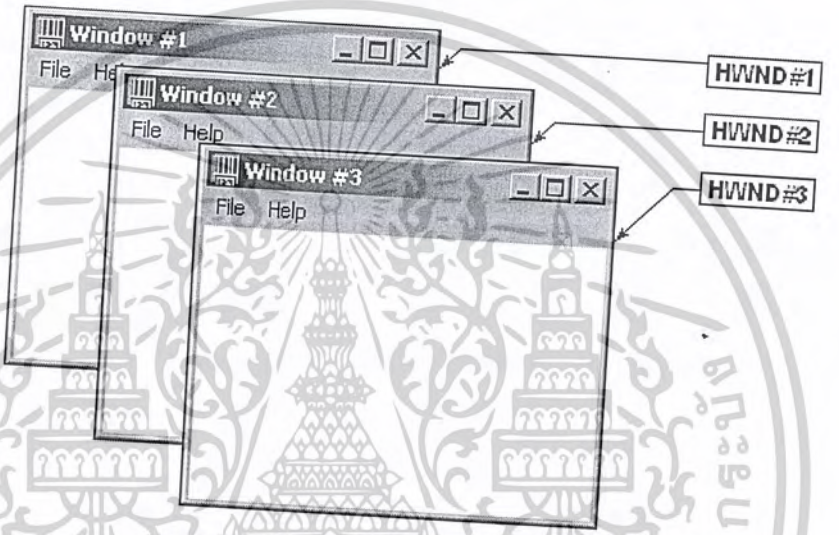
รูปที่ 3.1 ภาพรวมของโปรแกรม

เป็นที่รู้กันดีว่าระบบปฏิบัติการวินโดว์นั้นทำงานร่วมกันกับโปรแกรมต่างๆ ด้วยการรับส่ง Message หรือที่เรียกว่า Windows Message ซึ่งทำให้ทุกโปรแกรมสามารถทำงานและใช้ทรัพยากรของเครื่องร่วมกันได้อย่างสอดคล้อง ดังนั้นในโครงงานนี้จึงใช้การส่ง Windows Message เป็นหลักในการควบคุมการทำงานของโปรแกรมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

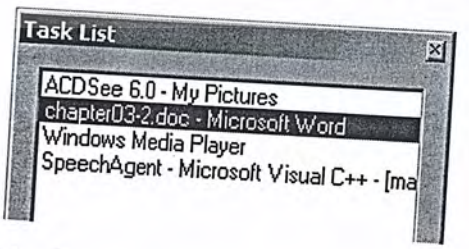
3.1 ส่วนออกคำสั่งควบคุมไปยังไมโครซอฟท์วินโดว์

3.1.1 การเปลี่ยนหน้าจอการทำงาน



รูปที่ 3.2 หน้าต่างของโปรแกรมบนวินโดว์

เนื่องจากระบบปฏิบัติการวินโดว์มีความสามารถในการทำงานแบบ Multi – processing คือ ทำงานหลายหน้าต่างพร้อมกันโดยใช้ค่า window handle (HWND) เป็นการอ้างถึงหน้าต่างดังกล่าว ในขณะที่เวลาเดียวกันนั้นมีโปรแกรมที่ถูกรันอยู่หลายโปรแกรมด้วยกัน โปรแกรมในโครงงานนี้จึงควรมีการสลับหน้าจอไปยังโปรแกรมต่างๆ ที่รันอยู่ในระบบได้ และควรมีคำสั่งที่สามารถสลับหน้าจอการทำงานไปยังโปรแกรมต่างๆ ที่ถูกรันอยู่ในขณะนั้นได้ โดยในโครงงานนี้ใช้หน้าต่างช่วยเหลือเพื่อช่วยออกคำสั่ง

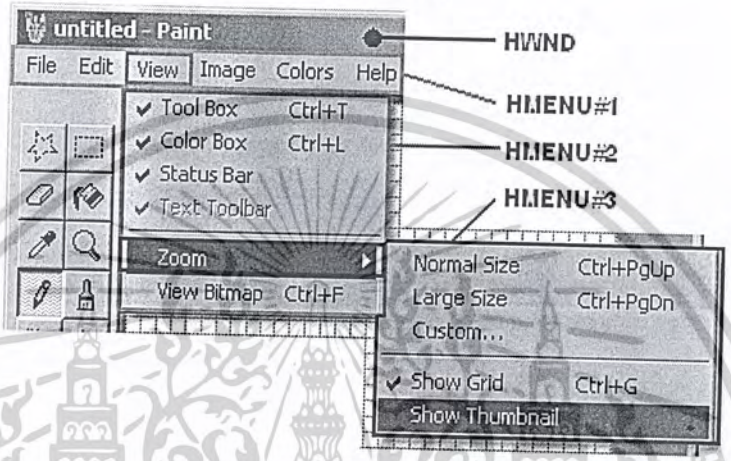


รูปที่ 3.3 หน้าต่างช่วยเหลือสำหรับการเลือกหน้าต่างขึ้นมาทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

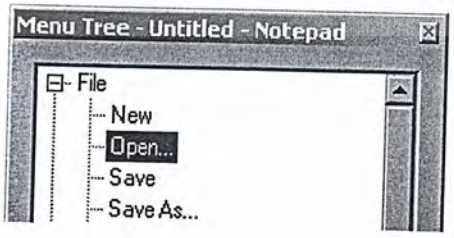
3.1.2 การเรียกเมนูของแต่ละหน้าต่าง

ส่วนประกอบหลักๆ ที่หน้าต่างโปรแกรมส่วนใหญ่มีก็คือ เมนูบาร์ ซึ่งบรรจุคำสั่งที่จำเป็นในการใช้งานโปรแกรมไว้เกือบทั้งหมด



รูปที่ 3.4 การอ้างอิงถึงเมนูบาร์

การดึงค่า menu handle (HMENU) นั้นสามารถทำได้โดยการส่งค่า window handle เข้าไปเพื่อเรียกดูค่า menu handle หลัก ที่ผูกไว้กับตัวหน้าต่างดังกล่าว หากเมนูนั้นมีเมนูย่อยอยู่ภายใน เมื่อเราเลือกเมนูนั้น มันจะ pop-up เมนูย่อยออกมา ในการเขียนโปรแกรมนั้นเราสามารถเรียกดูค่า menu handle ของเมนูย่อยได้โดยส่งค่า menu handle ที่เป็น parent ของมันเข้าไปโดยระบุลำดับที่ (index) ของเมนูย่อยที่เราต้องการเรียกดูให้กับมัน โดยค่าลำดับที่นั้นเริ่มต้นจาก 0 และในเมนูย่อยของเมนูย่อยก็สามารถทำได้เช่นเดียวกัน จากรูปที่ 3.4 จะเห็นได้ว่า HMENU#3 เป็นเมนูย่อยของ HMENU#2 และ HMENU#2 เป็นเมนูย่อยของ HMENU#1 ตามลำดับ

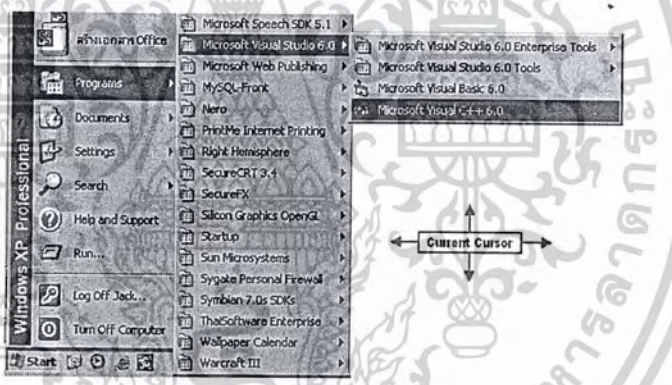


รูปที่ 3.5 หน้าต่างช่วยเหลือสำหรับการเลือกเมนู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การเรียก Start Menu เพื่อสั่งรันโปรแกรม

เราสามารถส่ง window message ไปยัง window ต่างๆ ที่กำลังถูกรันอยู่ในวินโดว์ได้ เพียงแค่เราทราบค่า window handle ของมัน(กรณีที่มันเป็น parent window ที่มีการกระจาย window message ไปยัง child ทุกตัว) โดย start menu ในวินโดว์นั้นก็เป็น window ตัวหนึ่งเหมือนกัน เพียงแต่มีรูปแบบที่ไม่เหมือนหน้าต่างของโปรแกรมทั่วๆ ไป โดยเราสามารถเรียก start menu ขึ้นมาได้ทุกเมื่อ ไม่ว่าเราจะกำลังทำงานอยู่บนหน้าต่างไหน โดยใช้ฟังก์ชัน PostMessage โดยส่ง WM_SYSCOMMAND ไปพร้อมกับพารามิเตอร์ SC_TASKLIST วินโดว์ก็จะ pop-up หน้าต่างของ start menu ขึ้นมาให้เราทันที จากนั้นเราก็ควบคุมการเลื่อนของแถบสีได้โดยไม่ต้องใช้หน้าต่างช่วยเหลือเหมือนส่วนที่ผ่านมา



รูปที่ 3.6 Start Menu ของวินโดว์

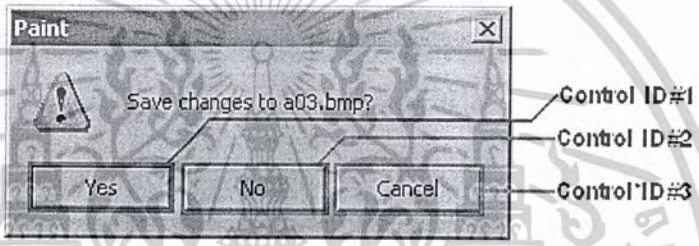
โดยเมื่อ pop-up หน้าต่างของ start menu ขึ้นมาแล้วเราสามารถจำลองการกด ปุ่มลูกศร ขึ้น-ลง ซ้าย-ขวา ของคีย์บอร์ด เพื่อเป็นการเลื่อนแถบสีเพื่อเลือกโปรแกรมที่จะสั่งรัน และ จำลอง Windows Message ของการกดปุ่ม ENTER กับ ESC เพื่อสั่งรันและยกเลิกการเรียก start menu ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.1.4 การออกคำสั่งแทนการกดปุ่ม Button

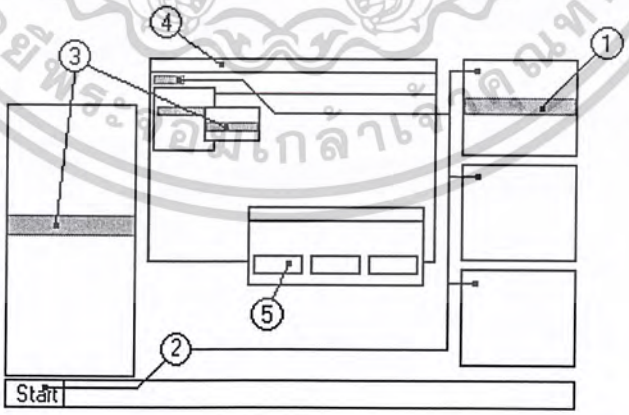
เนื่องจากปุ่มกดเป็น control ใน GUI ของวินโดว์ ซึ่งมีหมายเลขประจำตัว(ID) เฉพาะของแต่ละปุ่ม และการรับการกดปุ่มของ window ต่างๆ นั้นจะรับค่าหมายเลขประจำตัวมาพร้อมกับ Windows Message ที่เรียกว่า WM_COMMAND ทำให้เราสามารถจำลอง การกดปุ่มได้โดยส่ง WM_COMMAND ไปยังหน้าต่างนั้นๆ พร้อมกับหมายเลข ID ของปุ่มที่เราต้องการกด โดยค่า ID นี้มักอยู่ระหว่าง 1,000 ถึง 40,000



รูปที่ 3.7 ปุ่มบน Dialog box

3.1.5 การนำทุกส่วนมารวมกัน

เมื่อนำส่วนของการควบคุมทั้งหมดมารวมกัน ระบบจะมีลักษณะดังนี้



รูปที่ 3.8 ภาพรวมของโปรแกรมเมื่อนำส่วนควบคุมมารวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61369

จากรูปที่ 3.8 สามารถอธิบายแต่ละส่วนได้ดังนี้

(1) เมื่อตำแหน่งรับคำสั่งอยู่ที่หน้าต่างช่วยเหลือใดๆ ในขณะนี้สามารถเลื่อนแถบสีขึ้นลงเพื่อเลือกตัวเลือกที่เราต้องการ เช่น การเปลี่ยนหน้าต่างการทำงานหรือการเลือกเมนูให้โปรแกรมทำงานตามที่เราต้องการ

(2) เป็นการสั่งเพื่อเปลี่ยนตำแหน่งรับคำสั่ง ซึ่งในที่นี้มี 5 จุดด้วยกันคือ Start เมนู เมนูของแอพพลิเคชัน และหน้าจอช่วยเหลืออีก 3 จุด

(3) เหมือนกับข้อ 1 แต่ใช้กับส่วนที่เป็น เมนูของแอพพลิเคชัน และ Start เมนู

(4) เป็นการสั่งให้หน้าต่างที่เราเลือกเป็นหน้าต่างที่เราทำงาน สามารถเปลี่ยนขนาด และปิดโปรแกรมได้

(5) เมื่อมีการ pop-up หน้าต่างย่อยขึ้นมาโดยมีปุ่ม botton ให้เลือก ระบบจะไม่อนุญาตให้ทำการใดๆ นอกจากเลือกปุ่ม botton ดังกล่าวก่อน

จากส่วนที่ได้กล่าวมานี้จะถูกนำไปสร้างเป็นชุดคำสั่งโดยใช้ XML Grammar เพื่อการกำหนดคำสั่ง และ กฎต่างๆ ซึ่งอยู่ในหัวข้อถัดไป

3.2 ส่วนวิเคราะห์และการรู้จำคำพูด

ในส่วนการรู้จำนั้นใช้เครื่องมือ SAPI ที่อยู่ใน Speech SDK เป็นหลัก ซึ่ง SDK ตัวนี้มีรูปแบบการทำงานแบบฮิดเด็นมาร์คอฟโมเดลที่มีการวิเคราะห์คำพูดจากหน่วยเสียง โดยผู้ใช้จะต้องทำการเทรน SR engine ก่อนเพื่อเป็นการสร้างโมเดลของหน่วยเสียงขึ้นมา แล้วเก็บเป็นฐานข้อมูลของค่าพารามิเตอร์ต่างๆ ที่ใช้ในการรู้จำเสียงพูดของบุคคลนั้น โดยขั้นตอนการติดตั้ง Speech SDK และการเทรนนั้นดูได้จาก ภาคผนวก ก.

การออกแบบในส่วนนี้นั้น ทำได้โดยยกคำสั่งจากส่วนการควบคุมไมโครซอฟท์วินโดวส์มาจัดเป็นกลุ่มให้เหมาะสม เพื่อเขียนเป็น XML Grammar สำหรับใช้กับ Speech SDK โดย XML Grammar นี้เป็นตัวกำหนดไวยกรณ์ และกฎต่างๆ ให้กับ SR engine ที่อยู่ใน Microsoft Speech SDK ที่เรานำมาใช้ในโครงงานนี้ โดยคำสั่งทั้งหมด สามารถจัดเป็นกลุ่มได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Window

- Close : สั่งปิดหน้าต่างที่กำลังทำงานอยู่ปัจจุบัน
- Maximize : สั่งขยายหน้าจอที่กำลังทำงานเป็นแบบเต็มจอ
- Minimize : สั่งย่อหน้าจอของโปรแกรมที่กำลังทำงานเป็นไอคอนใน Task bar
- Restore : สั่งคืนขนาดหน้าจอที่กำลังทำงานเป็นขนาดตามปรกติ
- Stretch : สั่งให้ขึงหน้าจอที่กำลังทำงานให้พอดีกับหน้าจอช่วยเหลือ

Use

- Task List : สั่งเรียกใช้หน้าจอช่วยเหลือเกี่ยวกับการสลับหน้าจอทำงาน
- Menu Tree : สั่งเรียกใช้หน้าจอช่วยเหลือที่เกี่ยวกับเมนู
- Button List : สั่งเรียกใช้หน้าจอช่วยเหลือที่เกี่ยวกับปุ่มกดบนหน้าจอ
- Menu : สั่งให้ pop-up เมนูบาร์ของหน้าจอที่กำลังทำงาน
- Start Menu : เรียก start menu ของวินโดว์ขึ้นมาเพื่อสั่งรันโปรแกรม

Helper

- Hide : สั่งให้ซ่อนหน้าจอช่วยเหลือ
- Show : สั่งให้แสดงหน้าจอช่วยเหลือ

Cursor

- Up : เลื่อนตำแหน่ง cursor ขึ้น 1 ตำแหน่ง
- Down : เลื่อนตำแหน่ง cursor ลง 1 ตำแหน่ง
- Next : เลื่อนตำแหน่ง cursor ไปทางขวา 1 ตำแหน่ง หรือเข้าไปในโหนดย่อยของ Tree View Control
- Back : เลื่อนตำแหน่ง cursor ไปทางซ้าย 1 ตำแหน่ง หรือถอยออกจากโหนดย่อยใน Tree View Control
- Choose : เลือกไอเท็มที่ตำแหน่งที่ cursor กำลังชี้อยู่
- Escape : ยกเลิกการเรียก pop-up menu ต่างๆ

Confirm

- Yes : สั่งให้กดปุ่ม "Yes"
- No : สั่งให้กดปุ่ม "No"
- OK : สั่งให้กดปุ่ม "OK"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Cancel : สั่งให้กดปุ่ม "Cancel"
- Close : สั่งให้กดปุ่ม "Close"
- Open : สั่งให้กดปุ่ม "Open"
- ใช่ : สั่งให้กดปุ่ม "ใช่"
- ไม่ใช่ : สั่งให้กดปุ่ม "ไม่ใช่"
- ตกลง : สั่งให้กดปุ่ม "ตกลง"
- ยกเลิก : สั่งให้กดปุ่ม "ยกเลิก"
- ปิด : สั่งให้กดปุ่ม "ปิด"
- เปิด : สั่งให้กดปุ่ม "เปิด"

Exit Program : สั่งปิดโปรแกรม

โดยคำสั่งต่างๆ สามารถเปิด และ ปิดการตัดสินใจได้ โดยการเซ็ท ที่ Rule ใน Grammar ให้เป็น ACTIVE หรือ INACTIVE ได้จากตัวโปรแกรม จึงช่วยลดความผิดพลาดที่เกิดจากการ Recognition ได้ ไฟล์ XML Grammar ของโครงงานนี้ดูได้จากภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทดสอบการทำงานของโปรแกรม

4.1 การทดสอบการรู้จำ

โครงงานนี้ใช้ SR engine มาตรฐานซึ่งมากับชุดพัฒนา Speech SDK กับชุดของหน่วยเสียงมาตรฐานที่มีมาอยู่แล้วในชุดพัฒนา คือ หน่วยเสียงของภาษาอังกฤษ

โดยมีค่าคุณสมบัติของกลไกการรู้จำเสียงพูด(speech recognition engine) ดังนี้

ResourceUsage = 50 เป็นการระบุการใช้ Resource มีผลต่อการใช้ CPU ของ engine เมื่อใช้ resource เพิ่มขึ้น CPU จะรับภาระหนักมากขึ้นตามไปด้วย

HighConfidenceThreshold = 80 ค่า threshold ถูกใช้เพื่อแบ่งสัดส่วนค่าความมั่นใจเป็น 4 ส่วน คือ rejected, low, medium และ high โดยค่านี้บอกที่อยู่ของ high confidence ที่แบ่งช่วงของ high และ medium confidence ออกจากกัน

NormalConfidenceThreshold = 50 ค่านี้คั่นระหว่าง medium และ low confidence thresholds

LowConfidenceThreshold = 20 ค่านี้แบ่งช่วง low และ rejected confidence ออกจากกัน ถ้าค่า confidences ทั้งสามค่านี้เท่ากับ 0 อินพุตทุกคำที่เข้ามาจะถือว่าเป็น high confidence แต่ถ้าทั้งสามค่านั้นเป็น 100 ทุกคำจะเป็น low confidence

ResponseSpeed = 150 ms คือ ค่าที่กำหนดระยะเวลาให้ engine เงียบรอฟังก่อนที่จะทำการ recognition คำพูดที่กำลังพูดเข้าไป ค่านี้ถูกใช้เมื่อการ recognition ได้ผลลัพธ์ที่ชัดเจน

ComplexResponseSpeed = 500 ms เหมือนกับค่า ResponseSpeed แต่ค่านี้จะถูกใช้เมื่อการ recognition เกิดความสับสน

AdaptationOn = 1 กำหนดให้ใช้การโมเดลเสียง โดยมีค่าให้เลือกสองค่า คือ 1 กับ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ใช้คำสั่งในการทดลองดังนี้

1. window close
2. window maximize
3. window minimize
4. window restore
5. window stretch
6. use task list
7. use menu tree
8. use button list
9. use menu
10. use start menu
11. (cursor) up
12. (cursor) down
13. (cursor) next
14. (cursor) back
15. (cursor) escape
16. (cursor) choose
17. (confirm) yes
18. (confirm) no
19. (confirm) ok
20. (confirm) cancel

โดยในการทดลองช่วงหลังๆ จะมีการจัดกลุ่มคำสั่งดังนี้

ชุดคำสั่งปรับขนาด และ ปิดวินโดว์ {1,2,3,4,5}
ชุดคำสั่งเลือกหน้าต่างช่วยเหลือ {6,7,8}
ชุดคำสั่งเลื่อนเคอร์เซอร์ เลือก และ ยกเลิกเมนู {11,12,13,14,15,16}
ชุดคำสั่งการกดปุ่มยืนยัน {17,18,19,20}

เพื่อการรู้จำที่ถูกต้อง Microsoft Speech SDK 5.1 ได้เตรียมเครื่องมือในการเทรน และชุดของบทความตัวอย่างให้อ่านเพื่อเพิ่มความถูกต้องในการรู้จำของ speech recognition engine ซึ่งวิธีการเทรนนั้น สามารถดูได้จาก ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

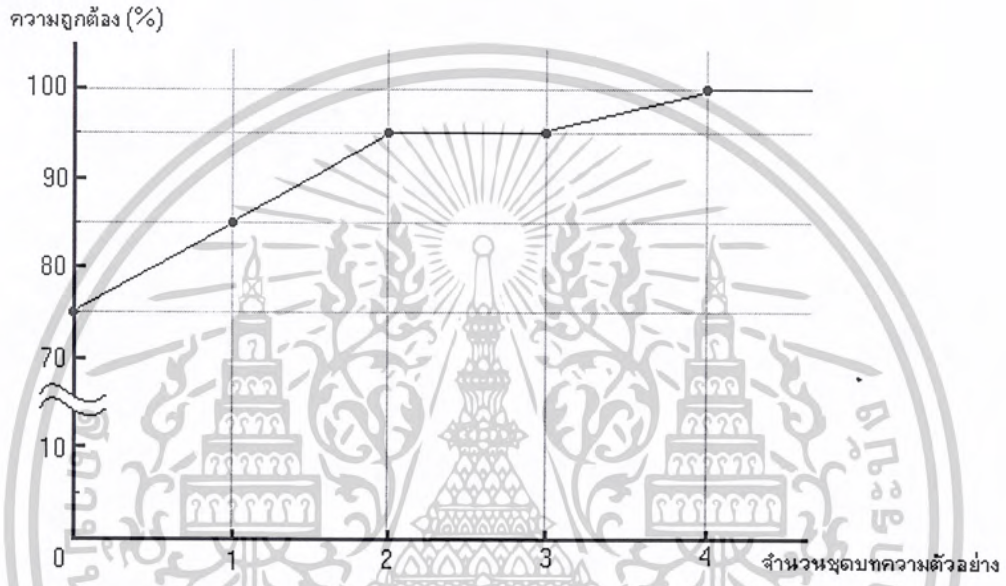
การทดลองต่อไปนี้จะเป็นการเปรียบเทียบความถูกต้องของการรู้จำจากการเทรนด้วยจำนวนบทความตัวอย่างที่ต่างกัน เพื่อหาจำนวนชุดตัวอย่างที่เหมาะสมในการเทรน ซึ่งผลการทดลองมีดังนี้

คำสั่งที่ใช้ทดสอบ คำสั่งที่	ผลการทดสอบในห้องที่ไม่มีเสียงรบกวน				
	ไม่ได้เทรน	เทรนด้วย ตัวอย่าง 1 ชุด	เทรนด้วย ตัวอย่าง 2 ชุด	เทรนด้วย ตัวอย่าง 3 ชุด	เทรนด้วย ตัวอย่าง 4 ชุด
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	15	15	15	5	5
6	9	7	6	6	6
7	7	7	7	7	7
8	9	8	8	8	8
9	9	9	9	9	9
10	9	10	10	10	10
11	11	11	11	11	11
12	12	12	12	12	12
13	18	13	13	13	13
14	14	14	14	14	14
15	15	17	15	17	15
16	16	16	16	16	16
17	17	17	17	17	17
18	18	18	18	18	18
19	19	19	19	19	19
20	20	20	20	20	20
ความถูกต้อง	75%	85%	95%	95%	100%

ตารางที่ 4.1 ผลการทดสอบการรู้จำโดยพิจารณาจำนวนชุดบทความที่ใช้เทรน

จาก ตารางที่ 4.1 พบว่าหลังจากการเทรนโดยใช้ชุดบทความตัวอย่างมากเกิน 2 ชุดขึ้นไปนั้น ให้ผลการรู้จำที่ดีขึ้นเพียงเล็กน้อย จำนวนที่เหมาะสมสำหรับการเทรนนั้น ใช้ชุดบทความตัวอย่างแค่ 2 ชุด ก็เพียงพอแล้วสำหรับการนำไปใช้งาน ผลจากการทดลองข้างต้นสามารถนำมาเขียนเป็นกราฟได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 กราฟแสดงผลการรู้จำจากการเทรนที่ต่างกัน

จากกราฟจะเห็นว่า จำนวนชุดตัวอย่างการเทรนที่ 2 กับ 3 ชุดนั้น ให้เปอร์เซนต์ความถูกต้องที่เท่ากัน และจะเพิ่มขึ้นอีกเป็น 100% จำนวนชุดตัวอย่าง 4 ชุด แต่นั้นก็ยังมีโอกาสที่การรู้จำจะผิดพลาดได้ในบางครั้ง จึงเกินความจำเป็นที่จะเทรนด้วยชุดตัวอย่างถึง 4 ชุด

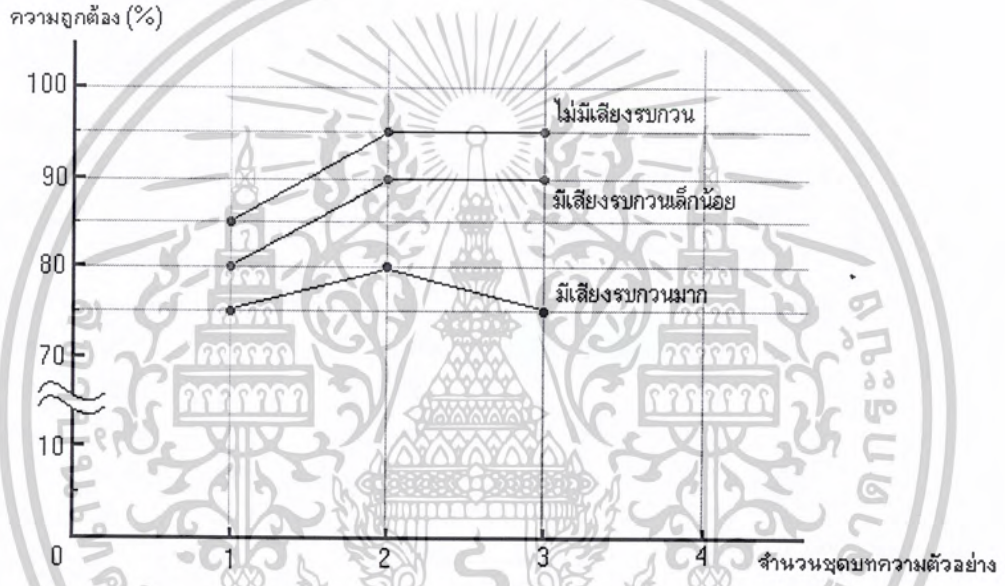
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองต่อไปจะเป็นการทดสอบ SR engine ในห้องที่มีเสียงรบกวน ด้วยชุดตัวอย่างการเทรน 3 ชุด เพื่อหาว่าจำนวนชุดตัวอย่างการเทรนมีผลทำให้การรู้จำสามารถทนต่อสภาพแวดล้อมที่มีเสียงรบกวนได้หรือไม่ ซึ่งได้ผลการทดลองดังนี้

คำสั่งที่ใช้ทดสอบคำสั่งที่	ผลการทดสอบในห้องที่มีเสียงรบกวน					
	ห้องที่มีเสียงรบกวนเล็กน้อย			ห้องที่มีเสียงรบกวนมาก		
	เทรนด้วยตัวอย่าง 1 ชุด	เทรนด้วยตัวอย่าง 2 ชุด	เทรนด้วยตัวอย่าง 3 ชุด	เทรนด้วยตัวอย่าง 1 ชุด	เทรนด้วยตัวอย่าง 2 ชุด	เทรนด้วยตัวอย่าง 3 ชุด
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	8	6	6	8	8	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	7	9	7	9	7	7
10	10	10	10	10	10	10
11	11	15	11	15	11	11
12	12	12	18	12	12	12
13	14	14	13	14	14	14
14	14	14	14	14	14	14
15	15	15	15	15	15	15
16	16	16	16	16	16	9
17	15	17	17	15	15	15
18	18	18	18	18	18	18
19	19	19	19	19	19	19
20	20	20	20	15	20	15
ความถูกต้อง	80%	90%	90%	75%	80%	75%

ตารางที่ 4.2 ผลการทดสอบการรู้จำในห้องที่มีเสียงรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 กราฟแสดงผลการรู้จำจากการเทรนที่ต่างกันในสภาพแวดล้อมต่างกัน

จากกราฟจะเห็นว่า จำนวนชุดตัวอย่างการเทรนที่ 2 ชุดนั้นมีแนวโน้มว่า เหมาะสมที่สุด สำหรับใช้เทรน ในทุกๆสภาพแวดล้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก SR engine ที่ใช้ในโครงงานนี้มีการทำงานของการรู้จำคำสั่งแบบต่อเนื่องทำให้เราสามารถจัดกลุ่มของคำสั่งได้ โดยจัดกลุ่มตามลักษณะร่วมกันบางอย่างของคำสั่ง เช่น การจัดการปรับขนาดวินโดว์ การเลื่อนเคอร์เซอร์ และการเรียกหน้าต่างช่วยเหลือ โดยใช้คำในพยางค์แรกของคำสั่งเหมือนกัน เช่น window, cursor และ use เป็นต้น ซึ่งตามความเป็นจริงในการออกคำสั่งกับระบบในเวลาใดเวลาหนึ่งนั้นเราจะใช้คำสั่งเพียงบางกลุ่มเท่านั้น ทำให้มีตัวเลือกในการตัดสินใจน้อยลง และการรู้จำคำสั่งมีความถูกต้องมากขึ้น สามารถทดลองให้เห็นผลได้ดังตารางต่อไปนี้

คำสั่งที่	ทุกคำสั่งอยู่ในระดับเดียวกัน	มีการจัดกลุ่มของคำสั่ง
1	1	1
2	2	2
3	3	3
4	4	4
5	15	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
ความถูกต้อง	95%	100%

ตารางที่ 4.3 ผลการทดสอบความถูกต้องของการรู้จำเมื่อจัดคำสั่งเป็นกลุ่ม

จากผลการทดลองพบว่าการจัดคำสั่งให้เป็นกลุ่มนั้น มีความถูกต้องในการรู้จำมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จากการทดลองข้างต้นพบว่าจำนวนการเทรนที่เหมาะสม คือ การเทรนด้วยชุดคำสั่งเพียง 2 ชุด โดยต่อไปนี้จะเป็นการทดสอบการใช้โปรแกรมโดยเทรนจากสภาพแวดล้อมต่างกัน คือ มีเสียงรบกวน และ ไม่มีเสียงรบกวน จากนั้นนำไปใช้กับสภาพแวดล้อมทั้งสองนี้ เพื่อหาว่าการเทรนในสภาพแวดล้อมที่มีเสียงรบกวน แล้วนำไปใช้ในสภาพแวดล้อมเดียวกันนั้น ให้ผลดีกว่าการเทรนจากสภาพแวดล้อมที่ปราศจากเสียงรบกวนหรือไม่

คำสั่งที่	ผลการทดลอง	
	ห้องที่ไม่มีเสียงรบกวน	ห้องที่มีเสียงรบกวน
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	8
7	7	9
8	8	8
9	9	6
10	10	10
11	11	11
12	12	14
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
ความถูกต้อง	100%	80%

ตารางที่ 4.4 ผลการทดสอบโปรแกรมโดยเทรนจากห้องที่ไม่มีเสียงรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่	ผลการทดลอง	
	ห้องที่ไม่มีเสียงรบกวน	ห้องที่มีเสียงรบกวน
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	8
7	7	7
8	8	8
9	9	7
10	10	10
11	11	11
12	12	14
13	13	14
14	14	14
15	15	15
16	16	16
17	17	17
18	18	19
19	19	19
20	20	20
ความถูกต้อง	100%	75%

ตารางที่ 4.5 ผลการทดสอบการรู้จำโดยเทรนจากห้องที่มีเสียงรบกวน

จากการทดลองพบว่าการเทรนในสภาพแวดล้อมที่ปราศจากเสียงรบกวนนั้นให้ผลการรู้จำที่ถูกต้องมากกว่าการเทรนจากสภาพแวดล้อมที่มีเสียงรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้งานกับโปรแกรมต่างๆ

การเรียกใช้เมนู

มีลักษณะการทำงานดังนี้

- เรียกเมนูของหน้าต่างขึ้นมา
- เลื่อนเคอร์เซอร์ไปยังรายการต่างๆในเมนู
- เลือกรายการ หรือ ยกเลิกการใช้เมนู

การควบคุมหน้าต่าง(ย่อ-ขยาย และ ปิดหน้าต่าง)

มีลักษณะการทำงานดังนี้

- สั่งให้หน้าจอวินโดว์ที่กำลังทำงานอยู่ขยายเต็มจอ ย่อลงขนาดปกติ ย่อลงจนเป็นไอคอน หรือ ขึงให้เข้ากับพอดีกับหน้าต่างช่วยเหลือที่อยู่ทางด้านขวาของจอ
- สั่งปิดหน้าจอวินโดว์ที่กำลังทำงานอยู่

การใช้ปุ่มในหน้าต่างย่อย

มีลักษณะการทำงานดังนี้

- สั่งให้กดปุ่มบนหน้าจอที่รอการยืนยัน เช่น Yes, No, OK และ Cancel

ชื่อโปรแกรม	การเรียกใช้เมนู	การควบคุมหน้าต่าง	การใช้ปุ่มในหน้าต่างย่อย
Paint	ได้	ได้	ได้
Notepad	ได้	ได้	ได้
MS Word	ไม่ได้	ได้	ไม่ได้
MS Exel	ไม่ได้	ได้	ไม่ได้
Windows Media Player	ได้	ได้	ได้
Adobe Photoshop 7.0	ได้	ได้	ได้
EditPluse 2.0	ได้	ได้	ได้
ACDSee 6.0	ไม่ได้	ได้	ได้
Windows Explorer	ได้	ได้	ได้
Internet Explorer	ได้	ได้	ได้

ตารางที่ 4.6 ผลการทดลองใช้งานกับโปรแกรมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองระหว่างการพัฒนาพบว่า

1. การรับคำสั่งมีความถูกต้องสูง อยู่ในระดับที่นำไปใช้งานได้
2. จำนวนชุดบทความตัวอย่างที่เหมาะสมสำหรับการเทรน คือ 2 ชุด
3. จำนวนชุดบทความตัวอย่างที่ใช้เทรน ถึงเพิ่มขึ้นมากเท่าไหร่ก็ไม่สามารถทำงานในสภาพแวดล้อมที่มีเสียงรบกวนมากๆ ได้
4. การเทรนที่ดีที่สุด ควรทำในสภาพแวดล้อมที่ไม่มีเสียงรบกวน
5. การรู้จำคำสั่งสามารถทำได้รวดเร็ว ประมาณ 1 วินาที หลังจากสิ้นสุดเสียงคำสั่ง
6. การจัดกลุ่มคำสั่งเป็นชุดๆ สามารถลดความผิดพลาดที่เกิดขึ้นในการรู้จำได้
7. ไม่สามารถใช้โปรแกรมที่พัฒนาขึ้นเรียกใช้เมนูที่เป็น รีบาร์คอนโทรล และ โฟลเดอร์วิวคอนโทรล ได้
8. การสลับหน้าจอการทำงานบางครั้งไม่สามารถย้าย focus ไปยังหน้าจอที่ต้องการจะทำงานได้ โดยหน้าต่างที่เราต้องการจะแค่กระพริบไตเติ้ลบาร์ แต่ไม่ขึ้นมาอยู่หน้าสุดเพื่อให้เราใช้งานตามต้องการได้

5.2 บทวิจารณ์และแนวทางพัฒนา

จากการทำงานของโปรแกรมที่พัฒนาขึ้นในโครงงานพบว่าสามารถใช้ฟังก์ชันการทำงานได้เกือบทั้งหมดของโปรแกรมบนวินโดว์ส่วนใหญ่ แต่ยังคงติดปัญหาเกี่ยวกับการทำงานบ้างอย่าง เช่น การสลับหน้าจอ ที่ในบางครั้งไม่สามารถย้าย focus ตามที่ออกคำสั่ง ซึ่งอาจเป็นผลมาจากการข้ามขั้นตอนบางอย่างในการย้ายการควบคุมของ Thread ที่เกี่ยวกับความปลอดภัยต่างๆ บนระบบวินโดวส์ ปัญหาของรีบาร์คอนโทรล และ โฟลเดอร์วิวคอนโทรล ซึ่งถ้าพัฒนาเพิ่มให้โปรแกรมสามารถใช้งานกับส่วนดังกล่าวได้จะทำให้ครอบคลุมการทำงานของวินโดวส์มากขึ้น นอกจากนี้ SR engine ไม่สามารถย้ายค่าพารามิเตอร์ต่างๆ ที่เกิดขึ้นจากการเทรนไปยังเครื่องอื่นได้ นั่นหมายถึงต้องเทรนใหม่ทุกครั้งที่ลงโปรแกรมในเครื่องใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาและปรับปรุงในขั้นต่อไป สามารถสรุปได้เป็นข้อๆ ดังนี้

1. ศึกษาระบบความปลอดภัยเกี่ยวกับการจัดการ process ต่างๆ ในระบบ เพื่อประสิทธิ-ภาพการทำงานที่ดีขึ้นของโปรแกรม
2. ศึกษาวิธีการติดต่อกับ รีบาร์คอนโทรล และ โฟลเดอร์วิวคอนโทรล
3. ศึกษาวิธีการย้ายพารามิเตอร์ที่เทรนไว้แล้วไปยังเครื่องอื่น
4. ปรับปรุงให้สามารถออกคำสั่งเป็นเสียงพูดภาษาไทยได้
5. เพิ่มลักษณะการทำงานพื้นฐาน เช่น พิมพ์ข้อความ หรือ วาดรูป ในบางโปรแกรมได้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

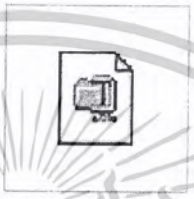
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การติดตั้งโปรแกรม และ ชุดเครื่องมือ Speech SDK

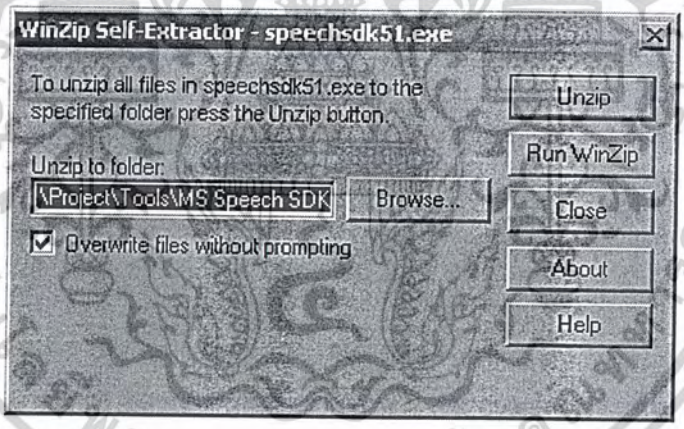
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้ง Speech SDK และโปรแกรม Voice to Command Agent 2

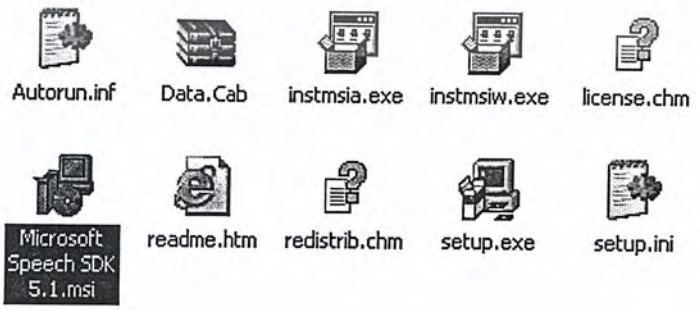
1. ดาวน์โหลดไฟล์ speechsdk51.exe ซึ่งมีขนาด 67.9 MB จาก www.microsoft.com หรือจากโฟลเดอร์ tools ในแผ่นติดตั้ง



2. ติดตั้ง Speech SDK เลือกโฟลเดอร์ที่จะทำการแตกไฟล์ติดตั้งลงไป



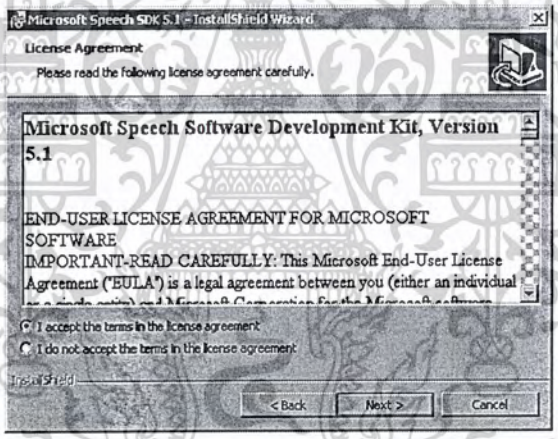
เมื่อแตกไฟล์เสร็จแล้วจะมีไฟล์ดังนี้



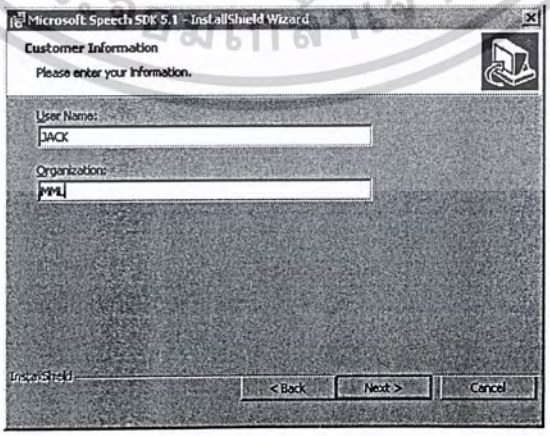
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกติดตั้งด้วยไฟล์ Microsoft Speech SDK 5.1.msi

กด Next จากนั้นตอบยอมรับข้อตกลงของไมโครซอฟท์

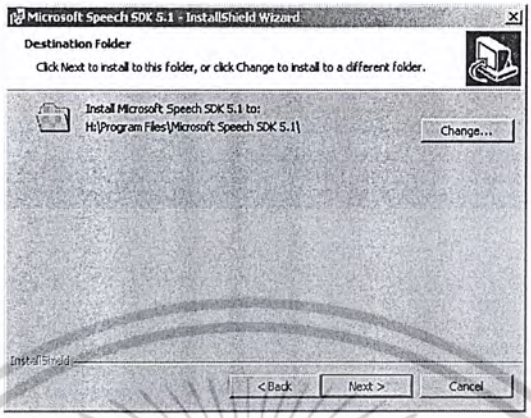


จากนั้นใส่ชื่อและบริษัท

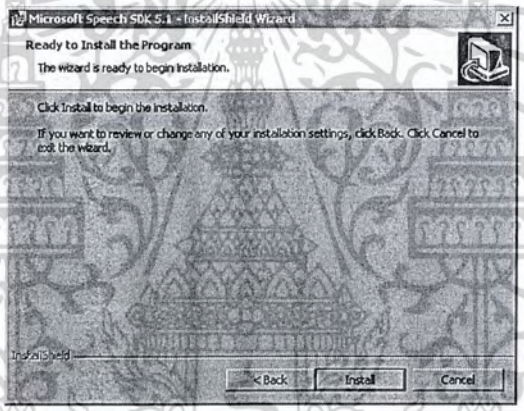


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

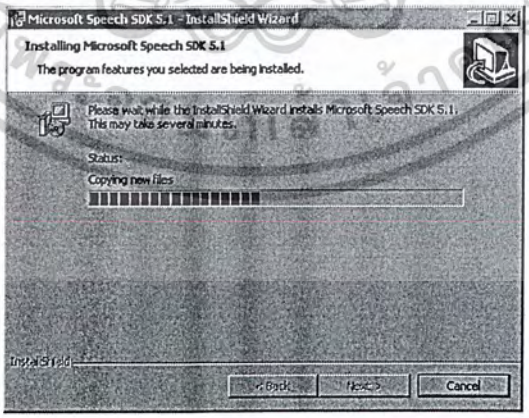
เลือกโฟลเดอร์ในที่นี้ใช้ค่า default จากนั้นกด Next



กด Install เพื่อยืนยันการติดตั้งอีกครั้งนึง

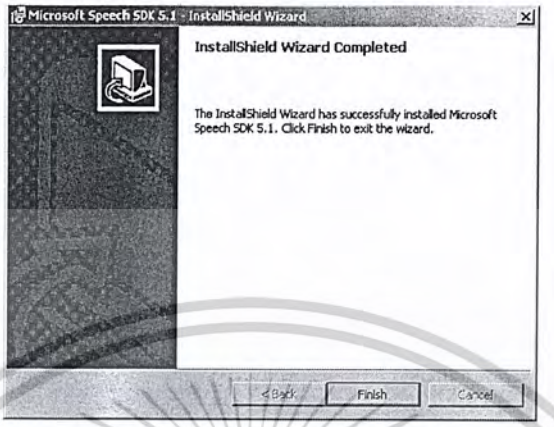


จากนั้นโปรแกรมก็จะเริ่มทำการติดตั้ง

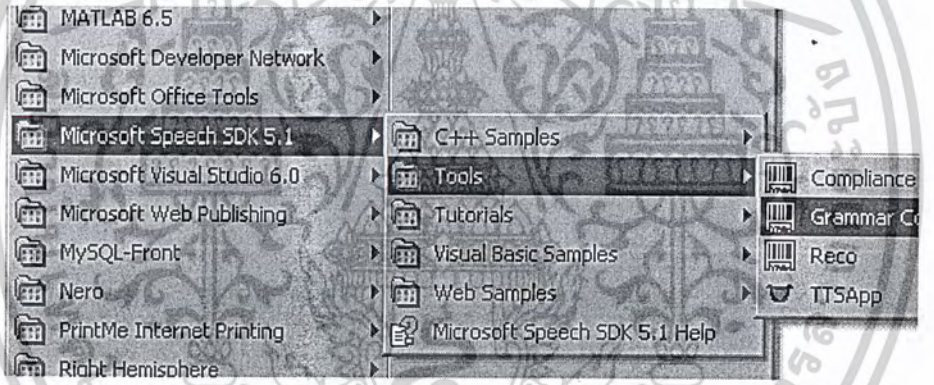


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

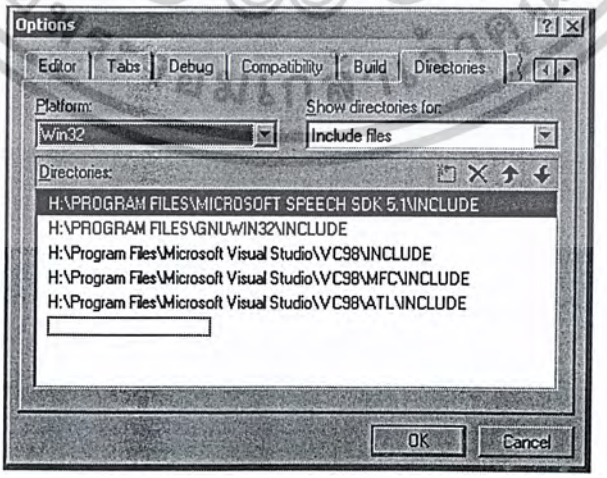
จนสิ้นสุดการติดตั้ง



เราก็จะมี Microsoft Speech SDK 5.1 ใน Start Menu ซึ่งมีเครื่องมือ ตัวอย่าง และเอกสารประกอบให้ใช้มากมาย

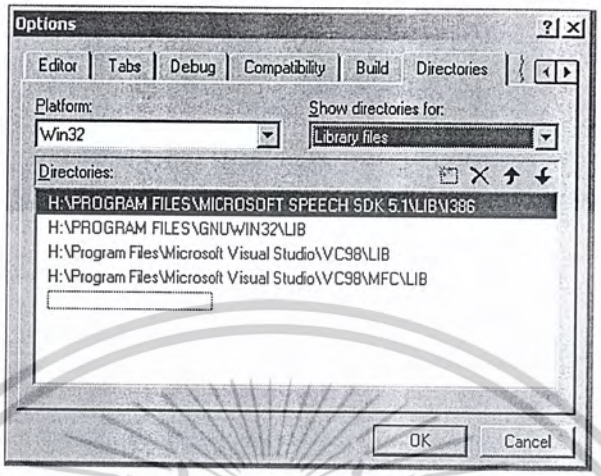


ระบุ Include files directory ของ Speech SDK ใน Visual C++ 6.0

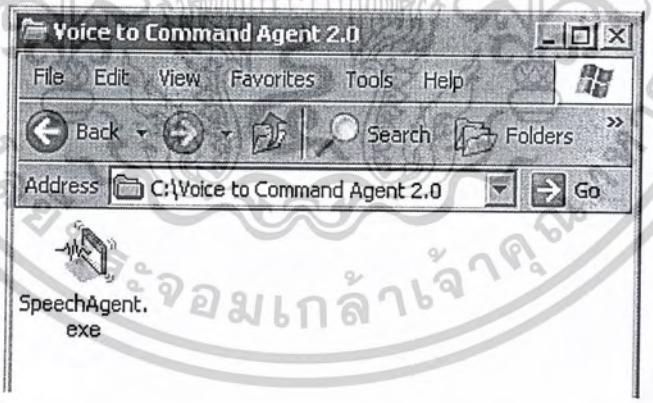


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบุ Library files directory ของ Speech SDK ใน Visual C++ 6.0

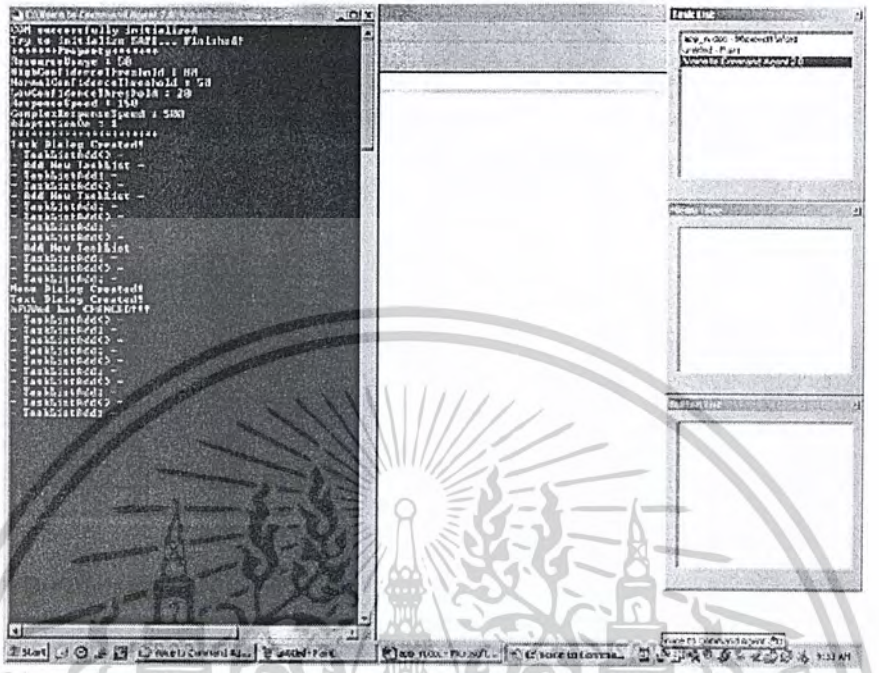


3. Copy ไฟล์โปรแกรมที่พัฒนาขึ้นในโครงงานนี้ คือ SpeechAgent.exe จากโฟลเดอร์ app\Debug ใน CD-ROM ไปไว้ในเครื่อง



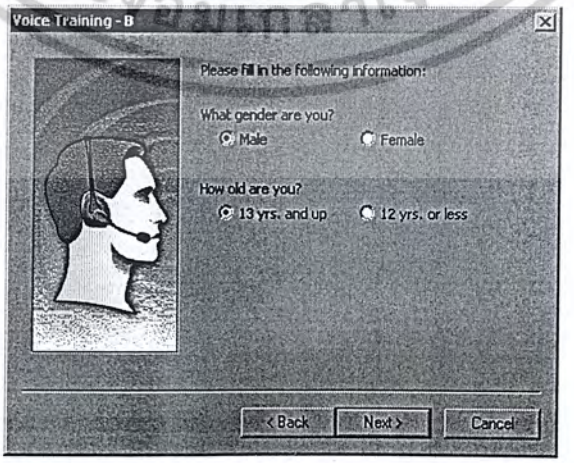
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรันโปรแกรมขึ้นมาจะมีหน้าจอการทำงานดังนี้



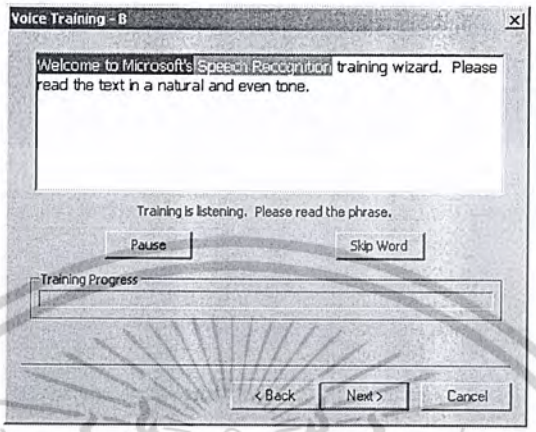
ที่ System Tray คลิกขวาที่ไอคอน Voice to Command Agent 2.0 จากนั้นเลือก Train Profile

จะขึ้นหน้าจอ Voice Training ขึ้นให้เราเลือก เพศ และ อายุ(โดยไมโครซอฟท์กำหนดไว้ให้เลือกได้สองช่วง คือ 13 ปี ขึ้นไป กับ 12 ปี หรือต่ำกว่า)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

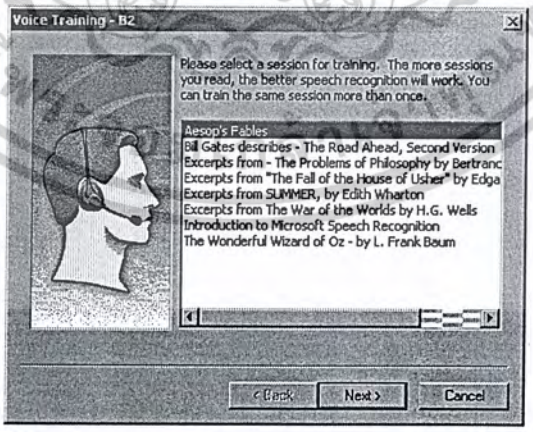
จากนั้นจะขึ้นข้อความตัวอย่างขึ้นมาให้อ่าน ถ้าคำไหนอ่านแล้วไม่ตอบสนองให้กด Skip Word ข้ามไปก่อนมันจะปรับค่าพารามิเตอร์จนสามารถยอมรับคำนี้ในภายหลัง



เมื่อ Training เสร็จแล้วเราสามารถทำการ Training เพิ่มเติมอีกได้ด้วยชุดบทความตัวอย่างที่ Microsoft เตรียมไว้ให้ โดยการเข้าไปที่ Control Panel เลือกไอคอน Speech

Speech

เลือก Train Profile... จะขึ้นหน้าต่างมาให้เลือกชุดบทความตัวอย่างให้เลือกใช้มากมาย ทั้งนี้ยิ่งมีการ Training มากเท่าไหร่ความถูกต้องของการรู้จำก็จะมากขึ้นตามไปด้วย โดยชุดบทความทั้งหมดนี้ ไมโครซอฟท์ได้จัดเตรียมไว้ให้อยู่แล้วไม่สามารถเพิ่มเองได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

XML Grammar ของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!-// filename : grammar.xml ////////////////////////////////////////////////////////////////-->
<GRAMMAR LANGID="409">
	<DEFINE>
		<ID NAME="VID_One" VAL="1"/>
		<ID NAME="VID_Two" VAL="2"/>
		<ID NAME="VID_Three" VAL="3"/>
		<ID NAME="VID_Four" VAL="4"/>
		<ID NAME="VID_Five" VAL="5"/>
		<ID NAME="VID_Six" VAL="6"/>
		<ID NAME="VID_Seven" VAL="7"/>
		<ID NAME="VID_Eight" VAL="8"/>
		<ID NAME="VID_Nine" VAL="9"/>
		<ID NAME="VID_Ten" VAL="10"/>
		<ID NAME="VID_Eleven" VAL="11"/>
		<ID NAME="VID_Twelve" VAL="12"/>
		<ID NAME="VID_Zero" VAL="13"/>
		<ID NAME="VID_Twen" VAL="20"/>
		<ID NAME="VID_A" VAL="65"/>
		<ID NAME="VID_B" VAL="66"/>
		<ID NAME="VID_C" VAL="67"/>
		<ID NAME="VID_D" VAL="68"/>
		<ID NAME="VID_E" VAL="69"/>
		<ID NAME="VID_F" VAL="70"/>
		<ID NAME="VID_G" VAL="71"/>
		<ID NAME="VID_H" VAL="72"/>
		<ID NAME="VID_I" VAL="73"/>
		<ID NAME="VID_J" VAL="74"/>
		<ID NAME="VID_K" VAL="75"/>
		<ID NAME="VID_L" VAL="76"/>
		<ID NAME="VID_M" VAL="77"/>
		<ID NAME="VID_N" VAL="78"/>
		<ID NAME="VID_O" VAL="79"/>
		<ID NAME="VID_P" VAL="80"/>
		<ID NAME="VID_Q" VAL="81"/>
		<ID NAME="VID_R" VAL="82"/>
		<ID NAME="VID_S" VAL="83"/>
		<ID NAME="VID_T" VAL="84"/>
		<ID NAME="VID_U" VAL="85"/>
		<ID NAME="VID_V" VAL="86"/>
		<ID NAME="VID_W" VAL="87"/>
		<ID NAME="VID_X" VAL="88"/>
		<ID NAME="VID_Y" VAL="89"/>
		<ID NAME="VID_Z" VAL="90"/>

		<ID NAME="VID_Run" VAL="91"/>
		<ID NAME="VID_Open" VAL="92"/>
		<ID NAME="VID_Copy" VAL="93"/>
		<ID NAME="VID_Past" VAL="94"/>
		<ID NAME="VID_Delete" VAL="95"/>

		<ID NAME="VID_Yes" VAL="96"/>
		<ID NAME="VID_No" VAL="97"/>
		<ID NAME="VID_OK" VAL="98"/>
		<ID NAME="VID_Cancel" VAL="99"/>
		<ID NAME="VID_Yes_TH" VAL="100"/>
		<ID NAME="VID_No_TH" VAL="101"/>
		<ID NAME="VID_OK_TH" VAL="102"/>
		<ID NAME="VID_Cancel_TH" VAL="103"/>
		<ID NAME="VID_Close_TH" VAL="104"/>
		<ID NAME="VID_Open_TH" VAL="105"/>

		<ID NAME="VID_NumTeen" VAL="111"/>
		<ID NAME="VID_NumTy" VAL="112"/>
		<ID NAME="VID_Thir" VAL="113"/>
		<ID NAME="VID_Fif" VAL="115"/>

		<ID NAME="VID_Choice" VAL="151"/>
		<ID NAME="VID_Number" VAL="201"/>
		<ID NAME="VID_NumBeginWord" VAL="202"/>
		<ID NAME="VID_NumSubWord" VAL="203"/>
		<ID NAME="VID_NumEndWord" VAL="204"/>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <ID NAME="VID_NumSubWord2" VAL="205"/>
        <ID NAME="VID_Confirm" VAL="301"/><!-- ãªé¡Ñ° Yes ¡Ñ° No -->
        <ID NAME="VID_Quick" VAL="302"/><!-- ãªé¡Ñ° A-Z -->
        <ID NAME="VID_Item" VAL="303"/><!-- ãªé¡Ñ° Run,Open...ÏÅÏ-->
        <ID NAME="VID_Enter" VAL="304"/>
        <ID NAME="VID_Back" VAL="305"/>
        <ID NAME="VID_Next" VAL="306"/>

        <ID NAME="VID_Exit" VAL="309"/>
        <ID NAME="VID_BrowserOFF" VAL="310"/>
        <ID NAME="VID_BrowserON" VAL="311"/>

        <ID NAME="VID_Window" VAL="1000"/>
        <ID NAME="VID_Close" VAL="1001"/>
        <ID NAME="VID_Minimize" VAL="1002"/>
        <ID NAME="VID_Maximize" VAL="1003"/>
        <ID NAME="VID_Restore" VAL="1004"/>
        <ID NAME="VID_Stretch" VAL="1005"/>

        <ID NAME="VID_HelperHIDE" VAL="1100"/>
        <ID NAME="VID_HelperSHOW" VAL="1101"/>

        <ID NAME="VID_Use" VAL="1102"/>
        <ID NAME="VID_MenuDlg" VAL="1103"/>
        <ID NAME="VID_TaskDlg" VAL="1104"/>
        <ID NAME="VID_CmndDlg" VAL="1105"/>
        <ID NAME="VID_StartProg" VAL="1106"/>
        <ID NAME="VID_Menu" VAL="1107"/>

        <ID NAME="VID_Cursor" VAL="1110"/>
        <ID NAME="VID_Up" VAL="1111"/>
        <ID NAME="VID_Down" VAL="1112"/>
        <ID NAME="VID_Left" VAL="1113"/>
        <ID NAME="VID_Right" VAL="1114"/>
        <ID NAME="VID_Choose" VAL="1115"/>
        <ID NAME="VID_Escape" VAL="1116"/>
        <ID NAME="VID_Tab" VAL="1117"/>
        <ID NAME="VID_HCursor" VAL="1118"/>
        <ID NAME="VID_SCursor" VAL="1119"/>

        <ID NAME="VID_Node" VAL="1120"/>
        <ID NAME="VID_Expand" VAL="1121"/>
        <ID NAME="VID_Collapse" VAL="1122"/>

    </DEFINE>
<!-////////////////////////////////////////////////////////////////////////////////////////////////////////////////-->
    <RULE ID="VID_Choice"  TOPLEVEL="INACTIVE">
        <L PROPID="VID_Choice">
            <P VAL="VID_OK">O.K.</P>
            <P VAL="VID_Cancel">Cancel</P>
            <RULEREF REFID="VID_NumEndWord" />
            <P VAL="VID_Zero">Zero</P>
        </L>
    </RULE>

    <RULE ID="VID_Number" TOPLEVEL="INACTIVE">
        <P PROPID="VID_Number">Number</P>
        <RULEREF REFID="VID_NumBeginWord" />
</RULE>

    <RULE ID="VID_NumBeginWord" TOPLEVEL="INACTIVE">
    <L PROPID="VID_NumBeginWord">
        <P VAL="VID_One"> one</P>
        <P VAL="VID_Two">two</P>
        <P VAL="VID_Three">three</P>
            <P VAL="VID_Four">
                <P>four</P>
                <O><RULEREF REFID="VID_NumSubWord" /></O>
            </P>
            <P VAL="VID_Five">five</P>
            <P VAL="VID_Six">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        <P>six</P>
                        <O><RULEREF REFID="VID_NumSubWord" /></O>
                    </P>
                    <P VAL="VID_Seven">
                        <P>seven</P>
                        <O><RULEREF REFID="VID_NumSubWord" /></O>
                    </P>
                    <P VAL="VID_Eight">
                        <P>eight</P>
                        <O><RULEREF REFID="VID_NumSubWord" /></O>
                    </P>
                    <P VAL="VID_Nine">
                        <P>nine</P>
                        <O><RULEREF REFID="VID_NumSubWord" /></O>
                    </P>
                    <P VAL="VID_Ten">ten</P>
                    <P VAL="VID_Eleven">eleven</P>
                    <P VAL="VID_Twelve">twelve</P>
                    <P VAL="VID_Twen">
                        <P>twen</P>
                        <RULEREF REFID="VID_NumSubWord2" />
                    </P>
                    <P VAL="VID_Thir">thir
                        <RULEREF REFID="VID_NumSubWord" />
                    </P>
                    <P VAL="VID_Fif">fif
                        <RULEREF REFID="VID_NumSubWord" />
                    </P>
        </L>
    </RULE>

    <RULE ID="VID_NumSubWord" >
     <L PROPID="VID_NumSubWord">
          <P VAL="VID_NumTeen">teen</P>
          <P VAL="VID_NumTy">
              <P>tie</P>
              <O><RULEREF REFID="VID_NumEndWord" /></O>
          </P>
     </L>
    </RULE>

    <RULE ID="VID_NumSubWord2" >
        <P PROPID="VID_NumSubWord2">tie</P>
        <O><RULEREF REFID="VID_NumEndWord" /></O>
    </RULE>

    <RULE ID="VID_NumEndWord" >
        <L PROPID="VID_NumEndWord">
            <P VAL="VID_One">one</P>
            <P VAL="VID_Two">two</P>
            <P VAL="VID_Three">three</P>
                <P VAL="VID_Four">four</P>
                <P VAL="VID_Five">five</P>
                <P VAL="VID_Six">six</P>
                <P VAL="VID_Seven">seven</P>
                <P VAL="VID_Eight">eight</P>
                <P VAL="VID_Nine">nine</P>
        </L>
    </RULE>
<!--////////////////////////////////////////////////////////////////////////////////////////////-->
        <RULE ID="VID_Confirm" TOPLEVEL="ACTIVE">
            <L PROPID="VID_Confirm">
                <P VAL="VID_Yes">Yes</P>
                <P VAL="VID_No">No</P>
                <P VAL="VID_OK">OK</P>
                <P VAL="VID_Cancel">Cancel</P>
                <P VAL="VID_Close">Close</P>
                <P VAL="VID_Open">Open</P>
                <P VAL="VID_Yes_TH">Chai</P>
                <P VAL="VID_No_TH">Mai Chai</P>
                <P VAL="VID_OK_TH">Tok Long</P>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        <P VAL="VID_Cancel_TH">Yok Lerg</P>
                        <P VAL="VID_Close_TH">Pid</P>
                        <P VAL="VID_Open_TH">Perd</P>
                </L>
        </RULE>
        <RULE ID="VID_Exit" TOPLEVEL="ACTIVE">
                <L>
                        <P>Exit Program</P>
                        <P>Program Exit</P>
                        <P>Program Quit</P>
                </L>
        </RULE>
        <RULE ID="VID_Window" TOPLEVEL="INACTIVE">
                <P>Window</P>
                <L PROPID="VID_Window">
                        <P VAL="VID_Close">Close</P>
                        <P VAL="VID_Minimize">Minimize</P>
                        <P VAL="VID_Maximize">Maximize</P>
                        <P VAL="VID_Restore">Restore</P>
                        <P VAL="VID_Stretch">Stretch</P>
                        <P VAL="VID_Next">Next</P>
                </L>
        </RULE>
        <RULE ID="VID_HelperHIDE" TOPLEVEL="INACTIVE">
                <O>I</O>
                <P>Don't need Helper</P>
        </RULE>
        <RULE ID="VID_HelperSHOW" TOPLEVEL="INACTIVE">
                <P>Use Helper</P>
        </RULE>
        <RULE ID="VID_Use" TOPLEVEL="INACTIVE">
                <P>Use</P>
                <L PROPID="VID_Use">
                        <P VAL="VID_MenuDlg">Menu Tree</P>
                        <P VAL="VID_TaskDlg">Task List</P>
                        <P VAL="VID_CmndDlg">Button List</P>
                        <P VAL="VID_StartProg">Start Menu</P>
                        <P VAL="VID_Menu">Menu</P>
                </L>
        </RULE>
        <RULE ID="VID_HCursor" TOPLEVEL="INACTIVE">
                <L PROPID="VID_HCursor">
                        <P VAL="VID_Up">Up</P>
                        <P VAL="VID_Down">Down</P>
                        <P VAL="VID_Left">Back</P>
                        <P VAL="VID_Right">Next</P>
                        <P VAL="VID_Choose">Choose</P>
                </L>
        </RULE>
<!-////////////////////////////////////////////////////////////////////////////////////////////////////////////////////-->
        <RULE ID="VID_SCursor" TOPLEVEL="INACTIVE">
                <L PROPID="VID_SCursor">
                        <P VAL="VID_Up">Up</P>
                        <P VAL="VID_Down">Down</P>
                        <P VAL="VID_Left">Left</P>
                        <P VAL="VID_Right">Right</P>
                        <P VAL="VID_Choose">Choose</P>
                </L>
        </RULE>
        <RULE ID="VID_Cursor" TOPLEVEL="INACTIVE">
                <L PROPID="VID_Cursor">
                        <P VAL="VID_Up">Up</P>
                        <P VAL="VID_Down">Down</P>
                        <P VAL="VID_Left">Back</P>
                        <P VAL="VID_Right">Next</P>
                        <P VAL="VID_Choose">Choose</P>
                        <P VAL="VID_Escape">Escape</P>
                        <P VAL="VID_Tab">Tab</P>
                </L>
        </RULE>
</GRAMMAR>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

รวินทร์ วิรัชพินทุ สุวารี เหลือเพิ่มพร. ระบบสั่งงานด้วยเสียง
ปริญญานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2546.

ฐนียา สัตยพานิช. ระบบรู้จำเสียงภาษาไทยชนิดปรับตัวเข้ากับผู้ใช้แบบเรียลไทม์
วิทยานิพนธ์ มหาวิทยาลัยเกษตรศาสตร์,2544.

Jeffrey Adam Bilmes. Natural Statistical Models for Automatic Speech Recognition
INTERNATIONAL COMPUTER SCIENCE INSTITUTE, October 1999

MSDN Library – January 2000

Microsoft Speech SDK 5.1 Document

http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้