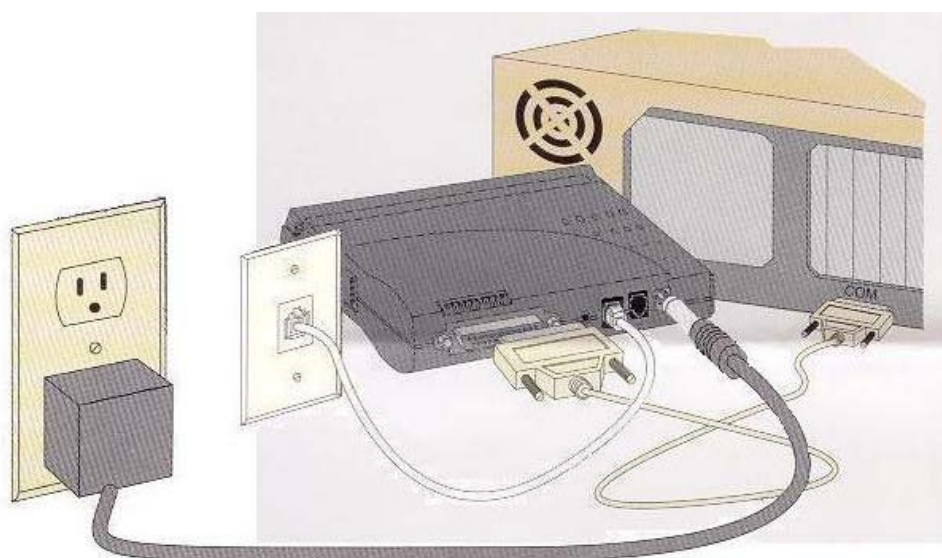


บทที่ 3

วิธีการติดต่อกับระบบโทรศัพท์

3.1 การเชื่อมต่ออุปกรณ์

การเชื่อมต่ออุปกรณ์ของระบบ โดยนำโมเด็มเสียงมาต่อกับพอร์ตอนุกรมสื่อสาร (Serial Port) ของเครื่องคอมพิวเตอร์ ด้วยสายอนุกรม (RS-232 Serial Cable) และต่อสายโทรศัพท์เข้ากับโมเด็มเสียง ดังภาพประกอบที่ 3.1



ภาพประกอบที่ 3.1 การเชื่อมต่ออุปกรณ์ของระบบ

3.2 หลักการพัฒนาโปรแกรมเพื่อติดต่อกับระบบโทรศัพท์

การเขียนโปรแกรมติดต่อกับระบบโทรศัพท์ สำหรับใช้ในงานสอบถามข้อมูลนั้น ฟังก์ชันที่สำคัญคือ โปรแกรมจะต้องสามารถรับสัญญาณสายโทรเข้าได้ รับสัญญาณกดปุ่มบนแป้นโทรศัพท์ได้ และเล่นเสียงออกทางคู่สายโทรศัพท์ได้ ซึ่งจะเขียนโปรแกรมด้วยภาษา Visual Basic 6.0 [Balena, 1999] โดยใช้ TAPI 3.0 Type Library ที่รองรับการทำงานกับระบบโทรศัพท์ บนระบบ Microsoft Windows 2000 มีรายละเอียดขั้นตอนดังต่อไปนี้

1. เริ่มต้นการทำงาน (Initialize TAPI)
2. การเชื่อมต่ออุปกรณ์โมเด็ม (Select An Address)
3. การเลือกลงทะเบียนเหตุการณ์โทรศัพท์ที่ต้องการรับ (Register Events)
4. การจัดการรับเหตุการณ์สายโทรเข้า (Handling An Incoming Call Event)

5. การจั้ดตั้งระบบให้ตรวจจับสัญญาณกดปุ่มบนแป้นโทรศัพท์ (Setting Digit DTMF Detection)
6. การตอบรับโทรศัพท์ (Answering A Call)
7. การจัดการรับเหตุการณ์กดปุ่มบนแป้นโทรศัพท์ (Handling A Digit Detection Event)
8. การเล่นเสียง (Playback Sound)
9. การตัดการโทรติดต่อ (Disconnect A Call)
10. สิ้นสุดการทำงาน (Shutdown TAPI)

3.2.1 เริ่มต้นการทำงาน

ขั้นตอนแรกของการเขียนโปรแกรมด้วย TAPI จะเริ่มต้นด้วยการสร้างออบเจ็กต์ TAPI ขึ้นมาจากอินเตอร์เฟซชื่อ ITTAPI แล้วให้ออบเจ็กต์ TAPI เรียกใช้เมธอดชื่อ Initialize เพื่อเตรียมหน่วยความจำและทรัพยากรอื่นๆ ที่ต้องการใช้งานในเครื่องคอมพิวเตอร์

3.2.2 การติดต่ออุปกรณ์โมเด็ม

ขั้นตอนนี้จะเลือกหาตำแหน่งของโมเด็มเสียงที่ต่ออยู่กับเครื่องคอมพิวเตอร์ โดยการใช้ออบเจ็กต์ที่สร้างขึ้นมาจากอินเตอร์เฟซชื่อ ITCollection, ITMediaSupport, และ ITAddress มีรายละเอียดขั้นตอนดังนี้

1. ออบเจ็กต์ TAPI เรียกใช้เมธอดชื่อ Addresses ทำให้ได้ค่าตำแหน่งของแอดเดรสต่างๆ ของอุปกรณ์ที่ใช้ติดต่อสื่อสารกับระบบคอมพิวเตอร์ทั้งหมด แล้วส่งค่าไปให้ออบเจ็กต์ชื่อ ITCollection เก็บค่าไว้

2. ออบเจ็กต์ชื่อ ITCollection ใช้เมธอดชื่อ Count นับจำนวนของแอดเดรสที่มีอยู่ทั้งหมด แล้วเรียกใช้เมธอด Item เพื่อนำค่าของตำแหน่งแอดเดรสแต่ละตำแหน่ง มาแทนด้วยค่าดัชนี (Index) แล้วส่งค่าต่อไปให้ออบเจ็กต์ชื่อ ITMediaSupport

3. ออบเจ็กต์ชื่อ ITMediaSupport ใช้เมธอดชื่อ QueryMediaType เพื่อค้นหาตำแหน่งแอดเดรสที่มีลักษณะของสื่อเป็นแบบเสียง (Media Type Audio) โดยกำหนดค่าพารามิเตอร์ IMediaType ของเมธอดนี้เป็น "TAPIMEDIATYPE_AUDIO" แล้วทำการค้นหาไปที่ละตำแหน่งดัชนีของแอดเดรส ที่เก็บไว้ในออบเจ็กต์ชื่อ ITCollection จนถึงตำแหน่งดัชนีตัวสุดท้าย

4. เมื่อออบเจกต์ ITMediaSupport พบตำแหน่งแอดเดรสที่เป็นแบบ “TAPIMEDIATYPE_AUDIO” แล้ว จะส่งค่าที่ได้ไปให้กับออบเจกต์ IAddress เมื่อถึงขั้นตอนนี้ ก็จะได้ Address Object ที่มีตำแหน่งของโมเด็มเสียงที่สามารถติดต่อใช้งานได้

3.2.3 การเลือกลงทะเบียนเหตุการณ์โทรศัพท์ที่ต้องการรับ

ในขั้นตอนนี้จะกำหนดการรับเหตุการณ์ทางโทรศัพท์ที่ระบบต้องการ ได้แก่

1. เหตุการณ์สายโทรเรียกเข้า (TE_CALLNOTIFICATION)
2. เหตุการณ์สถานะการโทรที่เปลี่ยนแปลง (TE_CALLSTATE) ซึ่งได้แก่สถานะ มีสัญญาณกระดิ่งดัง (CS_OFFERING), และ ตัดการโทรติดต่อ (CS_DISCONNECTED)

3. เหตุการณ์ DTMF ของการกดปุ่มบนเป็นโทรศัพท์ (TE_DIGITEVENT)

มีรายละเอียดขั้นตอนนี้

1. ออบเจกต์ TAPI ใช้เมธอดชื่อ EventFilter เพื่อกรองเลือกเหตุการณ์ทางโทรศัพท์ต่างๆ ที่ต้องการรับ สำหรับงานวิจัยนี้จะเลือก 3 เหตุการณ์ที่กล่าวมาข้างต้น

2. ออบเจกต์ TAPI ใช้เมธอดชื่อ RegisterCallNotifications เพื่อตกลงรับฟังสัญญาณการโทรเข้าที่ตำแหน่งแอดเดรสของอุปกรณ์โมเด็มที่ได้จาก Address Object

หมายเหตุ: เมธอดชื่อ RegisterCallNotifications จะส่งค่า (Return Value) ของตัวแปร glRegister กลับมา ซึ่งออบเจกต์ TAPI จะนำไปใช้ในเมธอดชื่อ UnregisterNotifications ในภายหลังอีกครั้ง เพื่อทำการยกเลิกการรับเหตุการณ์ของสายนี้ เมื่อได้เสร็จสิ้นการติดต่อไปแล้ว และจากนั้นจะเรียกเมธอด RegisterCallNotifications ใหม่อีกครั้ง เพื่อให้ระบบพร้อมจะรับฟังเหตุการณ์ของสายโทรเข้าใหม่ได้ต่อไป

3.2.4 การจัดการรับเหตุการณ์สายโทรเข้า

ในขั้นตอนนี้ เมื่อมีเหตุการณ์สายโทรเรียกเข้ามา ระบบจะสร้าง Call Object เพื่อควบคุมการติดต่อขึ้น มีรายละเอียดขั้นตอนนี้

1. สร้างส่วนโปรแกรมย่อย ที่ทำงานรับเหตุการณ์ของ TAPI โดยมี pEvent เป็นออบเจกต์ของเหตุการณ์ และมีตัวแปร TapiEvent ทำหน้าที่แสดงเหตุการณ์ที่เกิดขึ้น ถ้าหาก TapiEvent เป็น TE_CALLNOTIFICATION แล้ว จะให้ออบเจกต์ชื่อ pEvent ส่งค่าไปยังออบเจกต์ ICallNotificationEvent เพื่อรับเหตุการณ์ของสายโทรเข้าเอาไว้

2. ออบเจกต์ ITCallNotificationEvent ใช้เมธอดชื่อ Call เพื่อส่งค่าของเหตุการณ์สายโทรเข้าที่เกิดขึ้นนี้ ให้กับออบเจกต์ชื่อ ITCallInfo ซึ่งเป็นอินเตอร์เฟซที่ใช้เก็บข้อมูลเกี่ยวกับการโทรที่เกิดขึ้นเอาไว้

3. นำออบเจกต์ ITCallInfo ส่งค่าไปให้กับ ออบเจกต์ ITBasicCallControl ซึ่งเป็น Call Object ที่ใช้ในการควบคุมการติดต่อ ทั้งการรับสายโทรเข้า และการยกเลิกสายโทรเข้า

3.2.5 การจัดตั้งระบบให้ตรวจจับสัญญาณกดปุ่มบนแป้นโทรศัพท์

1. จากในส่วนโปรแกรมย่อย สำหรับรับเหตุการณ์ของ TAPI ถ้าหาก TapiEvent เป็น TE_CALLSTATE แล้ว จะนำออบเจกต์ชื่อ pEvent ส่งค่าไปให้กับออบเจกต์ชื่อ ITCallStateEvent เพื่อรับค่าเหตุการณ์สถานะของการโทรเอาไว้

2. ออบเจกต์ชื่อ ITCallStateEvent ใช้เมธอดชื่อ Call เพื่อส่งค่าของเหตุการณ์สถานะของการโทรที่เกิดขึ้นนี้ ให้กับออบเจกต์ชื่อ ITCallInfo ซึ่งเป็นอินเตอร์เฟซที่ใช้เก็บข้อมูลเกี่ยวกับการโทรที่เกิดขึ้นเอาไว้

3. ออบเจกต์ ITCallStateEvent ใช้เมธอด State เพื่อดูค่าสถานะของการโทรที่เกิดขึ้น ถ้าสถานะเป็น CS_OFFERING แสดงว่ามีสัญญาณกระดิ่งเข้ามาแล้ว จะนำออบเจกต์ ITCallInfo ส่งค่าไปให้กับ ออบเจกต์ ITLegacyCallMediacontrol ซึ่งเป็นอินเตอร์เฟซที่สนับสนุนงานประยุกต์ที่ต้องการติดต่อกับอุปกรณ์โดยตรง ในที่นี้เพื่อที่จะให้ระบบมีความสามารถตรวจจับสัญญาณกดปุ่มบนแป้นโทรศัพท์ได้

4. ออบเจกต์ ITLegacyCallMediacontrol ใช้เมธอด DetectDigits เพื่อกำหนดการตรวจจับรูปแบบของสัญญาณกดปุ่มของโทรศัพท์ ว่าเป็นแบบหมุน (Pulse) หรือ แบบกดปุ่ม (DTMF) โดยงานวิจัยนี้ใช้โทรศัพท์แบบกดปุ่ม จึงกำหนดค่าพารามิเตอร์ digitmode ให้เป็น LINEDIGITMODE_DTMFEND

3.2.6 การตอบรับโทรศัพท์

ในขั้นตอนนี้ ออบเจกต์ ITBasicCallControl จะใช้เมธอด Answer เพื่อรับสายโทรเข้า แล้วจะใช้ออบเจกต์ MMControl เพื่อเล่นเสียง (Greeting) กล่าวสวัสดีต้อนรับผู้ที่โทรเข้ามาและแจ้งรายการหลักให้ทราบ รายละเอียดการเล่นเสียงอยู่ในขั้นตอนที่ 3.2.8

3.2.7 การจัดการรับเหตุการณ์กดปุ่มบนแป้นโทรศัพท์

1. ในส่วนโปรแกรมย่อย สำหรับรับเหตุการณ์ของ TAPI ถ้าหาก TapiEvent เป็นเหตุการณ์ TE_DIGITEVENT เกิดขึ้นแล้ว ให้นำออบเจกต์ pEvent ส่งค่าไปให้กับออบเจกต์ ITDigitDetectionEvent เพื่อรับเหตุการณ์ของสัญญาณกดปุ่มที่เกิดขึ้นเอาไว้

2. ออบเจกต์ ITDigitDetectionEvent ใช้เมธอด digit จะได้ค่าของสัญญาณกดปุ่มที่รับเข้ามา ซึ่งจะเป็นค่าแอสกี (ASCII) จึงได้นำมาแปลงให้เป็นหมายเลขของปุ่มกด แสดงดังตารางที่ 3.1 แล้วส่งไปให้ตัวแปร digit

โดยทำการแปลงค่าดังนี้ $digit = objDigitDetectionEvent.digit - 48$

ตารางที่ 3.1 แสดงค่าของสัญญาณกดปุ่มที่รับมาได้และค่าที่แปลงเป็นหมายเลขของปุ่มกด

หมายเลขปุ่มกด	0	1	2	3	4	5	6	7	8	9	*	#
ค่าจากเมธอด Digit	48	49	50	51	52	53	54	55	56	57	42	35
ค่า Digit หลังแปลง	0	1	2	3	4	5	6	7	8	9	-6	-13

3.2.8 การเล่นเสียง

ในการควบคุมการเล่นเสียงจากแฟ้มข้อมูลเสียงนั้น จะใช้ซอฟต์แวร์คอมพิวเตอร์ชื่อ Mci32.ocx หรือที่เรียกว่า MMControl (Microsoft Multimedia Control) ซึ่งมีคำสั่งการทำงานแสดงดังตารางที่ 3.2

ตารางที่ 3.2 คำสั่งการทำงานของ MMControl

คำสั่งของ MMControl	รายละเอียดการทำงาน
MMControl.FileName = FileName	กำหนดชื่อแฟ้มข้อมูลเสียง
MMControl.Command = "Open"	เปิดแฟ้มข้อมูลเสียง
MMControl.Command = "Prev"	ไปที่ตำแหน่งเริ่มต้นของแฟ้มข้อมูลเสียง
MMControl.Command = "Play"	เล่นแฟ้มข้อมูลเสียง
MMControl.Command = "Stop"	หยุดเล่นแฟ้มข้อมูลเสียง
MMControl.Command = "Close"	ปิดแฟ้มข้อมูลเสียง เมื่อเลิกใช้งาน

ข้อสำคัญของการเล่นเสียงผ่านโมเด็มเสียงออกทางคู่สายโทรศัพท์

1. แฟ้มข้อมูลเสียงต้องเป็นรูปแบบเวฟ (Wave File) ที่มีนามสกุล .wav เท่านั้น สำหรับแฟ้มข้อมูลเสียงแบบอื่นๆ เช่น มิดิ (MIDI File) ที่มีนามสกุล .mid ไม่สามารถเล่นเสียงผ่านโมเด็มเสียงได้
2. ปรับตั้งระบบการเล่นเสียงของเครื่องคอมพิวเตอร์ (Audio Sound Playback) ให้ออกที่โมเด็ม ดังแสดงรายละเอียดไว้ในภาคผนวก ข

3.2.9 การตัดการโทรติดต่อ

การตัดการโทรติดต่อสามารถกระทำได้ 2 กรณี ดังนี้

1. ระบบทำการตัดการโทรติดต่อ ซึ่งทำได้โดยให้ออบเจกต์ ITBasicCallControl ใช้เมธอด Disconnect โดยตั้งค่าพารามิเตอร์ DISCONNECT_CODE เป็น DC_NORMAL หมายถึงแบบปกติ

2. ผู้โทรเข้ามาตัดการโทรติดต่อ ด้วยการวางหูโทรศัพท์

เมื่อผู้โทรเข้ามาวางหูโทรศัพท์ ระบบจะมีเหตุการณ์ TE_CALLSTATE เกิดขึ้นเป็น CS_DISCONNECTED เข้ามาแล้ว จากนั้นจะให้ออบเจกต์ TAPI เรียกใช้เมธอด UnregisterNotifications ด้วยค่าตัวแปร glRegister ที่ได้มาจากขั้นตอนที่ 3.2.3 (Register Events) เพื่อยกเลิกการรับฟังเหตุการณ์ของสายนี้ แล้วให้เรียกใช้เมธอด RegisterCallNotifications ใหม่อีกครั้ง เพื่อเตรียมรับฟังเหตุการณ์ของสายโทรเข้าใหม่ต่อไป

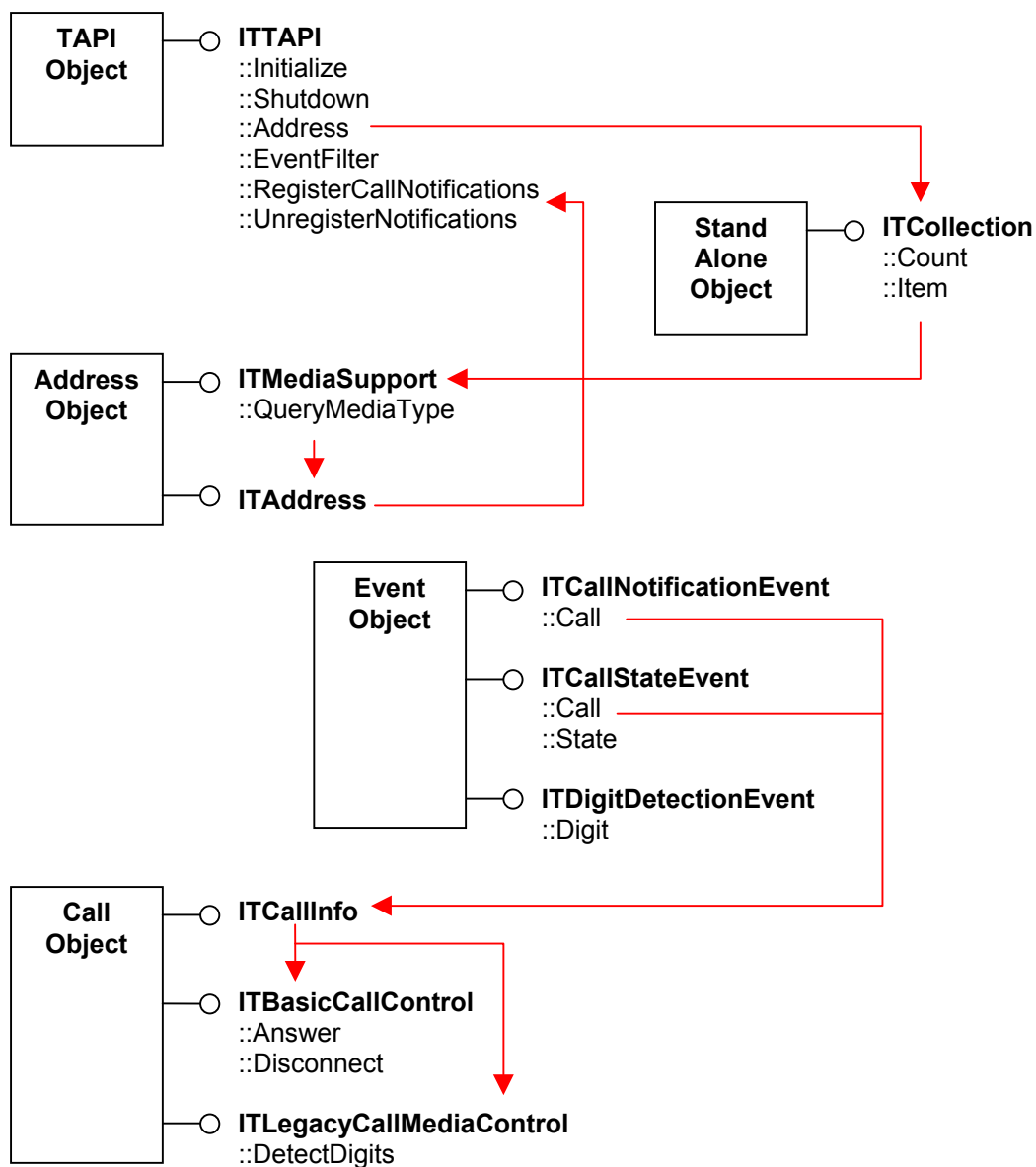
3.2.10 สิ้นสุดการทำงาน

เมื่อต้องการจบการทำงานของระบบ ให้ออบเจกต์ TAPI เรียกใช้เมธอดชื่อ Shutdown เพื่อคืนทรัพยากรที่ใช้กลับไปให้กับระบบคอมพิวเตอร์

3.3 การพัฒนาเป็นคลาสมอดูล

จากขั้นตอนการเขียนโปรแกรมการทำงานติดต่อกันระหว่างคอมพิวเตอร์กับระบบโทรศัพท์ ดังที่ได้กล่าวมาแล้วนั้น จะเห็นว่ามีความซับซ้อนซึ่งต้องใช้เวลาศึกษาค้นคว้าอย่างมาก โดยสามารถแสดงความสัมพันธ์ของออบเจกต์ อินเตอร์เฟสและเมธอดต่างๆ ดังในภาพประกอบที่ 3.2 และเพื่อให้มีความสะดวกรวดเร็วขึ้นในการนำมาใช้พัฒนาระบบสอบถามข้อมูลทางโทรศัพท์โดยทั่วไปได้ ในครั้งต่อไป จึงได้จัดทำวิธีการเหล่านี้ให้เป็นคลาสมอดูลชื่อ CCIS (Call Center for Interactive

Information Services) ซึ่งทำให้ผู้จะพัฒนาระบบสามารถนำมาใช้งานได้โดยไม่ต้องคำนึงถึงวิธีการเขียนโปรแกรมอินเตอร์เฟสกับ TAPI 3.0 อีก ตามรหัสโปรแกรมที่ปรากฏในภาคผนวก ค



ภาพประกอบที่ 3.2 ความสัมพันธ์ของออบเจกต์ อินเตอร์เฟส และเมธอด ของ TAPI 3.0

คลาสมอดูล CCIS

หน้าที่การทำงานหลัก คือ ทำการติดต่อกับโมเด็มเสียง รับสายโทรเข้าได้อัตโนมัติ และมีการตรวจจับสัญญาณกดปุ่มบนแป้นโทรศัพท์ที่ได้ ซึ่งมีรายละเอียดของคลาสดังนี้

- **Properties** เป็นคุณสมบัติของคลาส ที่สามารถให้แสดงหรือปรับตั้งสถานะได้ ได้แก่
 - **AutoAnswering** ให้ปรับตั้งการรับสายโทรเข้าว่าเป็นแบบอัตโนมัติหรือไม่ และแสดงค่าที่ตั้งไว้
 - **AddressName** ให้แสดงชื่อผู้ผลิตและรุ่นของอุปกรณ์โมเด็มที่ต่อกับเครื่องคอมพิวเตอร์
 - **ServiceProviderName** ให้แสดงชื่อโดเมนเนมของโมเด็ม
- **Methods** เป็นคำสั่งการทำงานของคลาส ได้แก่
 - **AnswerCall** คำสั่งให้รับสายโทรเข้า
 - **DisconnectCall** คำสั่งให้ตัดการโทรติดต่อ
 - **RegisterCallNotifications** คำสั่งให้รับฟังเหตุการณ์ทางโทรศัพท์
 - **UnregisterNotifications** คำสั่งให้ยกเลิกการรับฟังเหตุการณ์ทางโทรศัพท์
- **Events** เป็นเหตุการณ์ทางโทรศัพท์ที่คลาสตอบสนอง ได้แก่
 - **AnswerCall** เหตุการณ์ที่ระบบรับสายโทรเข้า
 - **DTMFEvent** เหตุการณ์ที่มีสัญญาณกดปุ่มบนแป้นโทรศัพท์เข้ามา
 - **OfferingCall** เหตุการณ์ที่มีสัญญาณกระดิ่งดัง
 - **DisconnectedCall** เหตุการณ์ตัดการโทรติดต่อ

รายละเอียดของคลาส CCIS และขั้นตอนการติดต่อกับระบบโทรศัพท์ดังที่ได้กล่าวมาทั้งหมด สามารถแสดงด้วย UML (Unified Modeling Language) ซึ่งเป็นภาษาสัญลักษณ์รูปภาพมาตรฐานสำหรับใช้ในการสร้างโมเดลเชิงวัตถุ [ชาติ วรกุลพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2544] ดังต่อไปนี้

แผนภาพคลาส (Class Diagram) ประกอบไปด้วย 4 ส่วน ได้แก่ ชื่อคลาส, ส่วนลักษณะประจำ (Attribute) เป็นส่วนที่แสดงถึงตัวแปรต่างๆ ส่วนดำเนินการ (Operation) เป็นส่วนที่แสดง Properties และ Methods และส่วนการตอบสนอง (Responsibility) เป็นส่วนที่แสดง Events ตามลำดับ ดังภาพประกอบที่ 3.3

สำหรับรายละเอียดของอินเตอร์เฟซต่างๆ ของ TAPI 3.0 ที่มีการเรียกใช้ภายในคลาส CCIS แสดงดังภาพประกอบที่ 3.4

แผนภาพลำดับ (Sequence Diagram) จะบอกขั้นตอนการทำงานระหว่าง
 ออบเจ็กต์ต่างๆ ของแต่ละอินเตอร์เฟซ การติดต่อส่งข่าวสารกัน และลำดับเวลาของการเกิดเหตุ
 การณ์ต่างๆ โดยแบ่งออกเป็นสองส่วน ดังภาพประกอบที่ 3.5 แสดงแผนภาพลำดับซึ่งมีผู้ดูแล
 ระบบเป็นผู้กระทำ และภาพประกอบที่ 3.6 แสดงแผนภาพลำดับซึ่งมีผู้ใช้บริการเป็นผู้กระทำ

CCIS	
<ul style="list-style-type: none"> - gobjTapi : ITTAPI - gobjTapiWithEvents : ITTAPI - gobjAddress : ITAddress - gobjReceivedCallInfo : ITCallInfo - gobjMediaSupport : ITMediaSupport - gobjBasicCallControlin : ITBasicCallControl - gobjLegacyCall : ITLegacyCallMediaControl - glRegistrationToken : Long - glMediaType : Long - AutoAnswerFlag : Boolean 	
<ul style="list-style-type: none"> + Get AddressName() : String + Get ServiceProviderName() : String + Get AutoAnswer() : Boolean + Let AutoAnswer(Flag : Boolean) + AnswerCall() + DisconnectCall() + RegisterCallNotifications() + UnregisterNotifications() - SelectAnAddress() - Class_Initialize() - Class_Terminate() - gobjTapiWithEvents_Event(TapiEvent : TAPI3Lib.TAPI_EVENT,pEvent : Object) 	
Events	
<ul style="list-style-type: none"> + OfferingCall() + AnswerCall() + DTMFEvent(digit : Integer) + DisconnectedCall() 	

ภาพประกอบที่ 3.3 แผนภาพคลาสของคลาส CCIS

ภาพประกอบที่ 3.4 อินเทอร์เน็ตในแผนภาพคลาสของคลาส CCIS

