บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

ปัญหาที่เกิดจากองค์กรทั้งรัฐบาลและเอกชนที่เกี่ยวกับการจัดการเอกสารในปัจจุบัน ส่วนใหญ่เป็นปัญหาเนื่องจากข้อความหรือการสื่อสารไม่ถึงผู้รับหรือถึงช้ากว่ากำหนดเวลาที่ต้องการ ส่วนมากจะใช้วิธีการส่งเอกสารโดยใช้ผู้นำส่ง ซอฟต์แวร์ที่ใช้จัดการสำหรับงานประเภทนี้ที่เป็นที่นิยมใช้มีหลายโปรแกรม เช่น โปรแกรมโลตัสโนท (Lotus Notes) เป็นต้น แต่เนื่องจากโปรแกรมเหล่านี้ค่อนข้างมีราคาแพงและไม่เคยมีการพัฒนาสำหรับคนไทย ทำให้การใช้งานส่วนใหญ่ต้องพึ่งพาโปรแกรมจากต่างประเทศ ทำให้ไม่สนองตอบต่อความต้องการของสังคมไทย ผู้วิจัยจึงมีแนวคิดที่จะพัฒนาโปรแกรมจัดการงานเอกสารขึ้นโดยอาศัยแนวคิดและทฤษฎีที่เกี่ยวข้องดังนี้

1. ระเบียบสำนักนายกรัฐมนตรี ว่าด้วยงานสารบรรณ พ.ศ.2536 ปรับปรุงใหม่
2. ระเบียบว่าด้วยการรักษาความปลอดภัยแห่งชาติ พ.ศ.2517
3. ทฤษฎีกระแสงาน (Workflow)
4. ทฤษฎีเทคโนโลยีเชิงวัตถุ (Object Oriented Technology)
5. ระบบผู้ขอใช้บริการ / ผู้ให้บริการ
6. ทฤษฎีเกี่ยวกับฐานข้อมูล

2.1. ระเบียบสำนักนายกรัฐมนตรีว่าด้วยงานสารบรรณ พ.ศ. 2526

2.1.1. หนังสือและชนิดของหนังสือราชการ

2.1.1.1. ความหมาย[1]

หนังสือราชการ คือ เอกสารที่เป็นหลักฐานในราชการ ได้แก่

2.1.1.1.1. หนังสือที่มีไปมาระหว่างส่วนราชการ

2.1.1.1.2. หนังสือที่ส่วนราชการมีไปถึงหน่วยงานอื่นซึ่งมิใช่ส่วนราชการ หรือที่มีไปถึงบุคคลภายนอก

2.1.1.1.3. หนังสือที่หน่วยงานอื่นใดซึ่งมิใช่ส่วนราชการ หรือบุคคลภายนอกมีมาถึงส่วนราชการ

2.1.1.1.4. เอกสารที่ทางราชการจัดทำขึ้นเพื่อเป็นหลักฐานในราชการ

2.1.1.1.5. เอกสารที่ทางราชการจัดทำขึ้นตามกฎหมาย ระเบียบ หรือข้อบังคับ

2.1.1.2. ชนิดของเอกสาร

2.1.1.2.1. หนังสือภายนอก

คือหนังสือติดต่อส่วนราชการที่เป็นแบบพิธี โดยใช้กระดาษตราครุฑ เป็นหนังสือติดต่อระหว่างส่วนราชการ หรือส่วนราชการมีถึงหน่วยงานอื่นใด ซึ่งมิใช่ส่วนราชการ หรือที่มีถึงบุคคลภายนอก

2.1.1.2.2. หนังสือภายใน

คือหนังสือติดต่อส่วนราชการที่เป็นแบบพิธี น้อยกว่าหนังสือภายนอก เป็นหนังสือติดต่อภายในกระทรวงทบวงกรมหรือจังหวัดเดียวกัน

2.1.1.2.3. หนังสือประทับตรา

คือหนังสือที่ใช้ตราประทับแทนการลงชื่อ ของหัวหน้าส่วนราชการระดับกรมขึ้นไป โดยให้หัวหน้าส่วนราชการระดับกอง หรือผู้ที่ได้รับมอบหมายจากหัวหน้าส่วนราชการระดับกรมขึ้นไป เป็นผู้รับผิดชอบลงชื่อย่อกำกับตรา หนังสือประทับตราให้ใช้ได้ทั้งระหว่างส่วนราชการกับส่วนราชการและระหว่างส่วนราชการกับบุคคลภายนอก เฉพาะกรณีที่ไม่ใช่เรื่องสำคัญ

2.1.1.3. หนังสือสั่งการ มี 3 ชนิด

2.1.1.3.1. คำสั่ง

คือบรรดาข้อความที่ผู้บังคับบัญชาสั่งการให้ปฏิบัติโดยชอบด้วยกฎหมาย

2.1.1.3.2. ระเบียบ

คือบรรดาข้อความที่ผู้มีอำนาจหน้าที่ได้วางไว้ โดยจะอาศัยอำนาจของกฎหมายหรือไม่ก็ได้ เพื่อถือเป็นหลักปฏิบัติงานเป็นการประจำ

2.1.1.3.3. ข้อบังคับ

คือบรรดาข้อความที่ผู้มีอำนาจหน้าที่ กำหนดให้ใช้โดยอาศัยอำนาจของกฎหมายที่บัญญัติให้กระทำได้

2.1.1.4. หนังสือประชาสัมพันธ์ มี 3 ชนิด

2.1.1.4.1. ประกาศ

คือบรรดาข้อความที่ทางราชการประกาศหรือชี้แจงให้ทราบ หรือแนะแนวทางปฏิบัติ

2.1.1.4.2. แถลงการณ์

คือบรรดาข้อความที่ทางราชการแถลง เพื่อทำความเข้าใจในกิจการของทางราชการหรือเหตุการณ์หรือกรณีใด ๆ ให้ทราบชัดเจนโดยทั่วกัน

2.1.1.4.3. ข่าว

คือบรรดาข้อความที่ทางราชการเห็นสมควรเผยแพร่ให้ทราบ

2.1.1.5. หนังสือที่เจ้าหน้าที่ทำขึ้นหรือรับไว้เป็นหลักฐานในราชการ

คือหนังสือที่ทางราชการทำขึ้นนอกจากที่กล่าวมาแล้วข้างต้น หรือหนังสือที่หน่วยงานอื่นใด ซึ่งมิใช่ส่วนราชการหรือบุคคลภายนอกมีมาถึงส่วนราชการ และส่วนราชการรับไว้เป็นหลักฐานของทางราชการ มี 4 ชนิด

2.1.1.5.1. หนังสือรับรอง

คือหนังสือที่ส่วนราชการออกให้เพื่อรับรองแก่ บุคคล นิติบุคคล หรือหน่วยงาน เพื่อวัตถุประสงค์อย่างใดอย่างหนึ่ง ให้ปรากฏแก่บุคคลโดยทั่วไปไม่จำเพาะเจาะจง

2.1.1.5.2. รายงานการประชุม

คือการบันทึกความคิดเห็นของผู้มาประชุม ผู้เข้าร่วมประชุม และมติของที่ประชุมไว้เป็นหลักฐาน

2.1.1.5.3. บันทึก

คือข้อความซึ่งผู้ใต้บังคับบัญชาเสนอต่อผู้บังคับบัญชา หรือผู้บังคับบัญชาสั่งการแก่ผู้ใต้บังคับบัญชา หรือข้อความที่เจ้าหน้าที่หรือหน่วยงานระดับต่ำกว่าส่วนราชการระดับกรมติดต่อกันในการปฏิบัติราชการ

2.1.1.5.4. หนังสืออื่น

คือหนังสือหรือเอกสารอื่นใด ที่เกิดขึ้นเนื่องจากการปฏิบัติงานของเจ้าหน้าที่เพื่อเป็นหลักฐานในทางราชการ

2.1.1.6. บทเบ็ดเตล็ด

2.1.1.6.1. หนังสือที่ต้องปฏิบัติให้เร็วกว่าปกติ

เป็นหนังสือที่ต้องจัดส่ง และดำเนินการทางสารบรรณด้วยความรวดเร็วเป็นพิเศษ แบ่งเป็น 3 ประเภท คือ

2.1.1.6.1.1. ด่วนที่สุด ให้เจ้าหน้าที่ปฏิบัติในทันทีที่ได้รับหนังสือนั้น

2.1.1.6.1.2. ด่วนมาก ให้เจ้าหน้าที่ปฏิบัติโดยเร็ว

2.1.1.6.1.3. ด่วน ให้เจ้าหน้าที่ปฏิบัติเร็วกว่าปกติ เท่าที่จะทำได้

ในกรณีที่ต้องการให้หนังสือส่งถึงผู้รับภายในเวลาที่กำหนด ให้ระบุคำว่า ด่วนภายใน แล้ว ลงวัน เดือน ปี และเวลาที่ต้องการให้หนังสือนั้นไปถึงผู้รับ

2.1.1.6.2. หนังสือเวียน คือหนังสือที่มีถึงผู้รับเป็นจำนวนมาก มีใจความอย่างเดียวกัน ให้เพิ่มรหัสพยัญชนะ ว หน้าเลขทะเบียนหนังสือส่ง ซึ่งกำหนดเป็นเลขที่หนังสือเวียนโดยเฉพาะ

2.2. ระเบียบว่าด้วยการรักษาความปลอดภัยแห่งชาติ พ.ศ.2517

2.2.1. การรักษาความปลอดภัยเกี่ยวกับเอกสาร

2.2.1.1. ความหมาย[2]

การรักษาความปลอดภัยเกี่ยวกับเอกสาร คือ มาตรการที่กำหนดสำหรับการปฏิบัติต่อเอกสารลับ เพื่อป้องกันมิให้ผู้ไม่มีอำนาจได้ล่วงรู้หรือเข้าถึงเอกสาร

2.2.1.2. ความมุ่งหมาย

การรักษาความปลอดภัยเกี่ยวกับเอกสารมีความมุ่งหมาย เพื่อกำหนดนโยบายและวิธีการในการพิทักษ์รักษาความลับของทางราชการ ซึ่งจะต้องปฏิบัติต่อเอกสารลับชั้นต่าง ๆ ทุกขั้นตอน

2.2.2. ชั้นความลับของทางราชการแบ่งออกเป็น 4 ระดับ

2.2.2.1. ลับที่สุด ในภาษาอังกฤษให้ใช้คำว่า TOP SECRET

2.2.2.2. ลับมาก ในภาษาอังกฤษให้ใช้คำว่า SECRET

2.2.2.3. ลับ ในภาษาอังกฤษให้ใช้คำว่า CONFIDENTIAL

2.2.2.4. ปกปิด ในภาษาอังกฤษให้ใช้คำว่า RESTRICTED

2.2.3. การรักษาความปลอดภัยเกี่ยวกับบุคคล

2.2.3.1. ความหมาย[2]

การรักษาความปลอดภัยเกี่ยวกับบุคคล คือ มาตรการที่กำหนดขึ้นสำหรับใช้ปฏิบัติต่อข้าราชการหรือผู้ที่จะได้รับความไว้วางใจให้เข้าถึงสิ่งที่เป็นความลับของทางราชการหรือให้ปฏิบัติหน้าที่ราชการที่สำคัญ เพื่อให้เป็นเชื่อแน่ว่าต้องเป็นผู้ที่ไม่เป็นภัยต่อความมั่นคงของประเทศชาติ

2.2.3.2. ความมุ่งหมาย

การรักษาความปลอดภัยเกี่ยวกับบุคคลกำหนดขึ้นด้วยความมุ่งหมาย เพื่อเลือกเฟ้นตรวจสอบให้ได้บุคคลที่มีลักษณะเหมาะสมแก่การบรรจุเข้ารับราชการ หรือให้ปฏิบัติหน้าที่และเพื่อกำหนดระดับความไว้วางใจในการมอบหมายให้ปฏิบัติหน้าที่เกี่ยวกับความลับ

2.3. ทฤษฎีกระแสงาน (Workflow)

2.3.1. ความหมาย[3]

คือการดำเนินการกระบวนการธุรกิจแบบอัตโนมัติทั้งหมดหรือบางส่วน หรือการใช้คอมพิวเตอร์อำนวยความสะดวกในการดำเนินการ กระบวนการธุรกิจทั้งหมดหรือบางส่วนกระบวนการประกอบธุรกิจในปัจจุบันมีความซับซ้อนมากขึ้น ซึ่งอาจจะมีหลายสาเหตุ เช่น เป็นสถานประกอบการขนาดใหญ่ มีพนักงานเป็นจำนวนมาก มีสาขาต่าง ๆ ทั้งภายในและต่างประเทศ เป็นต้น ทำให้การกระทำกิจกรรมต่าง ๆ ที่เกิดขึ้นอาจจะมีความยุ่งยากซับซ้อน สูญเสียค่าใช้จ่ายเป็นจำนวนมาก สิ้นเปลืองเวลา ทำให้ประสิทธิภาพในกระบวนการทำงานลดน้อยลง ทำให้มีผู้ค้นคิดกรรมวิธีที่จะให้มีการเพิ่มประสิทธิภาพให้ดียิ่งขึ้นเช่นการรื้อปรับระบบ (Reengineering) ซึ่งเป็นที่ยอมรับว่าทำให้ประสิทธิภาพในการทำงานขององค์กรดีขึ้น

การรื้อปรับระบบ หมายถึง การปรับเปลี่ยนขั้นตอนการทำงาน หรือวิธีการทำงานขององค์กร โดยมีวัตถุประสงค์เพื่อให้องค์กรทำงานได้อย่างมีประสิทธิภาพสูงสุดและมีค่าใช้จ่ายน้อยที่สุด ทั้งรวมถึง การปรับเปลี่ยนโครงสร้างองค์กร วัฒนธรรม รวมทั้งทัศนคติในการทำงาน ตามกฎระเบียบต่าง ๆ จะเห็นได้ว่าการรื้อปรับระบบที่สมบูรณ์แล้วจะต้องทำในทุก ๆ ด้าน ในการรื้อปรับโครงสร้าง ได้มีการนำระบบกระแสงาน (Workflow System) เข้ามาใช้เป็นส่วนหนึ่งในกระบวนการ

กระแสงานเป็นแนวทางในการแก้ปัญหากระบวนการธุรกิจ (Business Process) สมัยเก่า นำมาช่วยในการบริหารจัดการและสนับสนุนกระบวนการธุรกิจสมัยใหม่ เป็นส่วนที่นำไปใช้ควบคุมพลังเทคโนโลยีสมัยใหม่ให้สนับสนุนโครงสร้างของงาน เพื่อให้งานในกระบวนการธุรกิจนั้นเป็นไปตามวัตถุประสงค์ กระแสงานจะเป็นส่วนสนับสนุนโครงสร้างหลักของกระบวนธุรกิจ เป็นตัวกลางที่ทำหน้าที่ตอบสนองระหว่างคนกับคนหรือระหว่างงานกับงานอย่างเหมาะสม ซึ่งระบบบริหารกระแสงาน (Workflow Management System) เป็นตัวอย่างหนึ่งที่สนับสนุนให้ขั้นตอนในกระบวนการธุรกิจเป็น

ไปโดยอัตโนมัติโดยอาศัยการจัดการลำดับของกิจกรรมของงานแล้วนำไปมอบหมายให้คนและหรือแหล่งทรัพยากรทางเทคโนโลยีได้อย่างเหมาะสม

แรงจูงใจสำหรับการใช้งานระบบกระแสงาน มีหลาย ๆ หน่วยงานทั้งภาคธุรกิจและภาคราชการ นิยมนำระบบกระแสงานเข้ามาใช้งานเนื่องจากมีปัจจัยหลัก ๆ อยู่ 3 ประการ

- เพื่อปรับปรุงประสิทธิภาพระบบงาน และนำไปสู่การทำให้หน่วยงานมีค่าใช้จ่ายน้อยลงแต่มีภารกิจต่อคนมากขึ้น
- เพื่อปรับปรุงการควบคุมระบบงาน ผลเนื่องมาจากขั้นตอนที่ได้มาตรฐานของระบบ
- เพื่อปรับปรุงความสามารถในการจัดการกระบวนการ ทำให้เข้าใจปัญหาและแยกปัญหาได้ชัดเจนขึ้น

2.3.2. แนวคิดเบื้องต้นของระบบกระแสงาน[3]

มีแนวคิดพื้นฐานอยู่ 4 ส่วนคือ

2.3.2.1. กระบวนการเชิงตรรก (Process Logic)

ทำหน้าที่ในการจัดการกับภารกิจโดยทำหน้าที่ตรวจสอบ รายงาน และมอบหมายกระบวนการให้ภารกิจได้อย่างเหมาะสม

2.3.2.2. มอบหมายภารกิจเหมาะกับคน (Match-Making between People and Tasks)

ในการทำงานดังกล่าว จะเกิดขึ้นเมื่อมีความต้องการที่จะทำงานให้สำเร็จชิ้นหนึ่ง ระบบกระแสงานจะทำหน้าที่จัดการมอบหมายงานให้กับคนโดยอาศัยข้อมูลจากแหล่งข้อมูล (Resource) แล้วปฏิบัติตามกฎ (Role) ที่ได้ออกแบบไว้ในการติดตั้งสิ่งแวดล้อมให้ระบบกระแสงาน (Workflow Environment) แต่อย่างไรก็ตามก็มีความแตกต่างกันไปขึ้นอยู่กับการออกแบบผลิตภัณฑ์ระบบกระแสงานนั้นด้วย

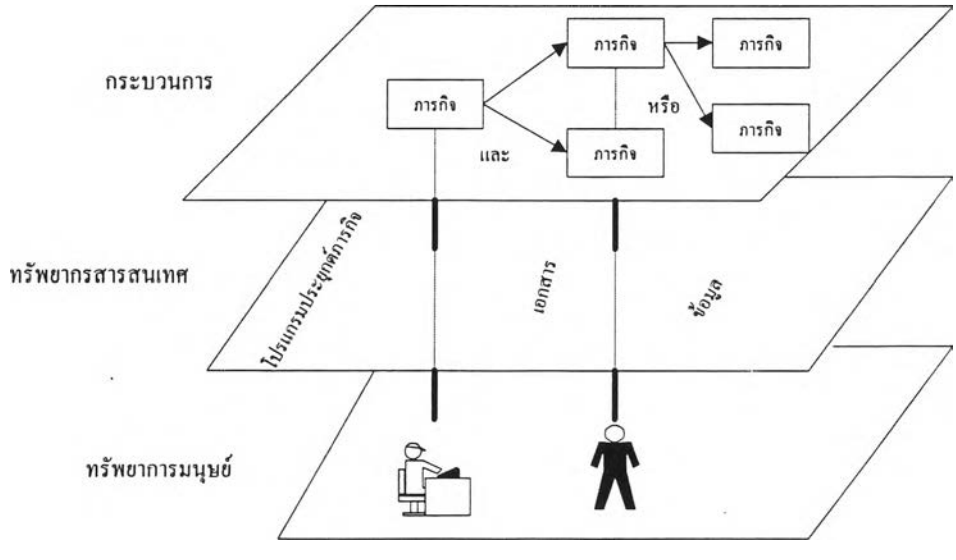
2.3.2.3. ให้แหล่งข้อมูลข่าวสารแก่ภารกิจ (Providing Information Resource for Task) ภารกิจบางอย่างอาจจะต้องการข้อมูลทั้งจากทรัพยากรมนุษย์ (Human Resources) และแหล่งข้อมูลข่าวสาร (Information Resources) เมื่อแหล่งข่าวสารข้อมูลถูกจัดเก็บอยู่ภายในระบบคอมพิวเตอร์ทำให้แน่ใจได้ว่าการจัดการ

ภารกิจให้เหมาะสมกับคนนั้นเป็นไปอย่างถูกต้อง ข้อมูลที่ถูกจัดเก็บอาจจะอยู่ในสื่อต่าง ๆ ของคอมพิวเตอร์เราพอจะแบ่งออกเป็น 2 ประเภท ดังนี้

- ทรัพยากรโปรแกรมประยุกต์ด้านภารกิจ (Task Application Resources) ใช้ในภารกิจระบบกระแสงาน (Workflow Tasks) เช่น โปรแกรมพิมพ์เอกสาร (Word Processor) ซึ่งเป็นโปรแกรมประยุกต์เพื่อเพิ่มผลผลิตส่วนบุคคล (Personal Productivity Application) หรือ โปรแกรมออกใบสั่งซื้อสินค้า ซึ่งเป็นโปรแกรมประยุกต์สำหรับส่งเสริมการทำธุรกิจ (Line-Of-Business Application) เป็นต้น
- ทรัพยากรสาระ (Content Resource) ซึ่งบรรจุข้อมูลข่าวสารต่าง ๆ ได้แก่ ข้อมูลทางด้าน เอกสาร (Document) ข้อมูล (Data)

2.3.2.4. บริหารกระบวนการ (Process Management)

ซึ่งถือว่าเป็นแนวคิดหลักของระบบกระแสงาน ระบบกระแสงานจะช่วยได้โดยอนุญาตให้ผู้ออกแบบสามารถสร้าง เก็บ และประเมินผลโดยวัดความสัมพันธ์ของเวลา ค่าใช้จ่าย ระดับของประสิทธิภาพของกระบวนการและส่วนประกอบของภารกิจ เพื่อนำมาทำการปรับปรุงโดยการออกแบบกระบวนการ (Process Design) และการจัดสรรทรัพยากรให้แก่กระบวนการ (Process Resourcing) ให้ดียิ่งขึ้น

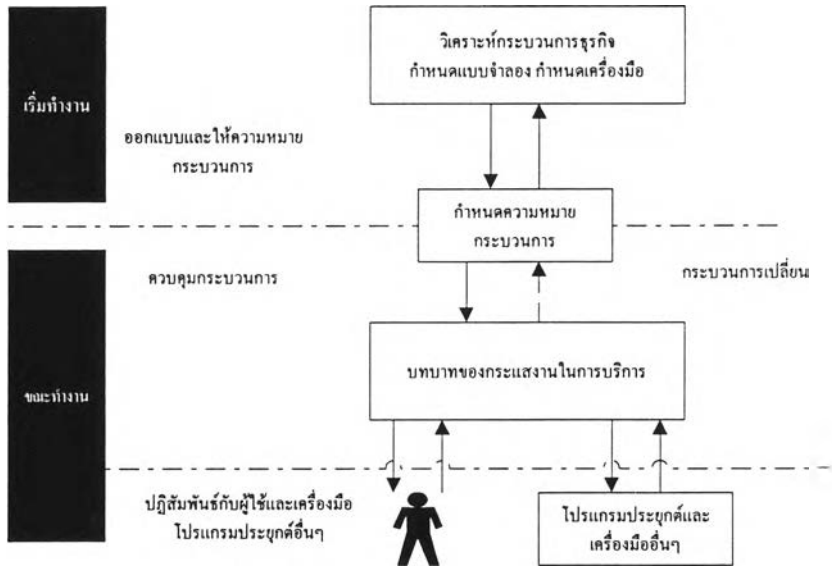


รูปที่ 2.1 แผนภาพอธิบายกระแสงานควบคุมกระบวนการตรรกและมอบหมายให้แก่ทรัพยากรภารกิจ (Workflow Enforces Process Logic and Delivers Task Resoureces)[3]

จากรูปที่ 2.1 อธิบายได้ว่าระบบกระแสงานได้สนับสนุนภารกิจในกระบวนการธุรกิจให้เหมาะสมกับผู้ปฏิบัติทำงานภารกิจเหล่านั้นสำเร็จไปด้วยทันตามกำหนดอย่างมีประสิทธิภาพ

2.3.3. แบบจำลองทั่วไปของกระแสงาน (The Generic Reference Model)[3]

แบบจำลองทั่วไป จะบรรยายถึงรูปแบบจำลองพื้นฐานของโครงสร้างระบบกระแสงานและส่วนที่เกี่ยวข้อง ว่ามีอะไรบ้างซึ่งแบบจำลองระดับบนสุดจะช่วยสนับสนุนหน้าที่หลักของระบบกระแสงาน 3 อย่างดังนี้ สามารถอธิบายได้ดังรูปที่ 2.2



รูปที่ 2.2 แสดงลักษณะของระบบกระแสงาน (Workflow System characteristic)[3]

2.3.3.1. กำหนดรูปแบบ (Build-Time function) ทำหน้าที่เกี่ยวกับการกำหนดกระบวนการของกระแสระบบงานพร้อมทั้งกิจกรรมจากรูปแบบที่มีอยู่จริง (Real World) ให้อยู่ในรูปแบบที่สามารถใช้งานกับคอมพิวเตอร์ได้บางครั้งจะเรียกว่าแบบจำลองกระบวนการ (Process Model) หรือ แม่แบบกระบวนการ (Process Template) หรือ กระบวนการเมทาเดทา (Process Metadata)

2.3.3.2. ควบคุมการทำงานของกระบวนการ (Run-Time control function) ทำหน้าที่บริหารจัดการกระบวนการของกระแสระบบงานในสิ่งแวดล้อมขณะปฏิบัติงาน พร้อมทั้งจัดลำดับกิจกรรมเพื่อให้เหมาะสมกับกระบวนการนั้นๆ

2.3.3.3. ติดต่อกับผู้ใช้และโปรแกรมประยุกต์อื่นๆ (Run-Time activity interaction function) ทำหน้าที่จัดการกิจกรรมให้เป็นไปตามลำดับขั้นขณะที่ทำการติดต่อกับผู้ใช้หรือเครื่องมือในโปรแกรมประยุกต์อื่นๆ

2.3.4. แบบจำลองการนำไปใช้งานของผลิตภัณฑ์ (Product Implementation Model)

ถึงแม้ว่าผลิตภัณฑ์ที่ใช้งานเกี่ยวกับระบบจัดการระบบกระแสงานจะมีออกมามากมายแต่เมื่อเปรียบเทียบส่วนประกอบพื้นฐานของผลิตภัณฑ์แล้วจะเห็นว่ามีส่วนประกอบพื้นฐานคล้ายคลึงหรือเหมือนกันจะแตกต่างกันเพียงรายละเอียดเท่านั้นรายละเอียดจากรูปที่ 2.3

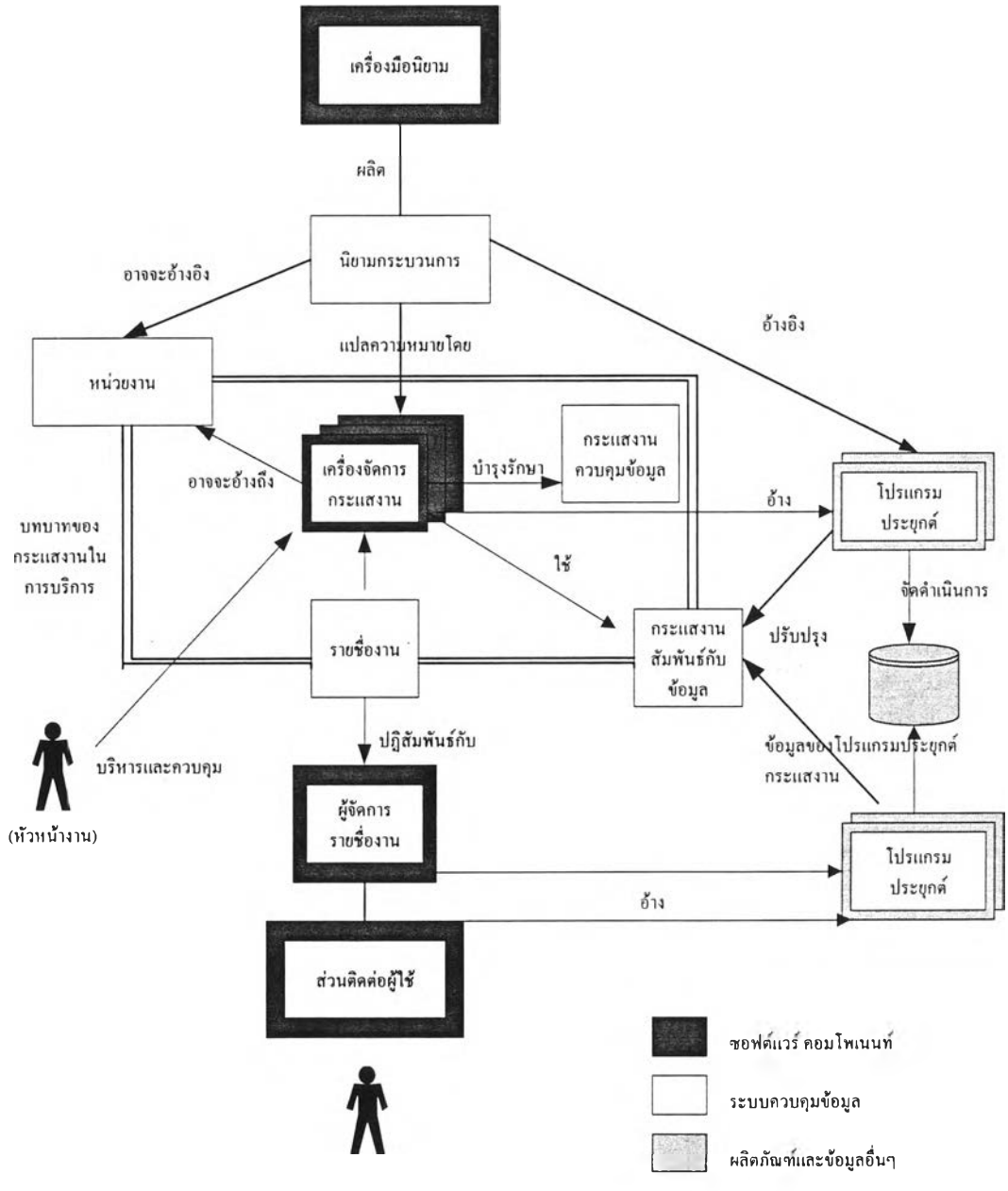
จากรูปสามารถอธิบายได้ว่าแบบจำลองทั่วไปมีส่วนประกอบ 3 ชนิด

- ซอฟต์แวร์คอมโพเนนท์ (Software Component) ทำหน้าที่สนับสนุนหน้าที่ต่างๆ ของระบบกระแสงาน
- สวนกำหนดชนิดของระบบและควบคุมข้อมูลต่าง ๆ (System Definition and Control Data) ซึ่งจะถูกใช้โดยส่วนประกอบซอฟต์แวร์
- โปรแกรมประยุกต์และโปรแกรมประยุกต์ฐานข้อมูล (Application and Application Databases) ไม่ได้เป็นส่วนประกอบของผลิตภัณฑ์ระบบงานแต่อาจจะถูกอ้างเข้าไปเป็นส่วนหนึ่งของระบบกระแสงาน

2.3.5. แผนภาพการเลือกนำไปใช้งาน (Alternative Implementation Scenarios)

จากแบบจำลองโครงสร้างของระบบกระแสงานทั่วไป จะเห็นว่ามีโครงสร้างพื้นฐานของผลิตภัณฑ์เหมือนกัน แต่การที่จะนำไปประยุกต์ใช้งานอาจจะมีปัจจัยหลายอย่างที่ทำให้ได้ผลิตภัณฑ์ออกมาแตกต่างกัน เช่น รูปแบบ (Platform) โครงสร้างพื้นฐานทางด้านเครือข่าย (Network Infrastructure) และการทำหน้าที่ของซอฟต์แวร์

ระบบกระแสงานที่แตกต่างกันทำให้ได้ผลิตภัณฑ์ออกมาแตกต่างกัน แต่โดยทั่วๆ ไปจะแยกเป็น 2 แนวทาง

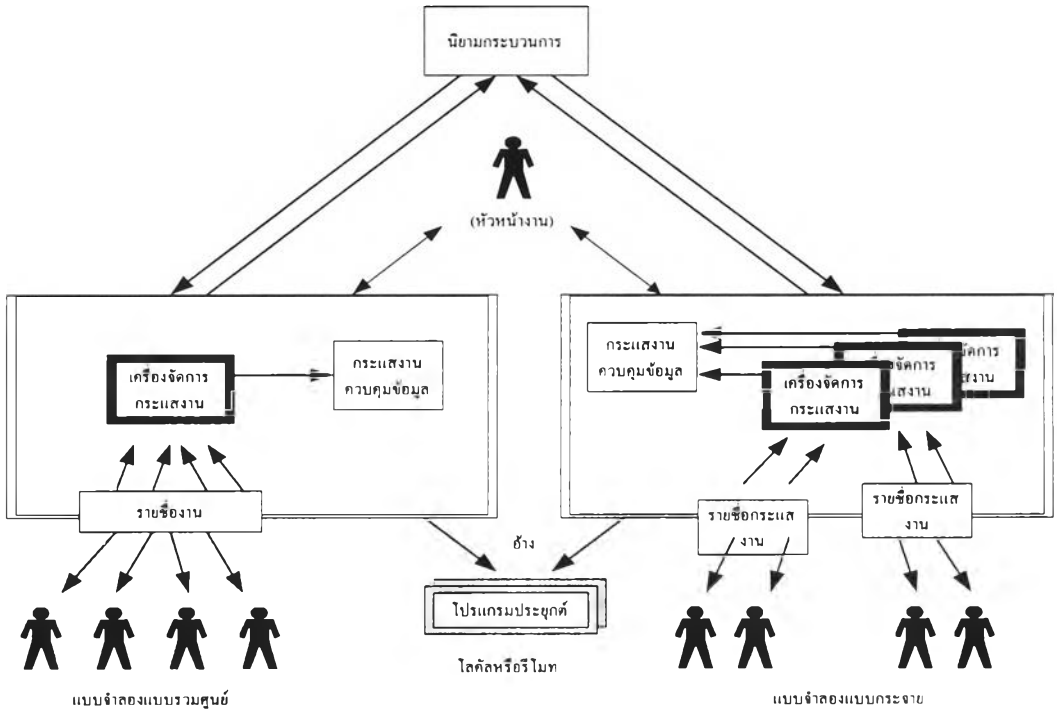


รูปที่ 2.3 แสดงโครงสร้างของผลิตภัณฑ์ระบบกระแสงานทั่วไป [3]

(Generic Workflow Product Structure)

- บริการแบบรวมศูนย์ (Centralize) หรือ แบบบริการกระจายศูนย์ (Distributed)
- แบบเจาะจงตำแหน่งผู้ใช้งาน (Worklist Handler location) กระจาย (Distribute Mechanism) คือไม่เจาะจงตำแหน่งผู้ใช้งาน

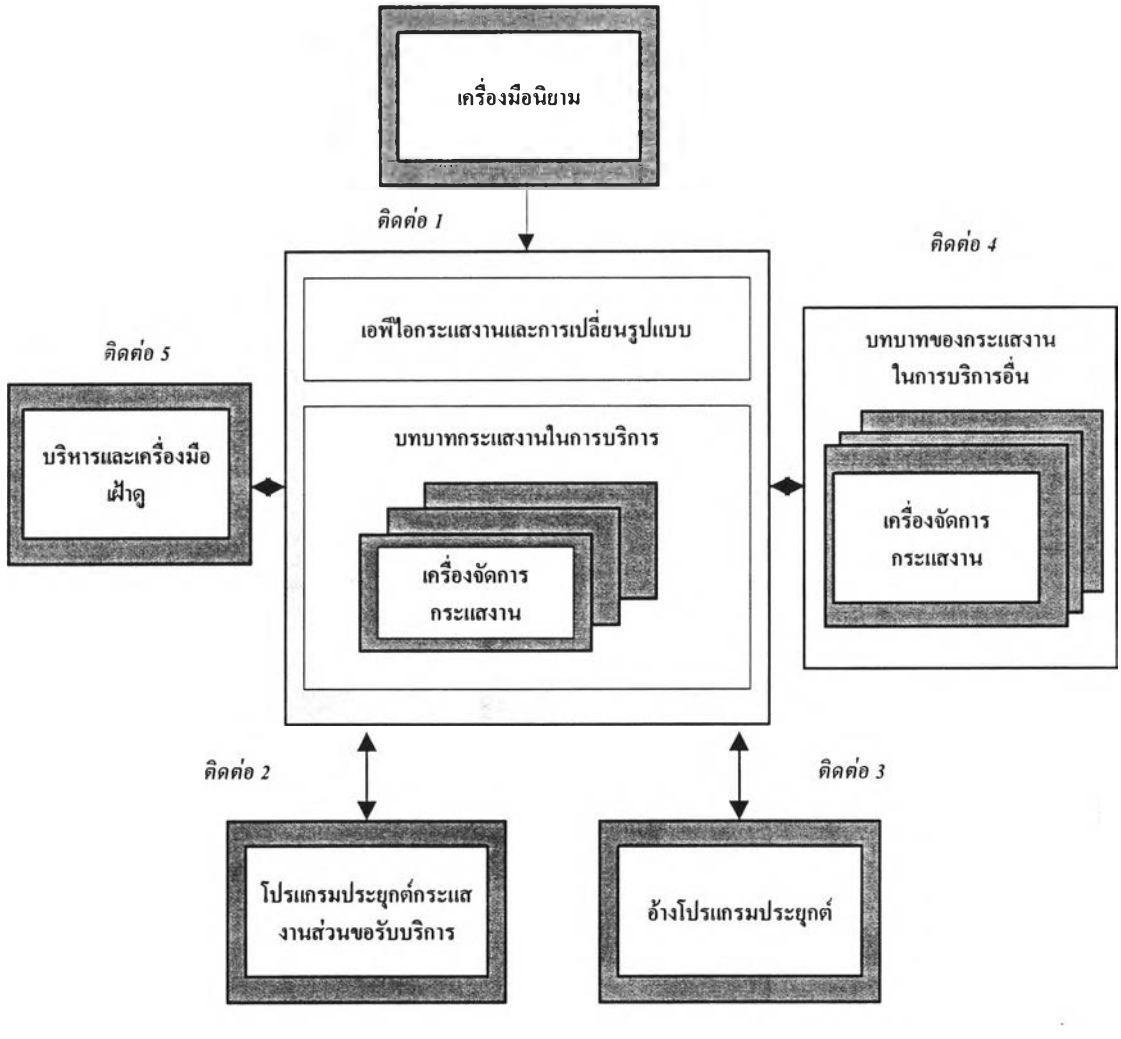
รายละเอียดอธิบายได้ในรูปที่ 2.4



รูปที่ 2.4 แสดงขอบเขตมาตรฐานบทบาทของกระแสงานในการบริการ[3]

2.3.6. แบบจำลองอ้างอิงของระบบกระแสงาน (Workflow Reference Model)

ถูกพัฒนาต่อมาจากแบบจำลองโครงสร้างทั่วไปของระบบกระแสงานประยุกต์ โดยทำการตรวจสอบตัวประสานที่เหมือนกันในโครงสร้างเดียวกันซึ่งสามารถที่จะทำงานในระดับต่าง ๆ ได้ ระบบกระแสงานทั้งหมดจะบรรจุไปด้วยจำนวนของส่วนประกอบทั่วไปอยู่ในเซทที่ถูกกำหนด แต่จำนวนมากน้อยจะแตกต่างกันไปแล้วแต่ชนิดของผลิตภัณฑ์แต่ทุกผลิตภัณฑ์จะมีตัวประสานและรูปแบบการแลกเปลี่ยนข้อมูลที่เป็นมาตรฐาน หรือส่วนประกอบที่จำเป็นเหมือนกัน ซึ่งจะสามารถเรียกได้โดยการอ้างอิง เช่น การประสานชนิดพื้นฐาน เป็นต้น



รูปที่ 2.5 แบบจำลองอ้างอิงของกระแสงาน คอมโพเนนท์และส่วนติดต่อ[3]

จากรูปที่2.5 แสดงถึงส่วนประกอบและตัวประสานหลักภายในสถาปัตยกรรมของระบบกระแสงานที่เหมือนกันโปรแกรมประยุกต์กระแสงาน (Workflow API) และโปรแกรมสำหรับเปลี่ยนแปลงรูปแบบให้เป็นมาตรฐาน (Interchange formats) จะถูกพัฒนาขึ้นมาใช้เป็นแกนสำหรับใช้ร่วมกัน ส่วนความต้องการประสานของแต่ละส่วนจะถูกเพิ่มเข้าตามมาโดยเฉพาะ

2.4. ทฤษฎีเทคโนโลยีเชิงวัตถุ (Object Oriented Technology)

เนื่องจากปัญหาในการพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่และมีความซับซ้อนสูง ๆ คือค่าใช้จ่ายและการที่ต้องใช้เวลามาก ตลอดทั้งการบำรุงรักษาและการแก้ไขไม่สามารถทำได้โดยง่าย ซึ่งจากปัญหาดังที่กล่าวมาแล้วจึงมีผู้คิดค้นวิธีการหรือเทคโนโลยีสำหรับการพัฒนาระบบขึ้นมาใหม่ เรียกการพัฒนาซอฟต์แวร์แบบเทคโนโลยีเชิงวัตถุ (Object Oriented Technology) ซึ่งมีนักพัฒนาระบบเป็นจำนวนมากนำมาใช้เป็นเครื่องมือในการพัฒนา การพัฒนาและออกแบบระบบสามารถทำได้โดยง่ายซอฟต์แวร์ที่พัฒนาขึ้นนำมาบำรุงรักษา แก้ไข และการนำกลับมาใช้ใหม่ (Reusable) ทำได้ง่าย

เทคโนโลยีเชิงวัตถุเป็นแนวคิดของการพัฒนาโปรแกรม ที่ผนึกเอาข้อมูลและโปรแกรมเข้าด้วยกันเป็นวัตถุ และกำหนดตัวประสานระหว่างวัตถุต่าง ๆ โดยถ้าวัตถุหนึ่งต้องการจะใช้ข้อมูลอีกวัตถุหนึ่งจะต้องติดต่อผ่านตัวประสานนี้เท่านั้น จากหลักการนี้ทำให้พบว่าการเปลี่ยนแปลงแก้ไขโปรแกรมใด ๆ หรือวัตถุใด ๆ จะกระทบกระเทือนต่อวัตถุอื่นหรือโปรแกรมอื่นน้อยมาก ซึ่งทำให้การแก้ไข้โปรแกรมจะกระทำได้โดยสะดวก ง่ายสำหรับการบำรุงรักษา นอกจากนี้ยังมีคุณสมบัติในการถ่ายทอด (Inheritance) คุณสมบัติจากวัตถุหนึ่งไปยังอีกวัตถุหนึ่งทำให้ตรงกับแนวคิดในการนำซอฟต์แวร์กลับมาใช้ใหม่

2.4.1. วงจรการพัฒนาระบบแบบเรชันแนลออปเจกซ์ทอรีโพรเซส (Rational Obejectory Process Software Development Life Cycle)

เป็นการพัฒนาซอฟต์แวร์ที่ใช้แนวคิดของการวนซ้ำ (Iteration) หลายรอบและการพัฒนาแบบเพิ่มพูน (Incremental) โดยทำการแยกซอฟต์แวร์ออกเป็นส่วน ๆ แล้วทำการพัฒนารอบละส่วนจนครบจำนวนรอบเท่ากับซอฟต์แวร์ที่แบ่งไว้ จะประกอบไปด้วย 4 ระยะดังนี้

- ระยะอินเซ็ปชัน (Inception)
- ระยะอีแลบบอเรชัน (Elaboration)
- ระยะคอนสตักชัน (Construction)
- ระยะทรานสิชัน (Transition)

2.4.1.1. ระยะอินเซ็ปชัน

เป็นขั้นตอนในการกำหนดขอบเขตของปัญหา โครงสร้างของผลิตภัณฑ์ ตลอดทั้งหน้าที่และความสามารถของผลิตภัณฑ์ เพื่อแบ่งส่วนของซอฟต์แวร์ออกเป็นส่วน ๆ ศึกษาข้อจำกัดต่าง ๆ ของระบบตลอดทั้งศึกษาแนวทางการนำเทคโนโลยี

ต่าง ๆ มาใช้งาน ศึกษาถึงความสัมพันธ์ที่มีต่อระบบอื่น ๆ ตลอดทั้งระบุข้อมูลที่จะนำเข้าและผลลัพธ์ที่จะได้จากระบบ ในขั้นตอนนี้จะใช้ แบบจำลองยูสเคส (Use case Model) เป็นเครื่องมือสำหรับกำหนดขอบเขตของปัญหาเพื่อให้ผู้พัฒนาและผู้ที่เกี่ยวข้องเข้าใจได้ง่ายสะดวกในการพัฒนาระบบ

2.4.1.2. ระยะอีแลบบอเรชัน

เป็นระยะที่จะทำการออกแบบสถาปัตยกรรมเบื้องต้นของซอฟต์แวร์ โดยศึกษาและวิเคราะห์จากขอบเขตของปัญหาและโครงสร้างของซอฟต์แวร์เบื้องต้นที่ได้ทำการศึกษาไว้จากระยะที่ 1 นำมาประเมินความเสี่ยงในด้านต่าง ๆ เช่น ทางด้านเทคนิค ด้านงบประมาณ เป็นต้น แล้วนำมากำหนดเป็นแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันเพื่อช่วยในการประเมินความเสี่ยงต่าง ๆ ดังกล่าว โดยแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันจะต้องสัมพันธ์กับยูสเคสที่ได้ทำการศึกษาไว้จากระยะที่ 1

2.4.1.3. ระยะคอนสตักชัน

เป็นระยะที่จะพัฒนาซอฟต์แวร์ต่อจากระยะที่ 2 เพื่อพัฒนาต้นแบบของผลิตภัณฑ์ โดยใช้แนวคิดการวนซ้ำและการพัฒนาแบบเพิ่มพูน แบ่งซอฟต์แวร์ออกเป็นส่วน ๆ แล้วแยกพัฒนาทีละส่วน ซึ่งในระยะนี้จะแบ่งออกเป็น 5 ขั้นตอนดังนี้

- วางแผน (Planning) เพื่อแจกแจงรายละเอียดของผลิตภัณฑ์ ได้แก่ ความสามารถ หน้าที่ เพื่อให้ผู้ร่วมทีมพัฒนาเข้าใจ
- การวิเคราะห์ (Analysis) ทำการศึกษารายละเอียดโดยใช้แผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันที่ได้จากระยะที่ 2 เพื่อนำมาวิเคราะห์หาแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันที่เหมาะสมกับระบบย่อยส่วนที่ 1
- การออกแบบ (Design) เป็นระยะที่นำแผนภาพต่าง ๆ ที่ได้จากการวิเคราะห์ในขั้นตอนที่ 2 มาออกแบบส่วนที่ติดต่อกับผู้ใช้ รวมทั้งการออกแบบโครงร่างฐานข้อมูลเชิงตรรกะ (Logical Database Schema)
- การพัฒนาต้นแบบโปรแกรม (Prototype Implementation) โดยทำการแปลงแผนภาพต่าง ๆ ที่ได้จากการศึกษาดังกล่าวข้างต้นออกมาเป็นโปรแกรม รวมทั้งทำการแปลงโครงร่างฐานข้อมูลเชิงตรรกะเป็นโครงร่างฐานข้อมูลเชิงกายภาพ (Physical Database Schema) กำหนดมาตรฐานและรูปแบบการโต้ตอบระหว่างโปรแกรมและผู้ใช้
- การทดสอบ (Testing) การทดสอบโปรแกรมที่ได้จากการพัฒนา เป็นการค้นหาข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นจากโปรแกรมที่พัฒนาเพื่อนำมาปรับปรุงแก้

ไขให้โปรแกรมทำงานอย่างมีประสิทธิภาพ ประกอบไปด้วย การทดสอบหน่วยของโปรแกรม (Program Unit Test) การทดสอบการเชื่อมต่อของโปรแกรม (Integration) การตรวจสอบก่อนการรับมอบโปรแกรม (Acceptance Test)

เมื่อได้ดำเนินการพัฒนาระบบย่อยที่ 1 ตามขั้นตอนที่ 1 ถึงขั้นตอนที่ 5 จนครบแล้ว จากนั้นจะเริ่มการพัฒนาระบบย่อยที่ 2 และส่วนอื่น ๆ ต่อไปจนครบ ซึ่งในระหว่างการพัฒนานี้จะมีการวิเคราะห์แผนภาพเดิมเพื่อหาความเหมาะสม และจะทำการเพิ่มแผนภาพใหม่ตามความจำเป็น ซึ่งจะส่งผลให้ได้ระบบที่สมบูรณ์ยิ่งขึ้น

2.4.1.4. ระยะทรานสิชัน

เป็นระยะที่ทำการปรับปรุงต้นแบบที่ได้จากพัฒนาในระยะที่ 3 เพื่อนำมาปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ พร้อมทั้งพัฒนาส่วนประกอบอื่นเพื่อให้ซอฟต์แวร์ทำงานได้อย่างมีประสิทธิภาพ และทำการออกแบบวิธีการติดตั้ง วิธีการบำรุงรักษา วิธีการแก้ไขเมื่อพบข้อบกพร่องและรวบรวมส่วนประกอบอื่น ๆ ของซอฟต์แวร์เพื่อบรรจุลงในผลิตภัณฑ์พร้อมที่จะนำไปใช้งาน

2.4.2. แนวคิดการออกแบบซอฟต์แวร์เชิงวัตถุ

แนวคิดพื้นฐานเชิงวัตถุจะประกอบไปด้วยส่วนต่าง ๆ ดังนี้

2.4.2.1. วัตถุและคลาส (The Object-&-Class)

- วัตถุ คือสิ่งใด ๆ ก็ตามที่ประกอบไปด้วยสถานะ (State) พฤติกรรม (Behavior) และสามารถระบุ (Identify) ได้
- คลาส คือเซตของวัตถุซึ่งมีลักษณะประจำ และพฤติกรรมแบบเดียวกัน
- การค้นหาคลาสและวัตถุของแต่ละระบบ การค้นหาจะมองจากโครงสร้าง (Structure) ระบบภายนอก (External System) ซึ่งจะค้นหาได้ตามตัวอย่างตารางที่ 2.1 การค้นหาให้พิจารณาจาก แคทเทกอรีทางซ้ายตาราง ดังตัวอย่างในสดมภ์กลาง เมื่อได้คลาสแล้วให้นำคลาสไปใส่ในสดมภ์ขวา พิจารณาตามขั้นตอนนี้จนหมดทุกระบบย่อย

คลาสแคเทกอรีตรวจสอบ ระบบย่อย.....

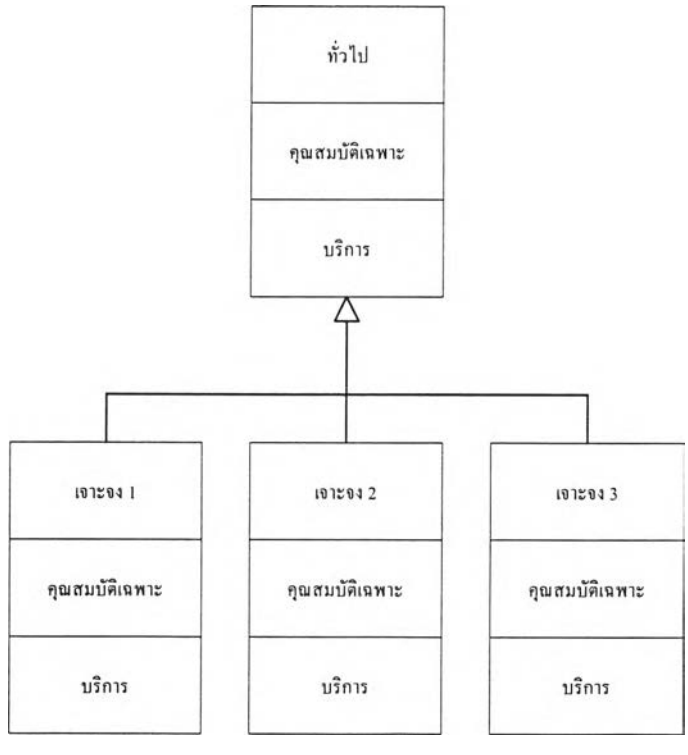
แคเทกอรี (Category)	ตัวอย่าง	รายชื่อคลาส
วัตถุที่จับต้องได้ทางกายภาพ	ที่ทำการไปรษณีย์ เครื่องบิน	
ข้อกำหนด การออกแบบ หรือคำบรรยายของสิ่งของ	ข้อกำหนดของผลิตภัณฑ์ คำอธิบายเที่ยวบิน	
สถานที่	ร้านค้า สนามบิน	
ทรานแซกชัน	การขาย การจ่าย การสำรอง	
บทบาทของคน	แคชเชียร์ นักบิน	
คอนเทนเนอร์ของสิ่งของ	ร้านค้า ถังบรรจุ เครื่องบิน	
สิ่งที่อยู่ภายในคอนเทนเนอร์	ชิ้นสิ่งของ ผู้โดยสาร	
สิ่งอื่นๆ ได้แก่คอมพิวเตอร์ อิเล็กโทร-แมคคานิคัลสำหรับควบคุมระบบ	ระบบพิสูจน์ทราบในบัตรเครดิต ระบบควบคุมการจราจรทางอากาศ	
แนวคิดทางนามธรรม	หิว กลัว	
หน่วยงาน	แผนกขาย หน่วยงานในสายการบิน	
เหตุการณ์	ขาย โจรกรรม การประชุม บิน ชน ลงจอด	
กระบวนการ	กระบวนการขาย กระบวนการจองที่นั่ง	
กฎและนโยบาย	นโยบายการคืน นโยบายการยกเลิก	
บัญชีรายชื่อ	บัญชีรายชื่อผลิตภัณฑ์ บัญชีรายชื่อชิ้นส่วน	
ระเบียนการเงิน งาน ติดต่อ วัตถุตามกฎหมาย	รับ บัญชีแยกประเภท สัญญาการจ้างงาน บันทึกการบำรุงรักษา	
มาตรวัดทางด้านการเงินและบริการ	เครดิต หุ้น	
คู่มือ หนังสือ	คู่มือพนักงาน คู่มือตรวจซ่อม	

ตารางที่ 2.1 ข้อพิจารณาเบื้องต้นการค้นหาคลาสและวัตถุ (Class and Object)

2.4.2.2. โครงร่าง (Structure)

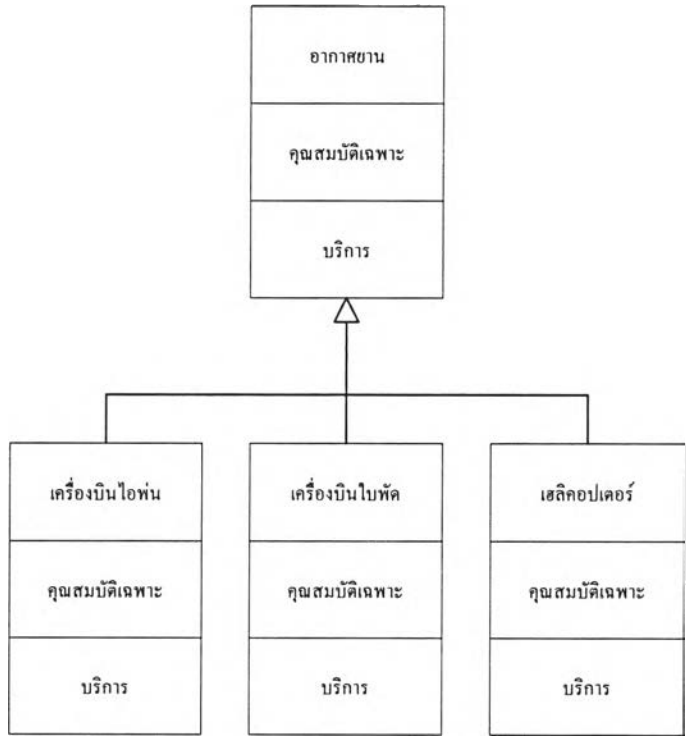
เป็นการแสดงถึงขอบเขตปัญหาของระบบที่กำลังจะพัฒนา โดยโครงร่างจะแสดงความสัมพันธ์ระหว่างคลาสและวัตถุว่ามีความสัมพันธ์กันอย่างไร โครงร่างแบ่งออกเป็น 2 อย่าง

- การกำหนดโครงร่างทั่วไป (Generalization-Specification Structure) ซึ่งเป็นโครงร่างที่แสดงการสืบทอดคุณสมบัติ (Inheritance) ของวัตถุ ดังแสดงในรูปที่ 2.6 และรูปที่ 2.7 เป็นตัวอย่างโครงร่างทั่วไป
- การกำหนดโครงร่างทุกส่วน (Whole-Part Structure) เป็นโครงร่างที่แสดงถึงส่วนประกอบที่สัมพันธ์กันของวัตถุที่ไม่ได้แสดงถึงการสืบทอดคุณสมบัติของวัตถุ แต่แสดงความสัมพันธ์ในลักษณะวัตถุหนึ่งประกอบด้วยอีกวัตถุหนึ่ง ดังแสดงในรูปที่ 2.8 และรูปที่ 2.9เป็นตัวอย่างโครงร่างทุกส่วน

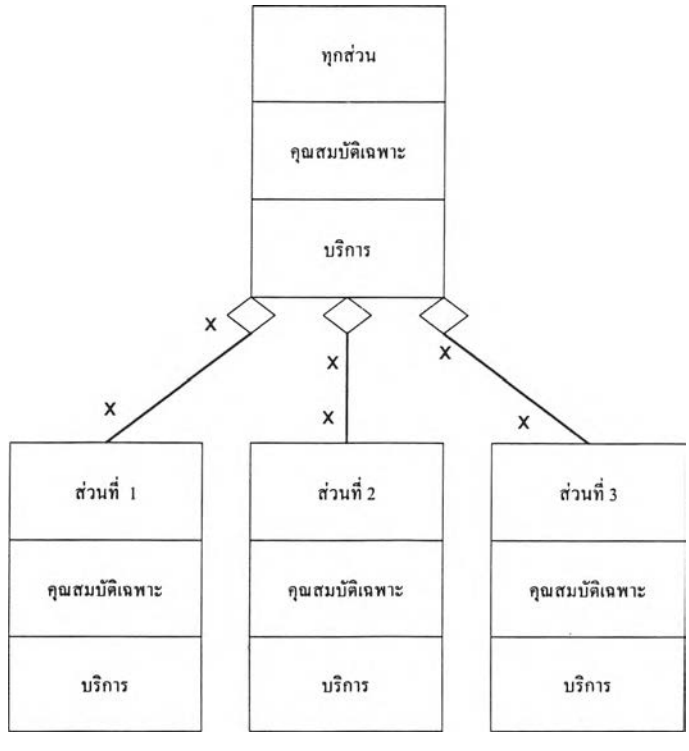


รูปที่ 2.6 แสดงแผนภาพโครงร่างทั่วไป

การ
จุฬาลงกรณ์มหาวิทยาลัย

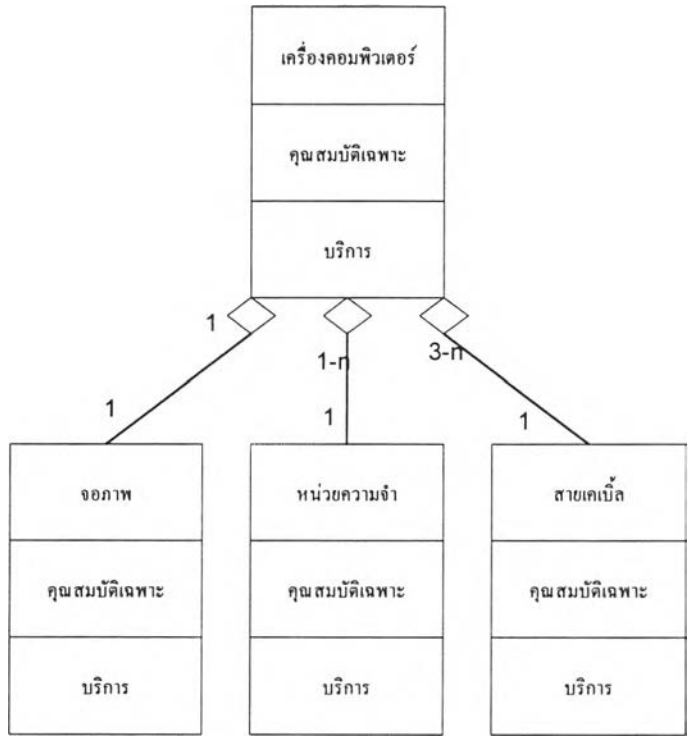


รูปที่ 2.7 แสดงตัวอย่างแผนภาพโครงร่างทั่วไป



หมายเหตุ x หมายถึง ข้อบังคับการเชื่อมโยง ที่เป็นไปได้ 0-n, 1-n เช่น 1-3, 1-5 เป็นต้น

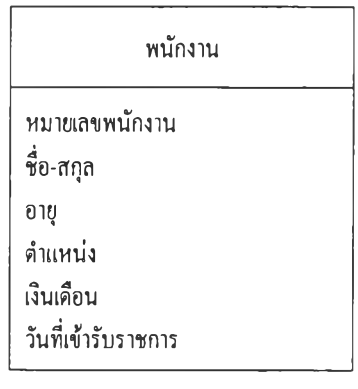
รูปที่ 2.8 แสดงแผนภาพโครงร่างทุกส่วน



รูปที่ 2.9 แสดงตัวอย่างแผนภาพโครงร่างทุกส่วน

2.4.2.3. คุณสมบัติเฉพาะ (Attribute)

คุณสมบัติเฉพาะเป็นส่วนประกอบของข้อมูลที่ใช้แสดงอินสตันธ์ (Instance) ของคลาส คุณสมบัติเฉพาะของแต่ละคลาสจะแตกต่างกันออกไป เพื่อแสดงความสัมพันธ์ระหว่างคลาสและวัตถุ ตัวอย่างของคุณสมบัติเฉพาะแสดงดังรูปที่ 2.10

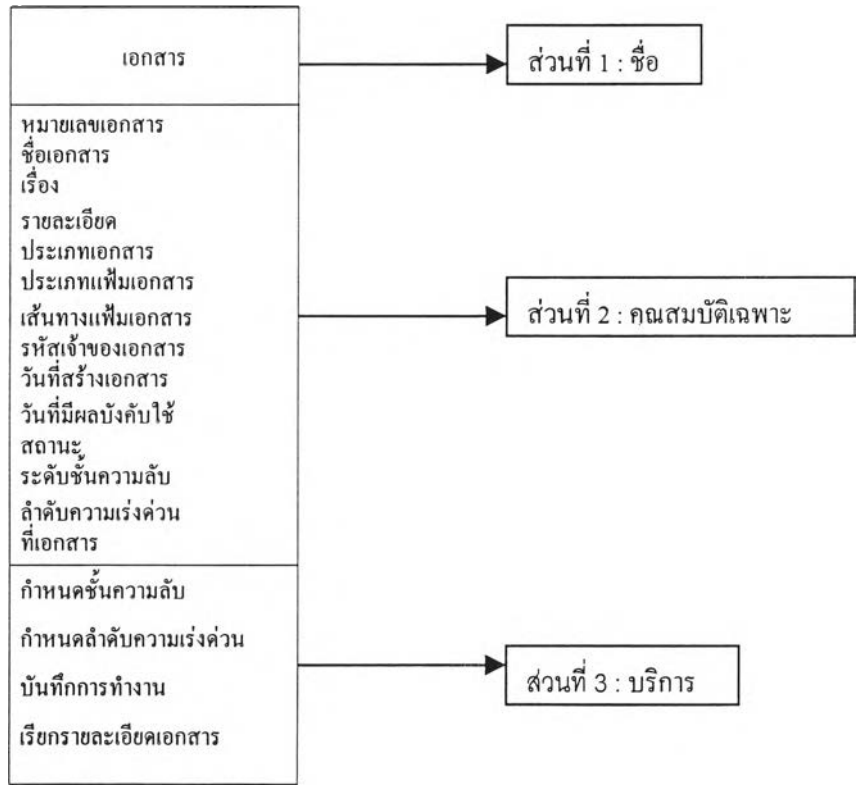


รูปที่ 2.10 แสดงตัวอย่างคุณสมบัติเฉพาะ (Attribute)

2.4.2.4. แผนภาพคลาส

แผนภาพคลาสเป็นแผนภาพหลักในแบบจำลองเชิงวัตถุ แสดงนามธรรม (Abstraction) ที่สำคัญของระบบและความสัมพันธ์ระหว่างนามธรรมเหล่านั้นคลาส คุณสมบัติเฉพาะ และบริการ

คลาสแต่ละคลาสจะถูกแสดงโดยใช้แบบจำลองเชิงวัตถุด้วยสัญญลักษณ์รูปสี่เหลี่ยมซึ่งประกอบไปด้วย 3 ส่วนดังนี้ ส่วนที่ 1 ชื่อคลาส ส่วนที่ 2 คุณสมบัติเฉพาะ ส่วนที่ 3 บริการ ดังตัวอย่างรูปที่ 2.11



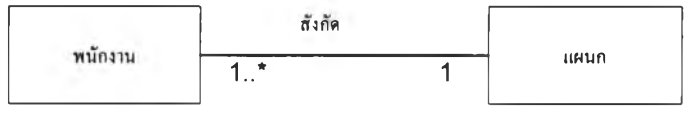
รูปที่ 2.11 สัญญูรูปและส่วนประกอบอื่น ๆ ของคลาส

2.4.2.5. ความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาสจะใช้เส้นตรงลากจากคลาสหนึ่งไปยังอีกคลาสหนึ่งซึ่งความสัมพันธ์ระหว่างคลาสมีหลายรูปแบบ

2.4.2.5.1. ความสัมพันธ์แบบแอซโซซิเอชั่น (Association)

เพื่อแสดงโครงร่างการขึ้นต่อกันระหว่างวัตถุของคลาสที่ต่างกัน โดยข้อความบนเส้นใช้อธิบายความสัมพันธ์ เช่น คลาสพนักงานกับคลาสแผนกมีความสัมพันธ์กันแบบแอซโซซิเอชั่น ดังตัวอย่างรูปที่ 2.12



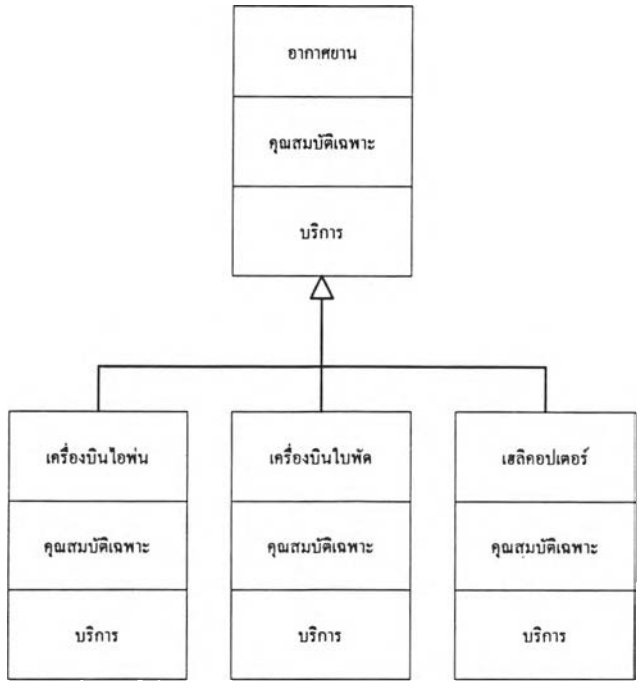
รูปที่ 2.12 ตัวอย่างความสัมพันธ์แบบแอซโซซิเอชั่น

2.4.2.5.2. แอกริเกชัน (Aggregation)

เป็นส่วนแสดงว่าคลาสส่วนประกอบขึ้นอยู่กับคลาสแม่ คือคลาสส่วนประกอบไม่สามารถถูกสร้างได้จนกว่าคลาสแม่จะถูกสร้าง โดยทำนองเดียวกันคลาสส่วนประกอบไม่สามารถถูกทำลายโดยคลาสอื่นยกเว้นคลาสแม่ของมัน แอกริเกชั่นมี 2 แบบคือ โดยอ้างอิง (by Reference) และโดยค่า (By Value)

2.4.2.5.3. การถ่ายทอด (Inheritance)

ความสัมพันธ์แบบการถ่ายทอดถูกใช้เมื่อคลาสหนึ่งมีส่วนร่วมของโครงร่างและคุณสมบัติกับคลาสอื่น มีลักษณะเหมือนกับความสัมพันธ์แบบโครงร่างทั่วไป ในแบบจำลองเชิงวัตถุเรียกคลาสที่มีคุณสมบัติทั่วไปว่า ซุปเปอร์คลาส (SuperClass) และเรียกคลาสที่มีการถ่ายทอดคุณสมบัติเพิ่มเติมว่าซับคลาส (SubClass) ซับคลาสมีคุณสมบัติและบริการเหมือนซุปเปอร์คลาสแต่อาจมีคุณสมบัติเฉพาะและบริการเพิ่ม ดังตัวอย่างในรูปที่ 2.13



รูปที่ 2.13 ตัวอย่างแผนภาพแสดงการถ่ายทอด

2.4.2.5.4. การขึ้นต่อกัน (Dependency)

คลาสที่มีความสัมพันธ์กันต้องการเรียกใช้บริการของอีกคลาสหนึ่งในแบบจำลองเชิงวัตถุเรียกความสัมพันธ์แบบนี้ว่าการขึ้นต่อกัน คือคลาสของผู้ขอบริการขึ้นอยู่กับบริการของคลาสผู้ให้บริการ แต่ไม่มีการขึ้นต่อกันภายในโครงสร้างของคลาส

2.4.2.6. บริการ (Services)

กำหนดบริการของแต่ละวัตถุเพื่อเป็นการตอบรับเมื่อวัตถุนั้นได้รับการตอบรับจากวัตถุอื่น ซึ่งโดยทั่วไปมี 2 ชนิดได้แก่

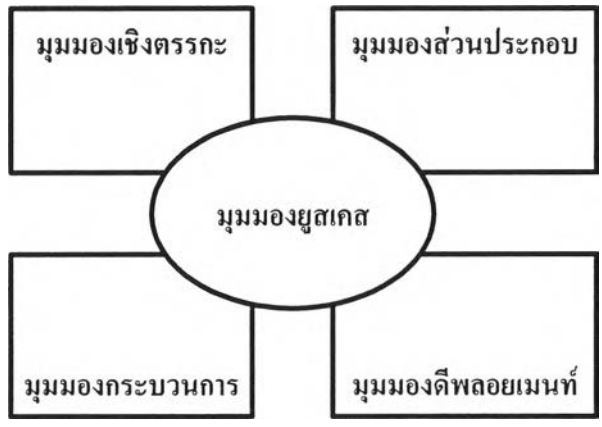
2.4.2.6.1. บริการพื้นฐานของทุกคลาส ซึ่งเป็นบริการที่ปกติจะไม่ถูกแสดงในส่วนบริการในแผนภาพแสดงคลาสและวัตถุ บริการพื้นฐานชนิดนี้จะเด่นชัดและคงอยู่ในทุกๆ คลาส บริการพื้นฐานเหล่านี้ได้แก่ การสร้างวัตถุ (Create Object) การสืบค้น (Search) การรับค่าและตั้งค่าคุณสมบัติเฉพาะ การเพิ่มและการลบความสัมพันธ์ของวัตถุ และการทำลายวัตถุ

2.4.2.6.2. บริการที่ขึ้นอยู่กับขอบเขตปัญหาที่กำหนด เป็นบริการที่กำหนดขึ้นโดยเฉพาะสำหรับปัญหาที่ทำการศึกษา

2.4.3. การออกแบบสถาปัตยกรรมซอฟต์แวร์โดยใช้ภาษายูเอ็มแอล (Software Architectureal Design Using UML)

เป็นการกำหนดโครงสร้างซอฟต์แวร์โดยใช้ภาษายูเอ็มแอลอธิบายหรือแทนซอฟต์แวร์ในมุมมองต่าง ๆ ซึ่งในภาษายูเอ็มแอลจะแบ่งออกเป็น 5 มุมมองตามรูปที่ 2.14 ดังนี้

จากรูปที่ 2.14 เป็นการอธิบายแนวคิดในการนำเสนอระบบในมุมมองต่าง ๆ ซึ่งทุกมุมมองจะเป็นการนำเสนอโดยมีวัตถุประสงค์ที่แตกต่างกัน และนำเสนอบุคคลที่แตกต่างกันออกไป แต่โดยตั้งอยู่บนพื้นฐานที่ว่า ไม่สามารถนำเสนอระบบโดยใช้มุมมองทุกมุมมองได้ ในภาษายูเอ็มแอลได้ออกแบบมุมมองให้มี 5 มุมมองดังนี้ (ดูรูปที่ 2.14 ประกอบ)



รูปที่ 2.14 แสดงมุมมองของสถาปัตยกรรมซอฟต์แวร์ในภาษายูเอ็มแอล

2.4.3.1. มุมมองเชิงตรรกะ (Logical View) เป็นมุมมองที่แสดงถึงองค์ประกอบระดับตรรกะ ได้แก่ คลาส ความสัมพันธ์ระหว่างคลาส และกลุ่มของคลาส

2.4.3.2. มุมมองกระบวนการ (Process View) เป็นมุมมองที่ประกอบไปด้วย กระบวนการ เทร็ด การสื่อสารระหว่างกระบวนการ

2.4.3.3. มุมมองดีพลอยเมนท์ (Deployment View) เป็นส่วนที่แสดงส่วนประกอบทางกายภาพ เช่น ระบบฮาร์ดแวร์ที่ใช้เชื่อมต่อ เป็นต้น

2.4.3.4. มุมมองส่วนประกอบ (Component View) เป็นส่วนที่แสดงโมดุล และระบบย่อยต่าง ๆ

2.4.3.5. มุมมองยูสเคส (Use Case View) เป็นการอธิบายระบบด้วยทรานแซกชันการทำงานต่าง ๆ ในมุมมองผู้ใช้ ซึ่งเป็นการเริ่มต้นการพัฒนาระบบ

2.4.4. ภาษายูเอ็มแอล (UML Language)

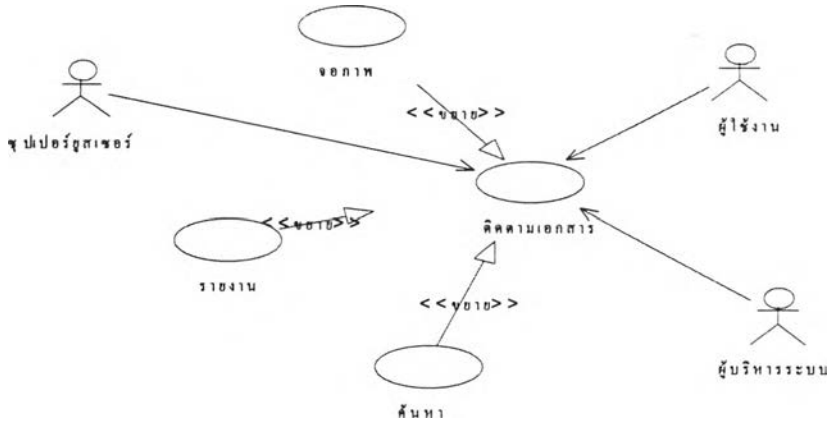
เป็นภาษาจินตทัศน์ (Visual Language) ที่ใช้ในการสร้างหรือแสดงแบบจำลองระบบใด ๆ ที่จะพัฒนาขึ้น และยังเป็นเอกสารประกอบสิ่งที่กำหนดหรือสร้างได้ ยูเอ็มแอล แบ่งออกเป็น 9 แผนภาพดังต่อไปนี้

2.4.4.1. แผนภาพยูสเคส (Use Case Diagram)

เป็นแผนภาพที่เป็นเครื่องมือสำหรับใช้ กำหนดขอบเขตของปัญหาโดยจะเน้นที่รูปภาพและสัญญลักษณ์ตลอดทั้งข้อความเพื่อสื่อให้ผู้พัฒนาระบบและผู้ที่เกี่ยวข้องเข้าใจได้ง่าย ในแผนภาพยูสเคสจะประกอบไปด้วยสัญญลักษณ์ต่าง ๆ ดังต่อไปนี้

- ระบบ (System) แทนด้วยรูปสี่เหลี่ยมหรือไม่มีก็ได้ แสดงถึงขอบเขตของแอคเตอร์ (Actors) ภายนอกกับฟังก์การทำงานภายในระบบ
- ยูสเคส (Use Case) แทนด้วยรูปวงรีภายในประกอบไปด้วยฟังก์ชันการทำงานที่กระทำกับระบบโดยภายในยูสเคสจะมีฟังก์ชันการทำงานที่กระทำกับงานเดียวตั้งแต่ต้นจนจบ
- แอคเตอร์ (Actors) แทนด้วยรูปคน ใช้แทนบุคคลหรือบทบาทที่บุคคลได้รับในหน้าที่กระทำกับระบบนั้น ๆ หรือระบบ หรือส่วนอื่น ๆ ที่ต้องการแลกเปลี่ยนข้อมูลข่าวสารกับระบบ เช่นผู้ใช้งานระบบ ผู้บริหารระบบ ผู้จัดการฝ่ายบุคคล เป็นต้น
- ความสัมพันธ์ (Relationship) แทนด้วยเส้นตรงที่ลากจากแอคเตอร์กับยูสเคส หรือระหว่างยูสเคสกับยูสเคสเพื่อแสดงถึงความสัมพันธ์
- สัญญลักษณ์อื่นๆ นอกจากที่กล่าวเช่น ลูกศรหัวโปร่งทิศทางเดียว ดังตัวอย่างในรูปที่ 2.15

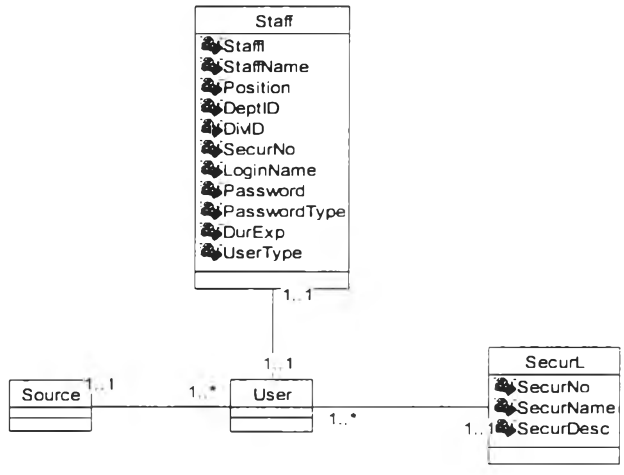
รูปที่ 2.15 อธิบายได้ว่าส่วนขยายของยูสเคสติดตามเอกสารคือ ยูสเคสจอภาพ ยูสเคสรายงาน ยูสเคสค้นหา ลูกศรก้างปลาหมายถึงการติดต่อสื่อสารกันระหว่างยูสเคสกับแอคเตอร์



รูปที่ 2.15 แสดงแผนภาพยูสเคส

2.4.4.2. แผนภาพคลาส (Class Diagram)

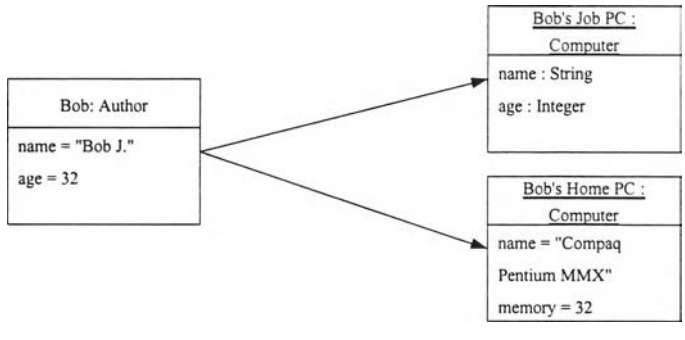
เป็นแผนภาพที่พัฒนาโดยนักวิเคราะห์ระบบ นักออกแบบ หรือผู้สร้างระบบ เพื่อใช้กำหนดแนวคิด และทำความเข้าใจกับระบบที่จะพัฒนาขึ้น ซึ่งประกอบไปด้วย คลาส และความสัมพันธ์ระหว่างคลาส ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 แสดงแผนภาพคลาส

2.4.4.3. แผนภาพออบเจ็กท์ (Object Diagram)

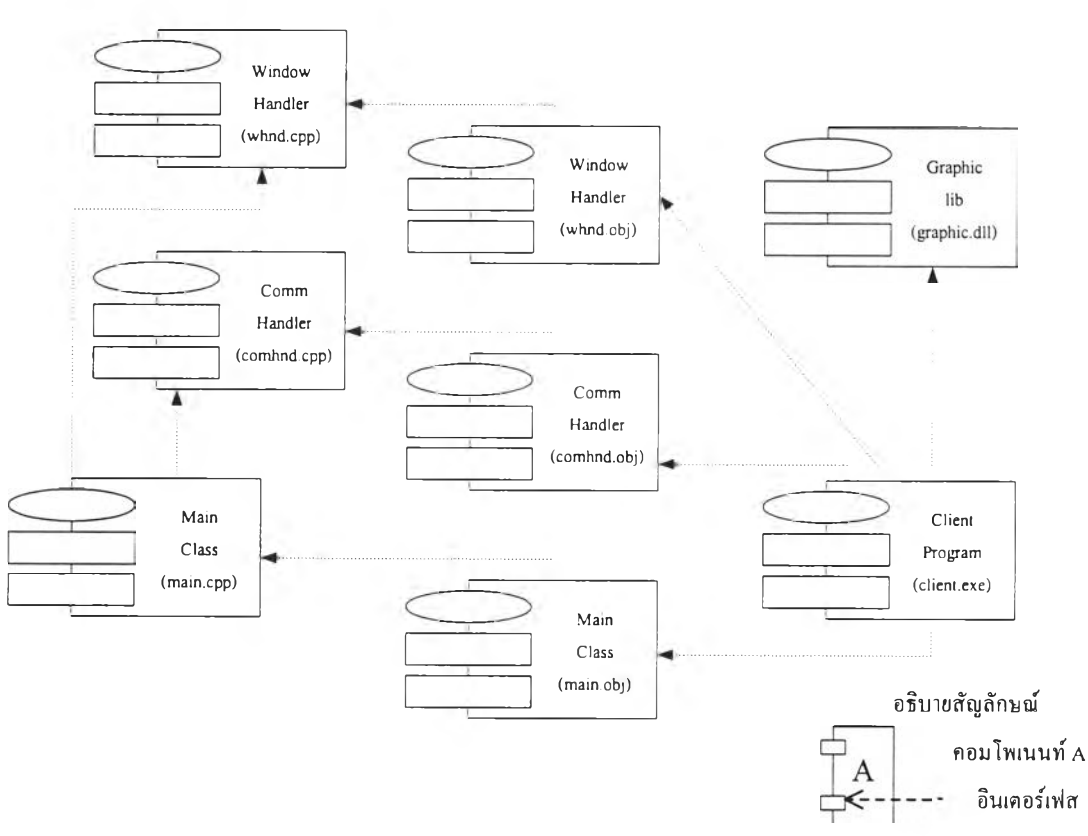
เป็นส่วนที่ใช้แสดงการสร้างออบเจ็กท์และความสัมพันธ์ระหว่างออบเจ็กท์ ซึ่งแสดงอธิบายได้ตามรูปที่ 2.17



รูปที่ 2.17 แสดงแผนภาพออบเจ็กท์[8]

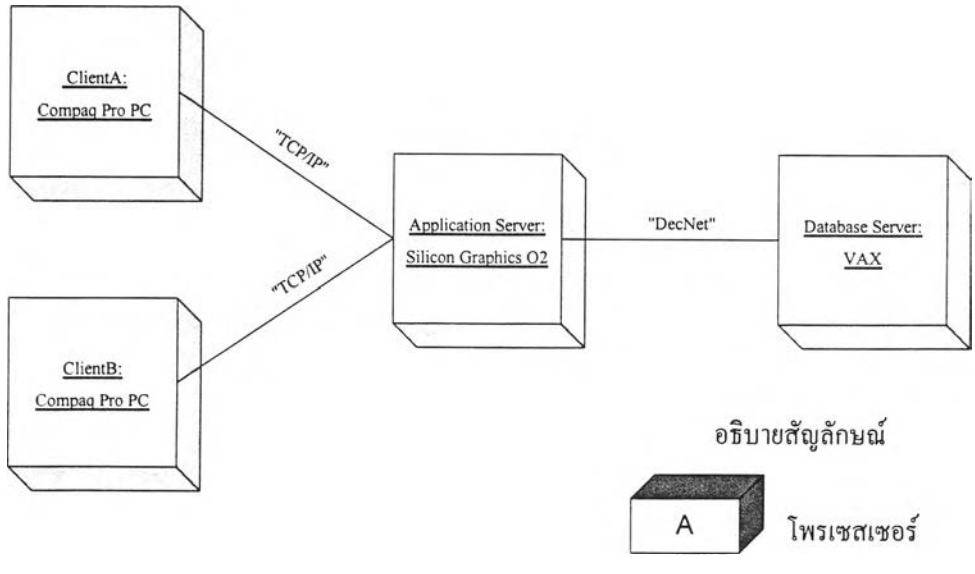
2.4.4.4. แผนภาพส่วนประกอบ (Component Diagram)

เป็นแผนภาพที่ใช้แสดงโครงสร้างทางกายภาพของระบบ ซึ่งประกอบไปด้วยโปรแกรมและความสัมพันธ์ของโปรแกรม อธิบายได้ตามรูปที่ 2.18



รูปที่ 2.18 แผนภาพแสดงส่วนประกอบ[8]

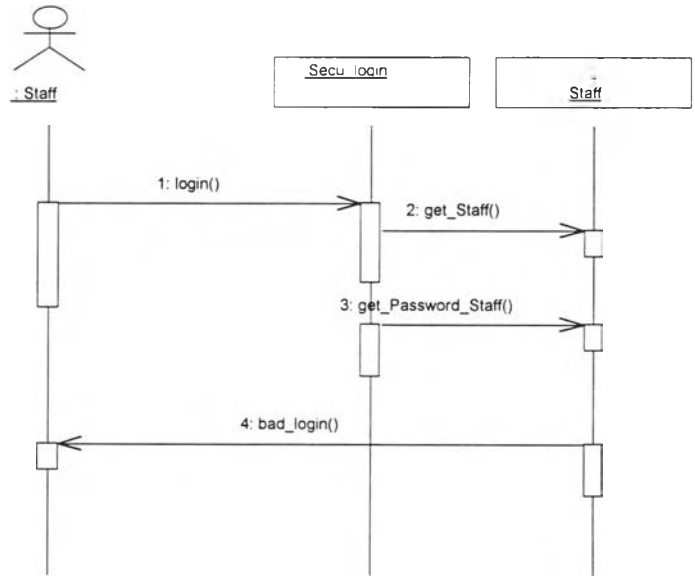
แผนภาพดีพลอยเมนท์ (Deployment Diagram) เป็นแผนภาพที่ใช้แสดงการเชื่อมต่อของระบบฮาร์ดแวร์ และกระจายคอมโพเนนท์ต่าง ๆ อธิบายได้ตามรูปที่ 2.19



รูปที่ 2.19 แผนภาพแสดงดีพลอยเมนท์[8]

2.4.4.5. แผนภาพซีเควน (Sequence Diagram)

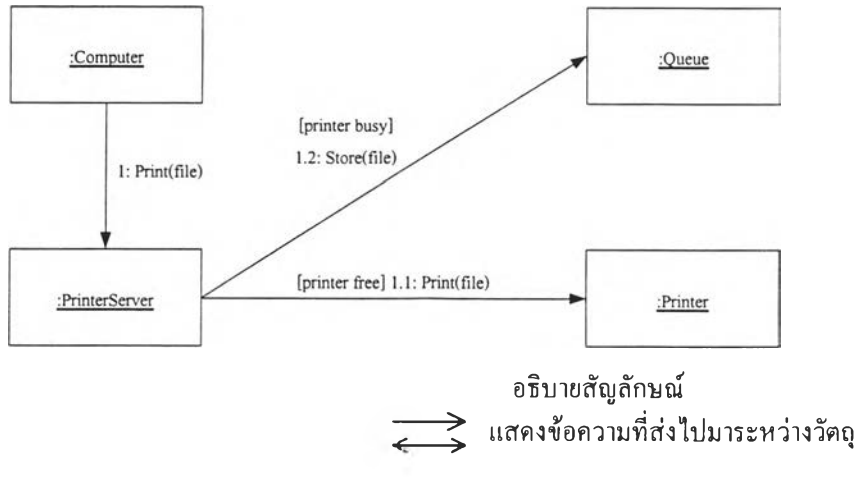
เป็นแผนภาพที่ใช้แสดงพฤติกรรมของระบบและการไหลเวียนของระบบ ตามลำดับเหตุการณ์ที่เกิด อธิบายได้ตามรูปที่ 2.20



รูปที่ 2.20 แสดงแผนภาพซีเควน[8]

2.4.4.6. แผนภาพคอลแลบอเรชัน (Collaboration Diagram)

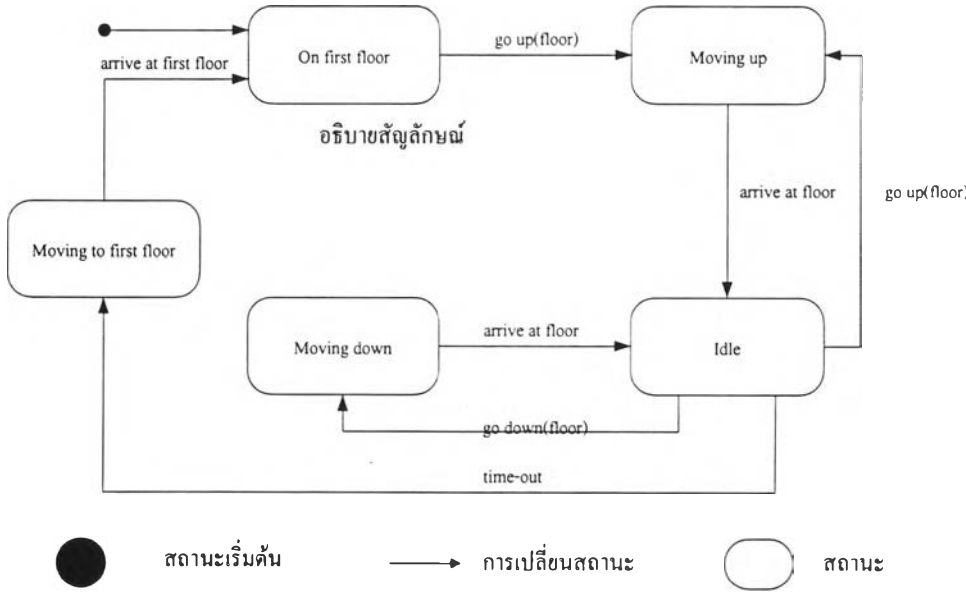
เป็นแผนภาพที่สร้างขึ้นเพื่อแสดงพฤติกรรมของระบบ โดยเน้นที่การรับ-ส่งข้อความและการควบคุม การไหลเวียนของระบบคล้ายแผนภาพซีเควนอธิบายได้ตามรูปที่ 2.21



รูปที่ 2.21 แผนภาพแสดงคอลแลบอเรชัน[8]

2.4.4.7. แผนภาพเสตทชาร์ท (Statechart Diagram)

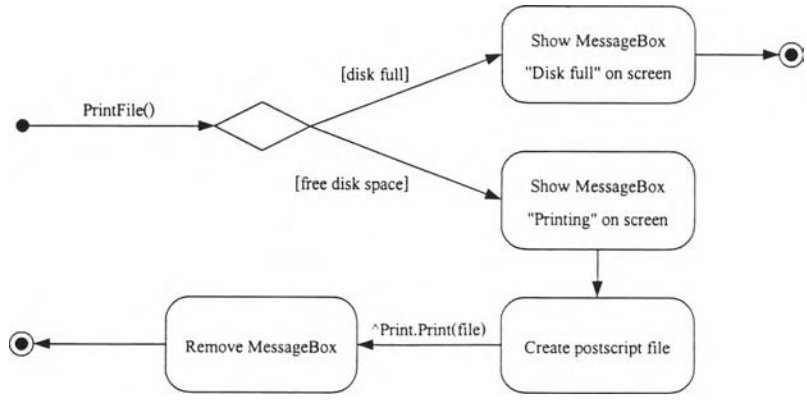
เป็นแผนภาพที่แสดงวงจรชีวิตของคลาส โดยจุดประสงค์เพื่อแสดงพฤติกรรมของคลาสในเหตุการณ์ต่าง ๆ อธิบายได้ตามรูปที่ 2.22



รูปที่ 2.22 แผนภาพแสดงเสตทชาร์ท[8]

2.4.4.8. แผนภาพแอ็คทิวิตี (Activity Diagram)

เป็นแผนภาพที่ใช้แสดงกระแสงานของระบบ อธิบายได้ตามรูปที่ 2.23



รูปที่ 2.23 แผนภาพแสดงแอคทิวิตี[8]

2.4.5. การพัฒนาโปรแกรมด้วยวิธีเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ใช้แนวคิดของชนิดข้อมูลนามธรรม (Abstract Data Type) ในการเขียนโปรแกรม ชนิดข้อมูลนามธรรมนี้เรียกว่าคลาส ประกอบด้วยสมาชิกที่เป็นข้อมูล (Data Elements) หรือแอททริบิวต์ (Attributes) และสมาชิกที่เป็นฟังก์ชัน (Member Function) หรือเมตธอด (Method) วัตถุเป็นตัวตน (Instance) ของคลาส จะมีลักษณะอย่างไรดูได้จากคลาส สมาชิกของคลาสชนิดข้อมูลจะเก็บสถานะต่าง ๆ ของวัตถุ และสมาชิกฟังก์ชันจะใช้ตอบสนองต่อวัตถุอื่นในระบบ วัตถุจึงประกอบด้วยสถานะของมันและบทบาทที่ตอบสนองต่อวัตถุอื่นๆ

การประกาศของคลาสมีการกำหนดส่วนที่เป็น สาธารณะ (Public) ส่วนป้องกัน (protection) และ ส่วนตัว (Private) มีการปิดบัง (Encapsulate) ข้อมูลส่วนตัวของวัตถุ สามารถกระทำต่อวัตถุหรือเข้าถึงข้อมูลของวัตถุด้วยฟังก์ชันหรือเมตธอด โดยทั่วไปภาษาการเขียนโปรแกรมเชิงวัตถุมีการใช้ ลำดับชั้น (hierachy) ของชนิดวัตถุด้วยการถ่ายทอด (Inheritance) คุณลักษณะประจำและเมตธอด

ในปัจจุบันนิยมใช้โปรแกรมเชิงวัตถุมีเพิ่มขึ้นมาก สังเกตได้จากมีโปรแกรมที่สนับสนุนการโปรแกรมเชิงวัตถุ เช่น ไอเฟล (Eiffel) ซี พลัสพลัส (C++) จาวา (Java) เดลไฟล์ (Delphi) วิชวลเบสิค (Visual Basic) เป็นต้น ในการวิจัยนี้ใช้ภาษาวิชวลเบสิค ซึ่งนิยมใช้กันมาก มีการทำงานที่รวดเร็ว และสนับสนุนการทำงานเชิงวัตถุ และใช้การพัฒนาระบบฐานข้อมูลแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) คือ ไมโครซอฟต์แอคเซส (Microsoft Access)

การใช้หลักการของโปรแกรมเชิงวัตถุ มีวัตถุประสงค์คือ

2.4.5.1. เพื่อให้เกิดการทำงานเป็นโมดุล (Modularization) คือสามารถแยกการทำงานเป็นหน่วยจำเพาะย่อยๆ ซึ่งแต่ละหน่วยสามารถทำงานได้อิสระ แล้วจึงทำการรวมแต่ละหน่วยเข้าด้วยกันเป็นโปรแกรมใหญ่

2.4.5.2. ความเข้ากันได้ (Compatibility) เป็นประเด็นสำคัญ เพราะโดยทั่วไปแล้วเป็นการยากที่จะเข้าถึงข้อมูลที่มีโครงสร้างต่างๆกันได้ การใช้โปรแกรมเชิงวัตถุจึงเป็นแนวทางในการแก้ปัญหา

2.4.5.3. การนำข้อมูลกลับมาใช้ใหม่ (Reusability) สำหรับโปรแกรมขนาดใหญ่ ซึ่งมีโครงสร้างข้อมูลขนาดใหญ่ ทำให้เกิดการกระจัดกระจายของตัวแปร และข้อมูลต่างๆ การนำตัวแปรหรือข้อมูลกลับมาใช้ใหม่จึงเกิดความสับสน การโปรแกรม

เชิงวัตถุสามารถที่จะแก้ปัญหานี้ได้โดยที่ผู้ใช้ไม่จำเป็นต้องรู้โครงสร้างทั้งหมดของโปรแกรม

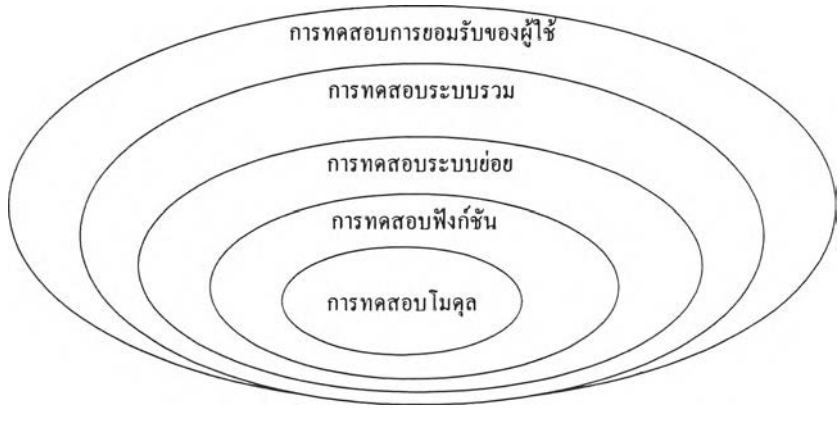
2.4.5.4. ความต่อเนื่อง (Continuity) ซึ่งสามารถใช้คุณสมบัติการโปรแกรมเชิงวัตถุในแง่ของการสรุป (Abstraction) ได้

2.4.6. การทดสอบระบบสารสนเทศ

รูปแบบการทดสอบระบบสารสนเทศอาจมีข้อแตกต่างกันในสภาวะแวดล้อมที่แตกต่างกัน ได้แก่

- การทดสอบโมดุล (Module Testing)
- การทดสอบฟังก์ชัน (Function Testing)
- การทดสอบระบบย่อย (Subsystem Testing)
- การทดสอบระบบรวม (System and Integration testing)
- การทอดสอบการยอมรับของผู้ใช้ (Acceptance Testing)

ดังรูปที่แสดงในรูปที่ 2.24 เป็นรูปแบบที่แตกต่างกันของการทดสอบที่ถูกออกแบบให้เริ่มต้นด้วยการทดสอบความสำเร็จของส่วนที่เล็กของระบบสารสนเทศไปเรื่อยๆ จนถึงการทดสอบระบบสารสนเทศทั้งหมดภายใต้การพัฒนา การทดสอบเสร็จสมบูรณ์เมื่อระบบได้รับการยอมรับจากผู้ใช้



รูปที่ 2.24 แผนภาพแสดงการทดสอบโปรแกรม

จากรูปที่ 2.24 แสดงขั้นตอนการทดสอบโปรแกรม เพื่อหาข้อผิดพลาดของโปรแกรมโดยมีขั้นตอนการทดสอบดังต่อไปนี้

2.4.6.1. เริ่มจากส่วนที่เล็กที่สุดของโปรแกรมคือโมดุลที่นักพัฒนาต้องการทดสอบ

2.4.6.2. การทดสอบฟังก์ชั่นเป็นการทดสอบการทำงานของฟังก์ชั่นที่เกิดจากการนำโมดุลหลาย ๆ โมดุลที่ทำหน้าที่เกี่ยวข้องกันมารวมกันเพื่อให้ทำงานใดงานหนึ่งที่ผู้ใช้ต้องการ ส่วนใหญ่แล้วจะถูกระบุไว้ในเอกสารระบุความต้องการ (Requirements Specification Document) เช่น การพิมพ์รายงาน การแสดงใบกำกับสินค้า เป็นต้น

2.4.6.3. การทดสอบระบบย่อย เป็นการนำฟังก์ชันหลาย ๆ ฟังก์ชันที่ทำหน้าที่คล้ายกันหรือเพื่อวัตถุประสงค์เดียวกันมารวมกันให้เกิดระบบย่อย เพื่อทำหน้าที่อย่างใดอย่างหนึ่ง เช่นในระบบบริหารเอกสารประกอบด้วย ระบบย่อย 4 ระบบได้แก่ ระบบติดตั้งค่าเริ่มต้นระบบ ระบบบริหารเอกสาร ระบบติดตามเอกสาร ระบบรักษาความปลอดภัยเอกสาร เป็นต้น ซึ่งแต่ละระบบย่อยจะทำหน้าที่แตกต่างกันออกไป การทดสอบระบบย่อยเพื่อให้แน่ใจได้ว่าไม่มีข้อผิดพลาดเกิดขึ้นใด ๆ กับซอฟต์แวร์

2.4.6.4. การทดสอบระบบรวมเป็นขั้นตอนการทดสอบที่กระทำหลังจาก ที่มีการนำระบบย่อย ต่าง ๆ มารวมกันเพื่อให้เกิดระบบใดระบบหนึ่ง เช่นระบบบริหารเอกสารเกิดจากระบบย่อยต่าง ๆ ดังที่กล่าวมาแล้ว เพื่อหาข้อผิดพลาดที่เกิดจากการพัฒนาโปรแกรมและต้องการทดสอบการทำงานที่สัมพันธ์การของระบบย่อยต่าง ๆ การทดสอบในระยะนี้อาจจะให้หน่วยงานภายนอกหรือผู้ที่ไม่มีส่วนเกี่ยวข้องร่วมทดสอบด้วยก็ได้

2.4.6.5. การทดสอบการยอมรับจากผู้ใช้งาน เป็นการทดสอบครั้งสุดท้ายที่จะนำไปติดตั้งใช้งานจริง เป็นการทดสอบโดยให้ผู้ใช้โปรแกรมมีส่วนร่วมในการทดสอบ เพื่อให้คำแนะนำในการเปลี่ยนแปลงก่อนนำไปติดตั้ง การทดสอบระยะนี้ยังเป็นการทดสอบว่าระบบที่พัฒนาขึ้นมาเป็นไปตามที่ผู้ใช้ต้องการหรือไม่ สัมพันธ์กับเอกสารระบุความต้องการหรือไม่

2.5. ระบบผู้ขอรับบริการ / ผู้ให้บริการ (Client /Server)

เนื่องจากปัจจุบันการเปลี่ยนแปลงทางด้านฮาร์ดแวร์ การสื่อสารโทรคมนาคม และระบบเครือข่ายได้เป็นไปอย่างรวดเร็ว ทำให้ระบบงานคอมพิวเตอร์ได้พัฒนาไปในทิศทางที่จะมีการใช้

ข้อมูลร่วมกัน และลดขนาดของเครื่องให้มีขนาดเล็กลงแต่มีประสิทธิภาพสูงขึ้น ซึ่งระบบผู้ขอรับบริการ / ผู้ให้บริการ ถือเป็นส่วนหนึ่งในการพัฒนาเทคโนโลยี

องค์ประกอบของระบบผู้ขอรับบริการ / ผู้ให้บริการ มีดังต่อไปนี้

- ผู้ขอรับบริการ (Client) หมายถึงเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้รับ – ผู้ส่งข้อมูลข่าวสาร และคำสั่งจากผู้ใช้ระบบงานไปให้แก่ผู้ให้บริการ (Server) เพื่ออ่านและประมวลผล และส่งกลับมาให้ผู้รับบริการ
- ผู้ให้บริการหมายถึงเครื่องคอมพิวเตอร์ที่ทำหน้าที่รับส่งข้อมูลข่าวสารจากเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้ขอรับบริการเพื่ออ่านและประมวลผล ส่งกลับมาให้เครื่องผู้ขอรับบริการ ซึ่งเครื่องผู้ให้บริการ 1 ตัว อาจจะมีเครื่องผู้ขอรับบริการมากกว่า 1 ตัวก็ได้ หรือในระบบงานจะมีเครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการกี่ตัวก็ได้
- ระบบเครือข่าย (Network) หมายถึงระบบงานที่ประกอบไปด้วย ฮาร์ดแวร์ ซอฟต์แวร์และอุปกรณ์สื่อสารอื่น ๆ เพื่อเป็นทางเดินของข้อมูลข่าวสาร คำสั่งโปรแกรมที่มีการรับส่งระหว่างผู้ขอรับบริการและผู้ให้บริการในระบบงาน

วัตถุประสงค์ของสถาปัตยกรรมแบบผู้ขอรับบริการ / ผู้ให้บริการ

- เชื่อมโยงข้อมูลให้ทุก ๆ เครื่องผู้ให้บริการ และทุก ๆ เครื่องผู้ขอรับบริการที่สามารถเชื่อมโยงข้อมูลกันได้สามารถแลกเปลี่ยนข้อมูลข่าวสารกัน
- เพื่อให้ซอฟต์แวร์ที่อยู่ในระบบสามารถทำงานที่สอดคล้องประสานกันได้ทำให้เกิดประสิทธิภาพสูงสุด
- เพื่อประหยัดค่าใช้จ่ายซึ่งระบบผู้ขอรับบริการและผู้ให้บริการนี้ผู้มีหน้าที่ดูแลสามารถวางแผนได้ว่าจะจัดลำดับความจำเป็นของฮาร์ดแวร์ ซอฟต์แวร์อย่างไร ไม่จำเป็นต้องติดตั้งระบบคอมพิวเตอร์ขนาดใหญ่ในเวลาเดียวกัน
- เพิ่มผลผลิตในการประมวลผลเพราะมีการแบ่งงานกันทำซึ่งอุปกรณ์แต่ละตัว จะถูกกำหนดงานที่เหมาะสมให้ ณ เวลาใดเวลาหนึ่ง (Time Sharing)
- สามารถขยายระบบงานได้ง่าย (Flexibility) เมื่อถึงจุดหนึ่งที่ระบบงานคอมพิวเตอร์ไม่สามารถรองรับงานได้ผู้มีหน้าที่รับผิดชอบสามารถเพิ่มเติมอุปกรณ์เข้าไปในระบบโดยที่ไม่จำเป็นต้องลงทุนในระบบมากนัก
- สามารถใช้ทรัพยากรที่มีอยู่อย่างเต็มประสิทธิภาพ (Resource Utilization) ในงานจัดการระบบสามารถที่จะนำอุปกรณ์ทุกอย่างที่มีอยู่มาใช้ร่วมกันได้เช่นเครื่องพิมพ์ความเร็วสูงเป็นต้น

I19401887

การแบ่งหน้าที่ของผู้ขอรับบริการ / ผู้ให้บริการ

เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธภาพจึงแบ่งหน้าที่ของผู้ขอรับบริการ / ผู้ให้บริการออกเป็น 2 ส่วนหลัก ๆ ดังนี้

ก. กิจกรรมของโปรแกรมประยุกต์ (Application Task)

- ส่วนติดต่อประสานผู้ใช้งาน หมายถึงโปรแกรมประยุกต์ในส่วนที่ผู้ใช้งานเรียกใช้ข้อมูล เช่นคำสั่งโปรแกรม เป็นต้น
- การแสดงผลลัพธ์ที่เกิดขึ้นบนหน้าจอจากการป้อนคำสั่งของผู้ใช้งานเพื่อให้เครื่องคอมพิวเตอร์ทำงาน (Presentation Logic)
- การประมวลผลที่เกิดขึ้นจากการป้อนคำสั่งให้เครื่องคอมพิวเตอร์ทำงาน (Application Logic)
- ส่วนของงานที่จะแสดงให้ผู้ใช้ระบบงานรับทราบว่าเครื่องคอมพิวเตอร์ได้รับทราบคำสั่งหรือได้แสดงผลลัพธ์การทำงานแล้ว (Data Request and Results Acceptance)
- ส่วนของโปรแกรมที่ทำหน้าที่ตรวจสอบความเป็นไปได้ของข้อมูล (Validation) ความปลอดภัยของข้อมูล และความถูกต้องสมบูรณ์ของข้อมูล
- โปรแกรมที่ทำหน้าที่ในการปรับปรุงแก้ไข อ่าน ลบทิ้ง เพิ่มเติม หรือจัดการกับข้อมูลทางด้านกายภาพ (Physical Data Management)

ข. ขนาดที่เหมาะสม (Rightsizing)

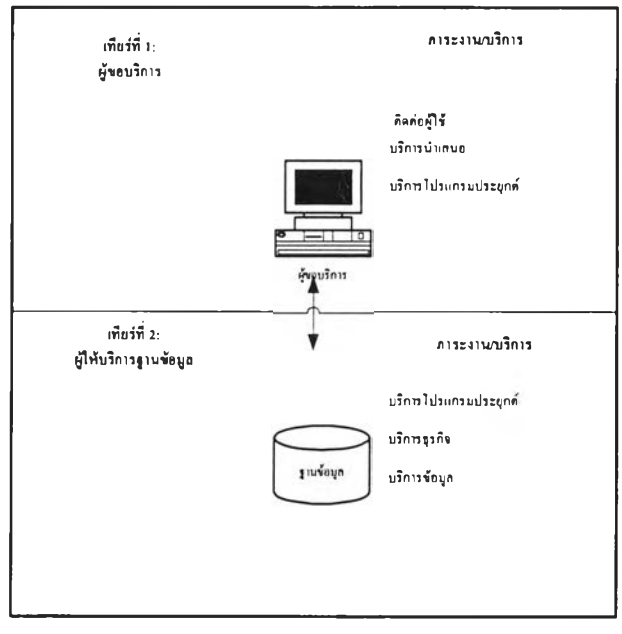
คือการวิเคราะห์ และออกแบบให้ระบบงานสามารถแบ่งหน้าที่ การทำงานการประมวลผลงานออกเป็นส่วน ๆ และให้แต่ละส่วนของงานนั้นทำการประมวลผลที่ เครื่องคอมพิวเตอร์ผู้ขอรับบริการหรือเครื่องผู้ให้บริการที่เหมาะสมได้ครบถ้วนอย่างอิสระโดยเป็นหน้าที่ภารกิจของนักวิเคราะห์และออกแบบระบบงาน ที่จะต้องใช้ความรู้ และประสบการณ์เพื่อพิจารณาว่างานของแต่ละโปรแกรมจะมีความเหมาะสมมากที่สุดในการประมวลผลที่เครื่องให้บริการหรือที่เครื่องขอรับบริการ เพื่อให้เกิด และประสิทธิภาพประสิทธิผลสูงสุด

2.5.1. การทำงานแแบบผู้ขอรับบริการ / ผู้ให้บริการ

สามารถแบ่งการทำงานของสถาปัตยกรรมแบบผู้ขอรับบริการและผู้ให้บริการได้เป็น 3แบบดังนี้

2.5.1.1. แบบทูเทียร์ (Two-Tier Application)

ซึ่งเป็นรูปแบบธรรมดาของระบบผู้ขอรับบริการและผู้ให้บริการ มาจากการแบ่งการทำงานของโปรแกรม (Application) ออกเป็นส่วนของผู้ขอใช้บริการและผู้ให้บริการ ผู้ขอใช้บริการเก็บส่วนแสดงผล ส่วนผู้ให้บริการจะทำหน้าที่เก็บรวบรวมข้อมูล ตัวอย่างระบบประเภทนี้ โปรแกรมส่วนใหญ่จะอยู่บนอินเตอร์เน็ต เช่น จดหมายอิเล็กทรอนิกส์ (E-mail) เทลเนต (Telnet) โกเฟอร์(Gopher) เอฟทีพี (Ftp) เว็บ (Web) เป็นต้นรูปที่ 2.25 เป็นการอธิบายหลักการเบื้องต้นของสถาปัตยกรรมแบบทูเทียร์



รูปที่ 2.25 สถาปัตยกรรมระบบผู้ให้บริการและผู้รับบริการแบบทูเทียร์ (Tow-Tier)[6]

ก. ข้อดีของสถาปัตยกรรมแบบทูเทียร์

เป็นโปรแกรมประยุกต์แบบง่ายไม่ต้องการบำรุงรักษามาก และไม่มีความซับซ้อนมากนักการที่จะพิจารณาว่าโปรแกรมประยุกต์แบบใดที่เหมาะกับสถาปัตยกรรมแบบทูเทียร์ควรพิจารณาดังนี้

- เป็นโปรแกรมประยุกต์ที่ใช้ฐานข้อมูลเดี่ยว
- ฐานข้อมูลบรรจุอยู่ภายในซีพียูเครื่องเดียว
- ฐานข้อมูลไม่มีการเปลี่ยนแปลงบ่อย ๆ ขนาดคงเดิม
- ความต้องการไม่มีการเปลี่ยนแปลงบ่อย
- ผู้ใช้งานไม่มีการเปลี่ยนแปลงบ่อย
- เป็นโปรแกรมประยุกต์ที่สมบูรณ์แล้วไม่ต้องการบำรุงรักษามากนัก

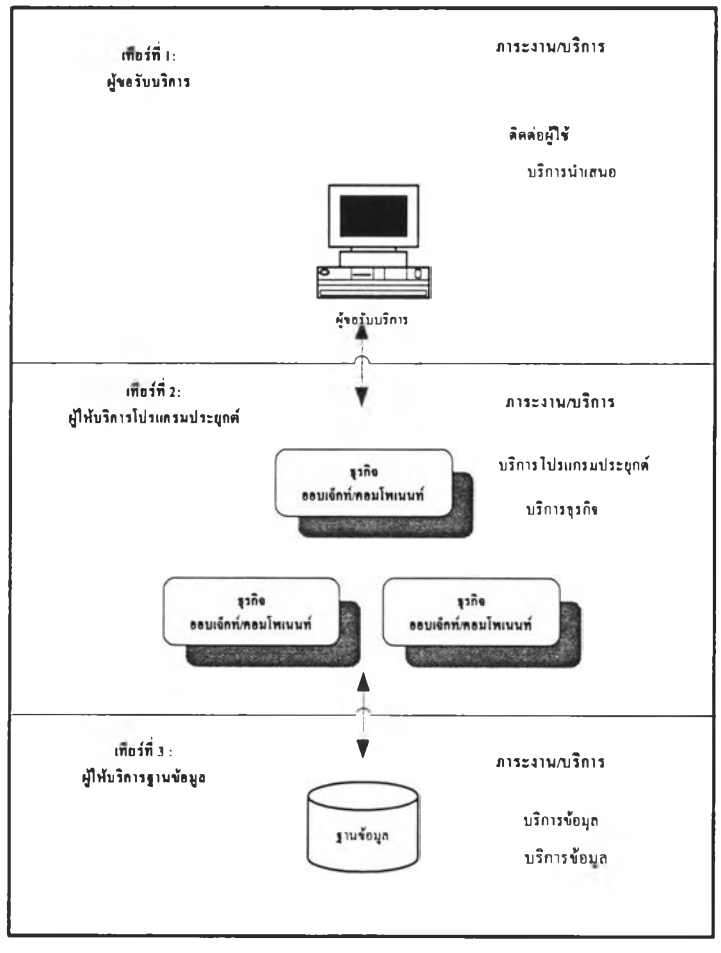
ข. ข้อเสียของสถาปัตยกรรมแบบทูเทียร์

- เมื่อมีความต้องการเพิ่มมากขึ้นทำให้เกิดปัญหาที่เครื่องที่ขอรับบริการเพราะไม่สามารถรองรับข้อมูลที่ซับซ้อนมาก ๆ ได้
- กรณีที่มีการเปลี่ยนแปลงข้อมูลหรือฟังก์ชันการทำงานบางส่วน ผู้ใช้จำเป็นต้องเปลี่ยนแปลง ทดสอบ และแจกจ่ายโปรแกรมในส่วนของเครื่องผู้ขอรับบริการใหม่

2.5.1.2. แบบทรีเทียร์ (Three -Tier Application) เนื่องจากปัญหาของทูเทียร์ที่การติดต่อกับฐานข้อมูลกระทำโดยตรงกับเครื่องผู้ให้บริการ หากมีการเปลี่ยนแปลงใด ๆ ที่เกิดขึ้นกับฐานข้อมูลจำเป็นที่จะต้องมีการเปลี่ยนแปลงที่เครื่องที่ขอรับบริการด้วย การแก้ปัญหานี้โดยการเพิ่มเทียร์ใหม่เข้ามาระหว่างเครื่องที่ขอรับบริการและเครื่องที่ให้บริการ โดยเครื่องขอรับบริการจะติดต่อกับเครื่องให้บริการโดยผ่านทางมิดเดิลเทียร์ (Middle Tier) จากนั้นมิดเดิลเทียร์จะติดต่อกับเครื่องให้บริการต่อไป อธิบายได้ตามรูปที่ 2.26

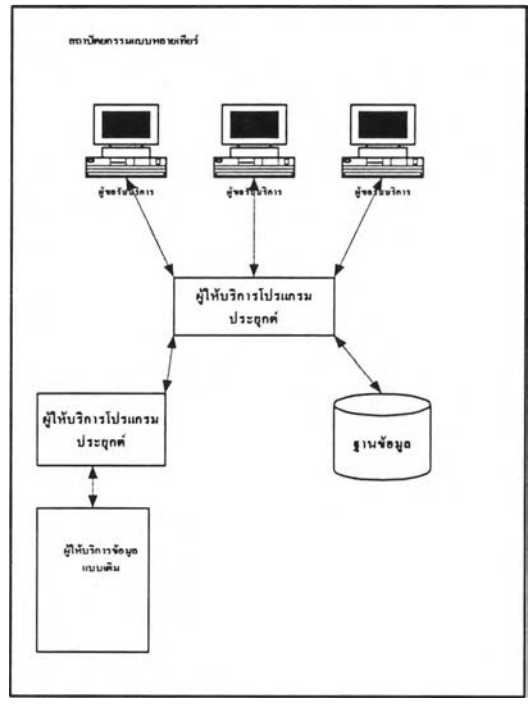
แบบมัลติเทียร์ (Multi-Tier Application) โปรแกรมประยุกต์ โดยทั่ว ๆ ไปจะประกอบด้วยส่วนที่ติดต่อกับผู้ใช้ สำหรับแสดงผลและเก็บรวบรวมข้อมูลเข้ามา กลุ่มฟังก์ชันต่าง ๆ ที่ทำหน้าที่ประมวลผลข้อมูลต่าง ๆ รวมถึงวิธีการเก็บรักษาข้อมูลถึงแม้ฟังก์ชันที่ใช้ในการเก็บรักษาข้อมูล

โดยทั่วไปจะทำงานอยู่ภายใต้เครื่องให้บริการของฐานข้อมูลส่วนกลาง บางครั้งเราเรียกรูปแบบลักษณะการทำงานแบบนี้ว่า โมเดลประยุกต์แบบ 2 ระดับ (Two-Tier Application Model) เนื่องจากโปรแกรมประยุกต์ที่ทำงานเดี่ยว ๆ นั้นมีขนาดใหญ่มาก จึงพัฒนาได้ช้า และต้องการบำรุงรักษาและเนื้อที่ในฮาร์ดดิสก์สูงมาก เมื่อมีการเปลี่ยนแปลงเล็กน้อยก็จะต้องเขียนโปรแกรมทับลงไปใหม่ คอมไพล์ใหม่ ทำให้การเปลี่ยนแปลงไปใช้ระบบอื่น ๆ ทำได้ยากมาก วิธีการแก้ปัญหาดังกล่าวทำได้โดยการแบ่งโปรแกรมประยุกต์เดี่ยว ๆ ออกเป็นโมดุลย่อย ๆ ที่ทำงานร่วมกัน



รูปที่ 2.26 สถาปัตยกรรมระบบผู้ให้บริการและผู้รับบริการแบบทรีเทียร์ (Three-Tier Application)[6]

การแยกส่วนที่ติดต่อกับผู้ใช้ออกมาจากฟังก์ชันอื่น ๆ ในโปรแกรมประยุกต์ ทำให้เราสามารถสร้างโปรแกรมประยุกต์สำหรับเครื่องขอรับบริการขนาดเล็ก ๆ ได้ทำให้โปรแกรมไม่มีความซับซ้อนและไม่ต้องทำงานมากเกินไปบนเครื่องขอใช้บริการทำให้ใช้เนื้อที่น้อยการบำรุงรักษาและการเปลี่ยนแปลงแก้ไขทำได้ง่าย ซึ่งในทางทฤษฎีแล้วการเขียนโปรแกรมประยุกต์สำหรับเครื่องขอใช้บริการแล้วสามารถทำได้โดยภาษาที่แตกต่างกัน เช่น จาวาแอพเพลตเป็นต้น อธิบายได้โดยรูปที่ 2.27



รูปที่ 2.27 สถาปัตยกรรมระบบผู้ให้บริการและผู้รับบริการแบบมัลติเทียร์ (Multi -Tier Architecture)[6]

การพัฒนาระบบงานประยุกต์โดยใช้เทคโนโลยีระบบผู้ขอบริการ / ผู้ให้บริการนับวันจะมีแนวโน้มการนำไปใช้งานสูงขึ้นเนื่องจากความเจริญก้าวหน้าเป็นไปอย่างรวดเร็วของฮาร์ดแวร์ ระบบการสื่อสารโทรคมนาคม ระบบเครือข่ายคอมพิวเตอร์ และระบบดังกล่าวเจริญขึ้นอย่างมีประสิทธิภาพและเป็นที่นิยมอย่างก้าวขวาง ตลอดทั้งการขยายตัวของระบบธุรกิจเป็นตัวกระตุ้นให้มีการพัฒนาระบบอื่นอย่างต่อเนื่องเพื่อสนองความต้องการในการนำไปใช้งาน ดังนั้นระบบผู้ขอใช้บริการและผู้ให้บริการจึงถูกออกแบบมาอย่างพอเหมาะ และเหมาะสมอย่างยิ่งต่อการลงทุนทางด้านระบบสารสนเทศ

2.6. ทฤษฎีเกี่ยวกับฐานข้อมูล

ระบบฐานข้อมูลนับว่าเป็นหัวใจและมีความสำคัญต่อระบบสารสนเทศเป็นอย่างมาก เป็นส่วนที่ใช้เก็บข้อมูลเกี่ยวกับสารสนเทศต่าง ๆ ที่ป้อนเข้าไปและให้ผลลัพธ์ออกมาตามที่ผู้ใช้งานต้องการหรือตามที่ผู้ออกแบบและพัฒนาระบบต้องการ การออกแบบฐานข้อมูลที่ดีจะมีความสำคัญและจะมีส่วนช่วยในการประมวลผล เช่นการค้นหาข้อมูลได้รวดเร็วขึ้น ประหยัดเนื้อที่ในการจัดเก็บ เทคโนโลยีการออกแบบฐานได้รับการพัฒนาต่อกันมาหลายยุคจนถึงปัจจุบันและจะ

พัฒนาต่อไปอีกเรื่อย ๆ ปัจจุบันได้มีการคิดค้นวิธีการและแบบจำลองทางด้านการออกแบบฐานข้อมูลเพื่ออำนวยความสะดวกให้นักออกแบบ ออกแบบฐานข้อมูลได้ง่ายและมีประสิทธิภาพยิ่งขึ้น เช่น แบบจำลองเชิงความสัมพันธ์ (Relational Model) ซึ่งเป็นที่นิยมของนักออกแบบฐานข้อมูล เนื่องจากมีเครื่องมือที่ใช้ในการพัฒนาจำหน่ายอยู่จำนวนมากทั้งยังมีประสิทธิภาพสูง

2.6.1. วงจรชีวิตการพัฒนาระบบฐานข้อมูล

การพัฒนาระบบสารสนเทศโดยทั่ว ๆ ไป จะดำเนินตามขั้นตอนต่าง ๆ ที่กำหนดไว้ในวงจรชีวิตของการพัฒนาระบบสารสนเทศ ซึ่งวงจรชีวิตของการพัฒนาระบบฐานข้อมูลก็ถือเป็นส่วนหนึ่ง ประกอบไปด้วยขั้นตอนการพัฒนาดังนี้

2.6.1.1. การศึกษาและวิเคราะห์ความต้องการของผู้ใช้ กำหนดจุดมุ่งหมาย ปัญหา ขอบเขตตลอดทั้งกฎระเบียบต่าง ๆ เพื่อใช้เป็นแนวทางในการออกแบบฐานข้อมูล

2.6.1.2. นำรายละเอียดที่ได้จากการวิเคราะห์และศึกษาในขั้นต้นมาเป็นข้อมูลในการออกแบบฐานข้อมูลซึ่งมี 3 แนวทางที่นิยมใช้

2.6.1.2.1. ให้ความสำคัญข้อมูล (Data Driven) เป็นแนวทางที่ให้ความสำคัญตัวข้อมูลมากกว่าตัวโปรแกรม โดยทำการออกแบบข้อมูล ให้มีความสมบูรณ์ก่อนการออกแบบโปรแกรมขั้นตอนสามารถอธิบายตามรูปที่ 2.28 ซึ่งแบ่งการออกแบบออกเป็น 3 ขั้นตอนดังนี้

- การออกแบบโครงสร้างฐานข้อมูลในระดับแนวคิด (Conceptual Design) เป็นการออกแบบโครงสร้างหลัก ๆ ของฐานข้อมูลโดยไม่สนใจในรายละเอียดว่าจะใช้อุปกรณ์ใด จัดเก็บอย่างไร เป็นต้น
- การออกแบบฐานข้อมูลในระดับลอจิคัล (Logical Design) เป็นการนำเอาโครงสร้างระดับฐานข้อมูลระดับแนวคิดมาแปลงให้อยู่ในรูปแบบที่กำหนดของ แบบจำลองที่เลือกโดยยังไม่สนใจว่าจะใช้ระบบจัดการฐานข้อมูลตัวใด ในขั้นตอนนี้จะได้โครงสร้างที่เราเรียกว่าเค้าร่าง ลอจิคัล (Logical Schema)

– การออกแบบฐานข้อมูลในระดับกายภาพ (Physical Design) เป็นขั้นตอนที่นำเอาเค้าร่างระดับลอจิคัลมาแปลงให้อยู่ในรูปแบบที่ระบบจัดการฐานข้อมูลกำหนด จะกำหนดโครงสร้างในการจัดเก็บ วิธีการเข้าถึงข้อมูล ในขั้นตอนนี้จะได้เค้าร่างระดับกายภาพ (Physical Schema)สำหรับนำไปใช้ในการออกแบบฐานข้อมูลตัวจริง

รูปที่ 2.28 การออกแบบฐานข้อมูลแบบให้ความสำคัญข้อมูล[7]

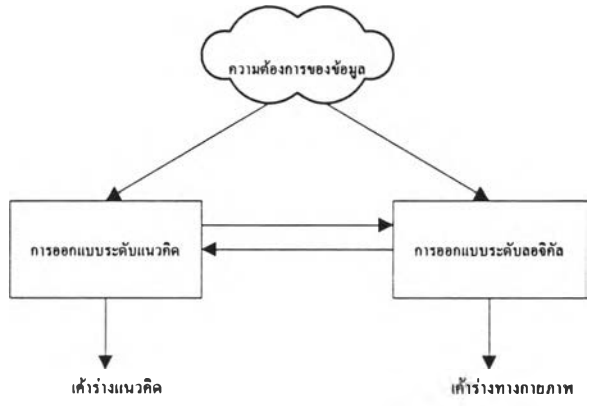
2.6.1.2.2. ให้ความสำคัญกับฟังก์ชัน (Function – Driven Approach) นิยมมาใช้สำหรับออกแบบโปรแกรม ซึ่งแนวทางนี้จะให้ความสำคัญกับตัวโปรแกรมมากกว่าข้อมูลสามารถแบ่งออกเป็น 3 ขั้นตอน

– การวิเคราะห์หน้าที่การทำงานของฟังก์ชัน (Function Analysis) เป็นขั้นตอนที่นำเอาความต้องการต่าง ๆ ที่เกี่ยวข้องกับการประมวลผลมาวิเคราะห์ เพื่อกำหนดเป็นขั้นตอนในการทำงานและข้อมูลต่าง ๆ ที่จำเป็น ในขั้นตอนนี้จะได้ เค้าร่างฟังก์ชัน (Functional Schema)

- การออกแบบโปรแกรมประยุกต์ระดับสูง (High - Level Application Design) เป็นขั้นตอนที่นำเอาเค้าร่างฟังก์ชัน มาเขียนให้อยู่ในรูปแบบมาตรฐาน (Application Specification) ซึ่งแสดงถึงการทำงานและการเข้าถึงข้อมูลของฟังก์ชัน ต่าง ๆ
- การออกแบบโปรแกรมประยุกต์ (Application Program Design) ขั้นตอนนี้เป็นการนำรูปแบบมาตรฐาน (Application Specification) มากำหนดรายละเอียดในรูปแบบของโปรแกรม และเป็นส่วนที่จะนำไปใช้พัฒนาเป็นโปรแกรมต่อไป ซึ่งทั้ง 3 ขั้นตอนสามารถอธิบายโดยรูปที่ 2.29

รูปที่ 2.29 การออกแบบฐานข้อมูลแบบให้ความสำคัญฟังก์ชัน[8]

2.6.1.2.3. ให้ความสำคัญทั้งสองส่วน (Join Data - and Function - Driven Approach) เป็นการออกแบบวิธีผสมระหว่างการให้ความสำคัญของข้อมูล และ ให้ความสำคัญฟังก์ชัน เป็นรูปแบบที่มีการตรวจสอบซึ่งกันและกันทำให้มีความถูกต้องมากขึ้นซึ่งสามารถอธิบายได้จากรูป 2.30



รูปที่ 2.30 การออกแบบฐานข้อมูลแบบให้ความสำคัญทั้งข้อมูลและฟังก์ชัน[7]

2.6.1.3. การนำไปใช้งานกับข้อมูลจริง (Implementation and Loading) เป็นขั้นตอนที่นำเอาโครงร่างที่ได้ออกแบบไว้มาสร้างเป็นฐานข้อมูลและเก็บข้อมูลจริง และทำการแปลงข้อมูลจากระบบเดิมเพื่อนำมาใช้กับฐานข้อมูลที่พัฒนาขึ้น หากระบบเดิมมีการใช้ระบบคอมพิวเตอร์ในการประมวลผล

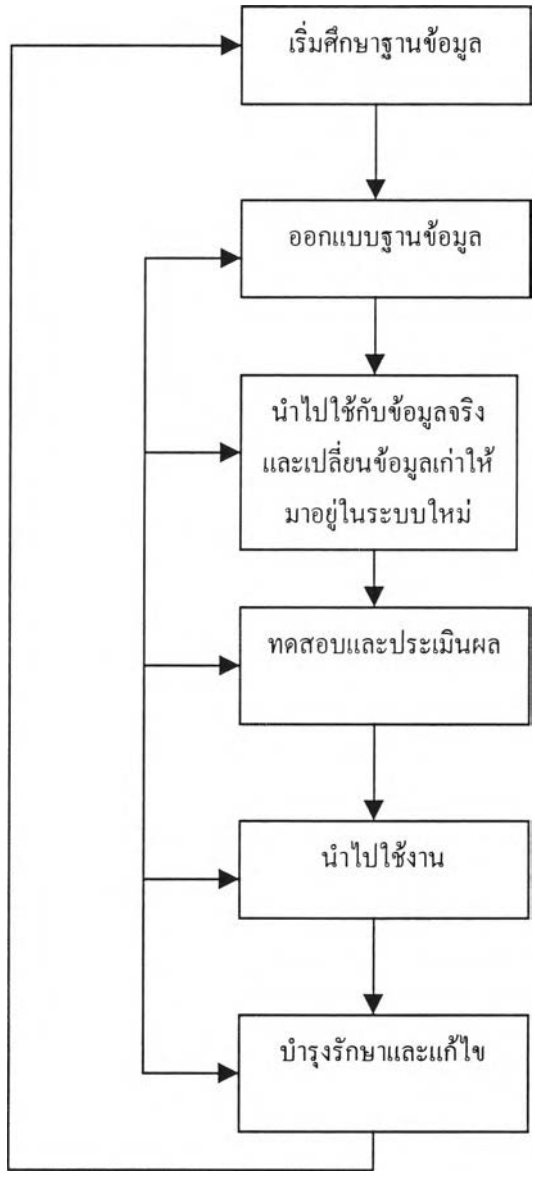
2.6.1.4. การทดสอบและประเมินผล (Testing and Evaluation) เป็นขั้นตอนที่นำระบบฐานข้อมูลมาทำการทดสอบและประเมินผลความสามารถเพื่อเป็นแนวทางสำหรับการปรับปรุงประสิทธิภาพให้ดีขึ้น

2.6.1.5. การนำไปปฏิบัติ (Operation) เป็นขั้นตอนที่นำฐานข้อมูลไปใช้กับระบบงานจริง

การบำรุงรักษาและแก้ไข (Maintenance and Evolution) การบำรุงรักษาหลังจากที่นำฐานข้อมูลไปใช้กับงานจริง เพื่อให้ฐานข้อมูลทำงานได้อย่างมีประสิทธิภาพ หรือเพื่อแก้ไขข้อผิดพลาดที่เกิดขึ้นจากการทำงาน หรือในกรณีที่มีการเปลี่ยนแปลง หรือเพิ่มเติมความต้องการใหม่ ซึ่งขั้นตอนการออกแบบฐานข้อมูลตั้งแต่ต้นสามารถอธิบายได้โดยรูปภาพที่ 2.31

2.6.1.6. ในขั้นตอนการออกแบบทั้งแบบให้ความสำคัญกับข้อมูล และให้ความสำคัญกับฟังก์ชัน จะใช้แบบจำลองนำเสนอ เนื่องจากแบบจำลองเป็นเครื่องมือที่ใช้รูปแทนคำอธิบายต่าง ๆ เพื่อแสดงถึงข้อเท็จจริงของข้อมูล รวมทั้งการกระทำต่าง ๆ ที่เกิดกับข้อมูลนั้นในระดับแนวคิด รูปภาพที่ใช้แทนข้อมูลและข้อเท็จจริงต่าง ๆ จะถูกเชื่อมโยงกันตามโครงสร้างมาตรฐานเพื่อแสดงถึงความเกี่ยวพันของข้อมูลแต่ละส่วน ซึ่งเมื่อนำโครงสร้างมารวมกัน ก็จะเกิดเป็นโครง

สร้างของส่วนการประมวลผล หรือโครงสร้างฐานข้อมูลของระบบสารสนเทศที่จะพัฒนาต่อไป



รูปที่ 2.31 วงจรการพัฒนาระบบฐานข้อมูล[7]

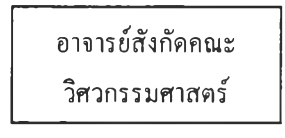
2.6.2. แบบจำลองเอนทิตี-รีเลชันชิฟ (Entity-Relationship Model)

ในการออกแบบฐานข้อมูลที่จะนำไปใช้ในการพัฒนาระบบสารสนเทศต่อไป จะต้องอาศัย แบบจำลองของข้อมูลเพื่อนำเสนอในรายละเอียดต่าง ๆ ที่เกี่ยวข้องกับข้อมูลที่ใช้ในฐานข้อมูล เนื่องจากแบบจำลองเป็นรูปแบบการเสนอรายละเอียดที่เป็นมาตรฐาน จึงสามารถนำไปเสนอต่อผู้ใช้ในระดับที่มีมุมมองที่แตกต่างกันได้เป็นอย่างดี สำหรับแบบ

จำลองที่นิยมใช้ได้แก่ แบบจำลองเอนทิตี-รีเลชันชิฟหรือนิยมเรียกว่าแบบจำลองอีอาร์ (E-R Model) ซึ่งประกอบด้วยส่วนต่างๆ ดังนี้

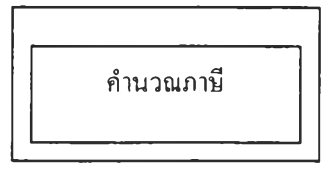
2.6.2.1. เอนทิตี (Entities) เป็นรูปภาพที่ใช้แทนคลาส ของสิ่งของต่าง ๆ ที่สามารถระบุได้ในความเป็นจริง เช่น แผนกในหน่วยงาน พนักงานในบริษัท เป็นต้น หรืออาจเป็นเพียงนามธรรมไม่สามารถจับต้องได้ เช่น จำนวนวันลาของพนักงาน เป็นต้น ซึ่งในแบบจำลองอีอาร์จะแบ่งเป็น 2 แบบคือ

2.6.2.1.1. เอนทิตีแบบทั่วไป (Regular Entities) หรือ เอนทิตีแบบแข็ง (Strong Entities)เป็นเอนทิตี ที่ประกอบด้วยสมาชิกที่มีคุณสมบัติ บ่งบอกถึงเอกลักษณ์ของแต่ละสมาชิกนั้น เช่น เอนทิตีของอาจารย์คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งอาจารย์แต่ละท่านที่สังกัดคณะวิศวกรรมศาสตร์ที่มีหมายเลขประจำตัวไม่ซ้ำกันเลย เป็นต้น จะใช้รูปภาพสี่เหลี่ยมผืนผ้าแทน ดังรูปที่ 2.32



รูปที่ 2.32 รูปแสดงเอนทิตีชนิดทั่วไปหรือชนิดแข็ง

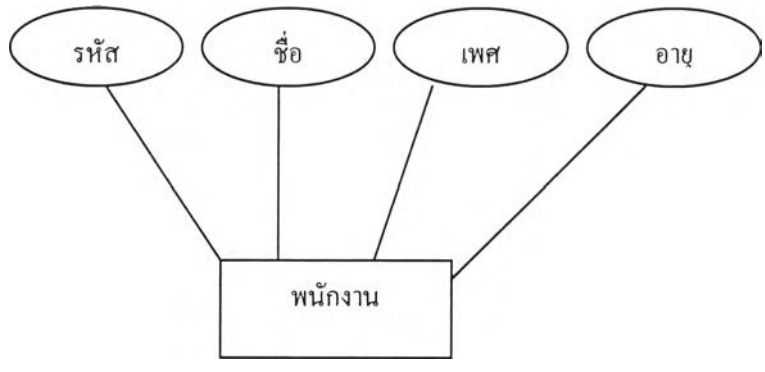
2.6.2.1.2. เอนทิตีชนิดอ่อน (Weak Entities) เป็นเอนทิตีที่ประกอบด้วยสมาชิกที่ไม่สามารถบ่งบอกเอกลักษณ์ของตัวเองได้โดยตรงต้องอาศัยคุณสมบัติใดคุณสมบัติหนึ่งของเอนทิตีทั่วไป มาประกอบกับคุณสมบัติของตัวมันเองจะใช้รูปสี่เหลี่ยมผืนผ้าซ้อนกันสองรูป ดังรูปที่ 2.33



รูปที่ 2.33 รูปแสดงเอนทิตีชนิดอ่อน

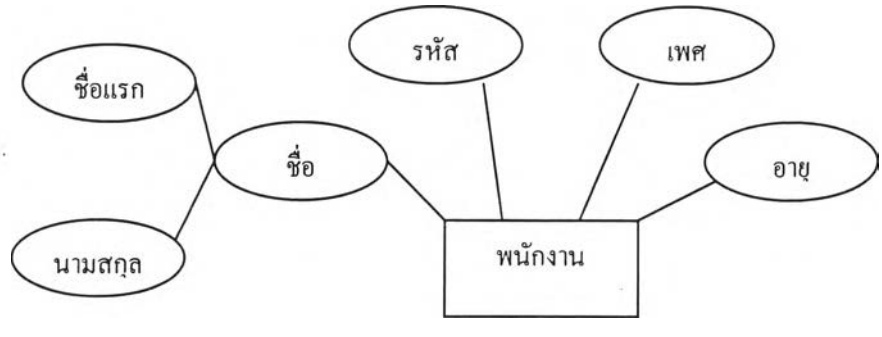
2.6.2.2. ลักษณะประจำ (Attribute) ของ เอนทิตีซึ่งจะนำมาทำความสัมพันธ์ในการออกแบบฐานข้อมูลเช่น หมายเลขพนักงานในบริษัท ชื่อพนักงาน วันเดือนปีเกิด เงินเดือน เป็นต้น คุณสมบัติแบ่งออกได้ 6 ชนิดดังนี้

2.6.2.2.1. ลักษณะประจำอย่างง่าย (Simple Attribute) เป็นคุณสมบัติ ที่มีค่าที่ไม่สามารถแบ่งแยกย่อยต่อไปได้อีก เช่น เพศ เงินเดือน อายุ อำเภอ จังหวัด เป็นต้น ตัวอย่างอธิบายตามรูปที่ 2.34



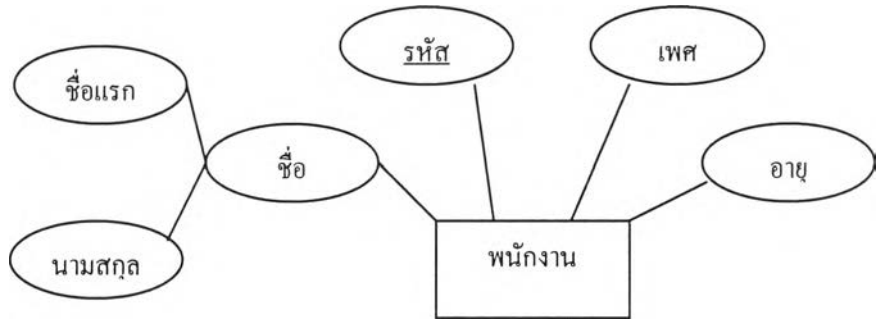
รูปที่ 2.34 แสดงลักษณะประจำอย่างง่าย

2.6.2.2.2. ลักษณะประจำประกอบ (Composite Attribute) เป็นลักษณะประจำที่ค่าภายในสามารถแบ่งแยกเป็น ลักษณะประจำย่อยได้อีก เช่น ชื่ออาจจะแบ่งได้เป็น ชื่อ ชื่อสกุล หรือคำนำหน้า เป็นต้น หรือ ที่อยู่อาจจะแบ่งได้เป็น บ้านเลขที่ หมู่ที่ ถนน ตำบล อำเภอ จังหวัด เป็นต้น สามารถใช้สัญลักษณ์ อธิบายดังรูปที่ 2.35



รูปที่ 2.35 แสดงลักษณะประจำประกอบ

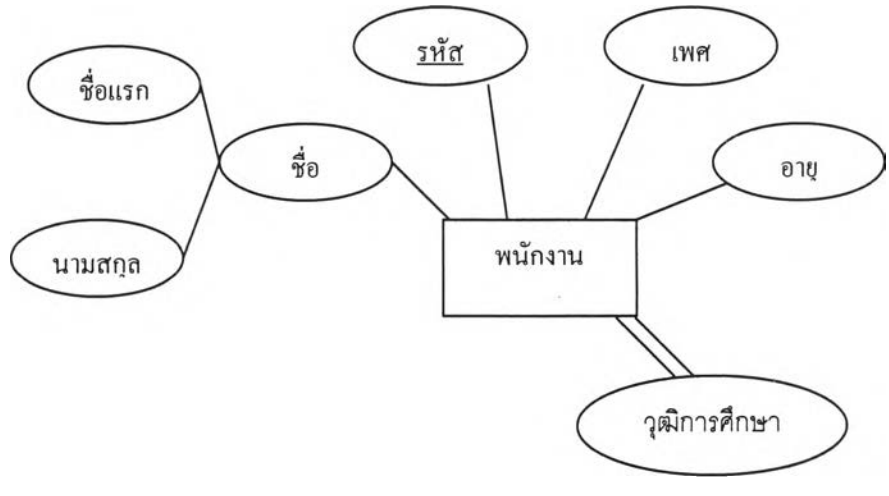
2.6.2.2.3. ลักษณะประจำที่เป็นเอกลักษ์ (Identifier หรือ Key) เป็นลักษณะประจำที่มีค่าไม่ซ้ำกันเลย จะนำมาใช้ในการกำหนดเอกลักษณ์ให้แก่เอนทิตีนั้นๆ เช่นรหัสพนักงาน โดยใช้สัญญลักษณ์เช่นเดียวกับลักษณะประจำอื่นๆ เพียงแต่ขีดเส้นใต้ที่ลักษณะประจำเท่านั้นอธิบายได้ตามรูปที่ 2.36 เป็นต้น



รูปที่ 2.36 แสดงลักษณะประจำที่เป็นเอกลักษณ์

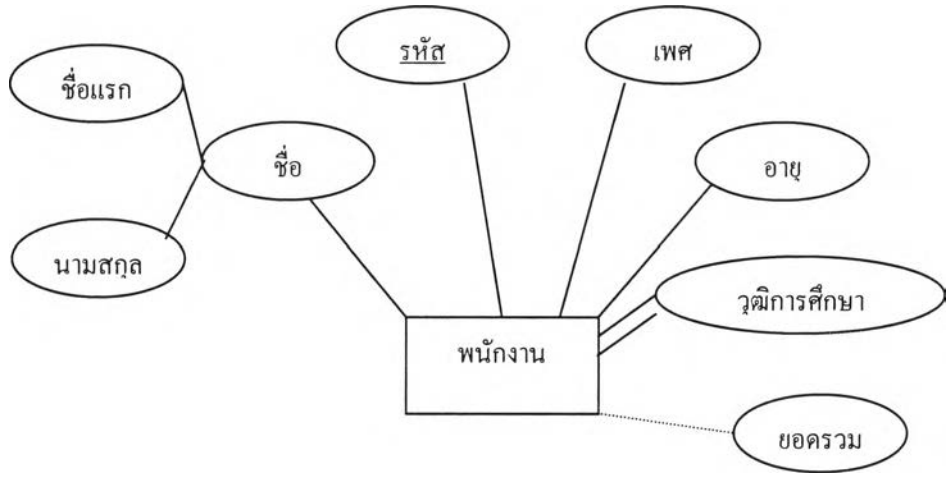
2.6.2.2.4. ลักษณะประจำที่มีค่าข้อมูลค่าเดียว (Single - Valued Attribute) เป็นลักษณะประจำที่มีค่าของข้อมูลเพียงค่าเดียว เช่นลักษณะประจำอายุพนักงาน ลักษณะประจำของเงินเดือนพนักงาน เป็นต้น รูปภาพที่ใช้แสดงแทนจะใช้สัญญลักษณ์ เช่นเดียวกับ ลักษณะประจำอย่างง่าย

2.6.2.2.5. ลักษณะประจำที่มีค่าข้อมูลหลายค่า (Multi - Valued Attribute) เป็นลักษณะประจำที่มีค่าของข้อมูลได้หลายค่า เช่น ลักษณะประจำวุฒิการศึกษา สำหรับรูปภาพจะใช้สัญญลักษณ์ เช่นเดียวกับลักษณะประจำอย่างง่าย แต่ใช้สัญญลักษณ์ เส้น 2 เส้น ดังรูปที่ 2.37



รูปที่ 2.37 แสดงลักษณะประจำที่มีค่าข้อมูลหลายค่า

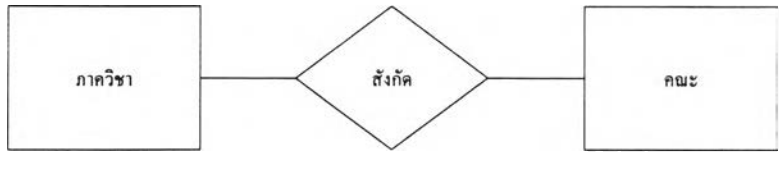
2.6.2.2.6. ลักษณะประจำที่เป็นอนุพันธ์ (Derived Attribute) เป็นลักษณะประจำที่ได้จากการนำเอาค่าของลักษณะประจำอื่น มาคำนวณ ซึ่งค่าของลักษณะประจำประเภทนี้ จะต้องเปลี่ยนแปลงทุกครั้ง เมื่อมีการเปลี่ยนแปลงค่าของลักษณะประจำที่นำมาคำนวณ เช่นเงินสะสมของพนักงานแต่ละคนเพื่อนำไปคำนวณภาษีได้จากผลรวมของเงินเดือนแต่ละเดือนของพนักงานแต่ละคน สัญญลักษณ์ที่ใช้เหมือนกันกับลักษณะประจำอย่างง่าย แต่ใช้เส้นเชื่อมด้วยเส้นประ อธิบายได้ในรูปที่ 2.38



รูปที่ 2.38 แสดงลักษณะประจำที่เป็นอนุพันธ์

2.6.3. ความสัมพันธ์ (Relationship)

เป็นการนำเอาเอนทิตีตั้งแต่ 2 เอนทิตีขึ้นไปทำความสัมพันธ์กันชนิดใดชนิดหนึ่ง เช่นความสัมพันธ์ “สังกัด” ที่ใช้แสดงความสัมพันธ์ภาควิชาสังกัดคณะ เป็นต้น ดังนั้นในการตั้งชื่อให้กับความสัมพันธ์จึงควรจะตั้งชื่อที่แสดงถึงความสัมพันธ์นั้นๆ รูปภาพที่ใช้แสดงความสัมพันธ์จะใช้รูปภาพสี่เหลี่ยมข้าวหลามตัด ภาพที่แสดงความสัมพันธ์จะปรากฏอยู่เดี่ยวๆ ไม่ได้แต่จะอยู่คู่กับเอนทิตีเสมออธิบายได้ตามรูปที่ 2.39



รูปที่ 2.39 แสดงความสัมพันธ์ระหว่าง 2 เอนทิตี

สมาชิกในเอนทิตีที่เกี่ยวข้องกับความสัมพันธ์จะถูกเรียกว่า ผู้มีส่วนร่วม (Participant) ซึ่งจะถูกนำไปใช้ในการกำหนดประเภทของความสัมพันธ์ดังนี้

2.6.3.1. ความสัมพันธ์ชนิดหนึ่งต่อหนึ่ง (One-to-one Relationship) เป็นความสัมพันธ์ที่แต่ละผู้มีส่วนรวมของเอนทิตีหนึ่งจะมีความสัมพันธ์กับอีกผู้มีส่วนร่วมของอีกเอนทิตีหนึ่งเพียงเอนทิตีเดียวดังตัวอย่างในรูปที่ 2.40

รูปที่ 2.40 แสดงความสัมพันธ์ชนิดหนึ่งต่อหนึ่ง

2.6.3.2. ความสัมพันธ์ชนิดหนึ่งต่อกลุ่ม (One-to-Many) เป็นความสัมพันธ์ที่แต่ละผู้มีส่วนร่วมของเอนทิตีหนึ่งมีความสัมพันธ์กับผู้มีส่วนร่วมของอีกเอนทิตีหนึ่งมากกว่า 1 ผู้มีส่วนร่วม เช่นกรณี ภาควิชาสังกัดคณะได้เพียง 1 คณะ แต่คณะสามารถมีภาควิชามาสังกัดได้มากกว่าหนึ่งภาควิชา อธิบายดังรูปที่ 2.41

รูปที่ 2.41 แสดงความสัมพันธ์ชนิดหนึ่งต่อกลุ่ม

2.6.3.3. ความสัมพันธ์ชนิดกลุ่มต่อกลุ่ม (Many-to-Many) เป็นความสัมพันธ์ที่ผู้มีส่วนร่วมมากกว่าหนึ่งของเอนทิตีมีความสัมพันธ์กับผู้มีส่วนร่วมของอีกเอนทิตีหนึ่งมากกว่าหนึ่งผู้มีส่วนร่วม เช่น กรณีผู้ส่งเอกสาร สามารถส่งเอกสารไปให้ผู้รับได้มากกว่าหนึ่งคน และผู้รับสามารถรับเอกสารจากผู้ส่งได้มากกว่าหนึ่งคนเช่นกันอธิบายได้โดยรูปที่ 2.42

รูปที่ 2.42 แสดงความสัมพันธ์ชนิดกลุ่มต่อกลุ่มรายการอ้างอิง