บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาและพัฒนาโปรแกรมสนับสนุนงานบริการการขอใช้ห้องเรียนและห้องปฏิบัติการของคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร ซึ่งผู้ศึกษา ค้นคว้าจากเอกสารและงานวิจัยที่เกี่ยวข้อง โดยเสนอตามลำดับดังนี้

2.1 การบริการการขอใช้ห้องเรียนและห้องปฏิบัติการ

2.2 วิธีการคิดค่าใช้จ่าย

2.3 แนวคิดการพัฒนาระบบ

2.4 แนวคิดเชิงวัตถุ (Object-Oriented concept)

2.4.1 แบบจำลองเชิงวัตถุด้วยยูเอ็มแอล

2.4.2 ภาษาเชิงวัตถุ

2.5 แนวคิดเกี่ยวกับ Open source

2.5.1 MySQL โปรแกรมในการจัดการฐานข้อมูล

2.5.2 PHP ภาษาในการพัฒนาระบบ

2.6 งานวิจัยที่เกี่ยวข้อง

2.1 การบริการการขอใช้ห้องเรียนและห้องปฏิบัติการ

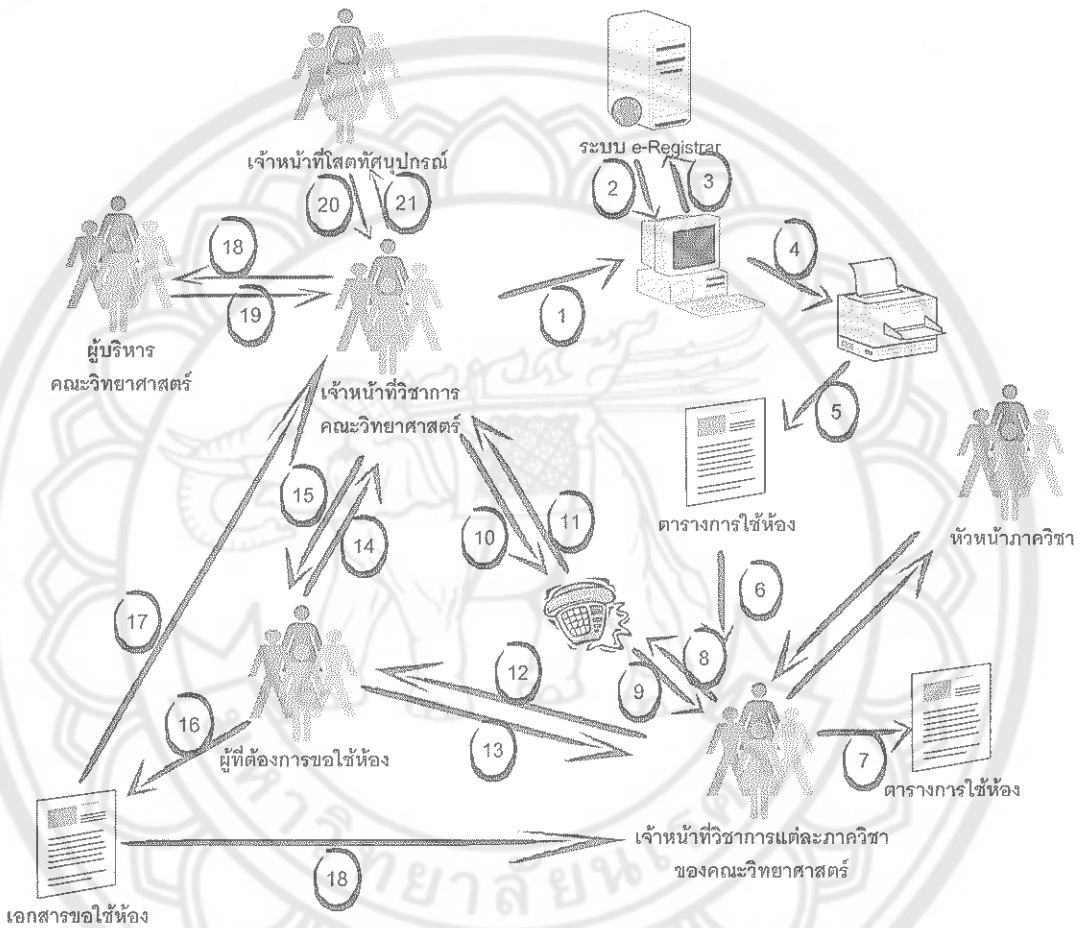
คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร เป็นคณะที่มีห้องเรียนจำนวนมาก พร้อมที่จะให้บริการแก่หน่วยงานภายในคณะและหน่วยงานภายในมหาวิทยาลัยเป็นหลัก เพื่อให้เป็นการใช้ทรัพยากรอย่างคุ้มค่า

กระบวนการขอใช้ห้องเรียนและห้องปฏิบัติการก็มีขั้นตอนในการขอใช้ห้องเรียนและห้องปฏิบัติการดังแสดงในภาพ 1ซึ่งมีรายละเอียดของกระบวนการดังนี้

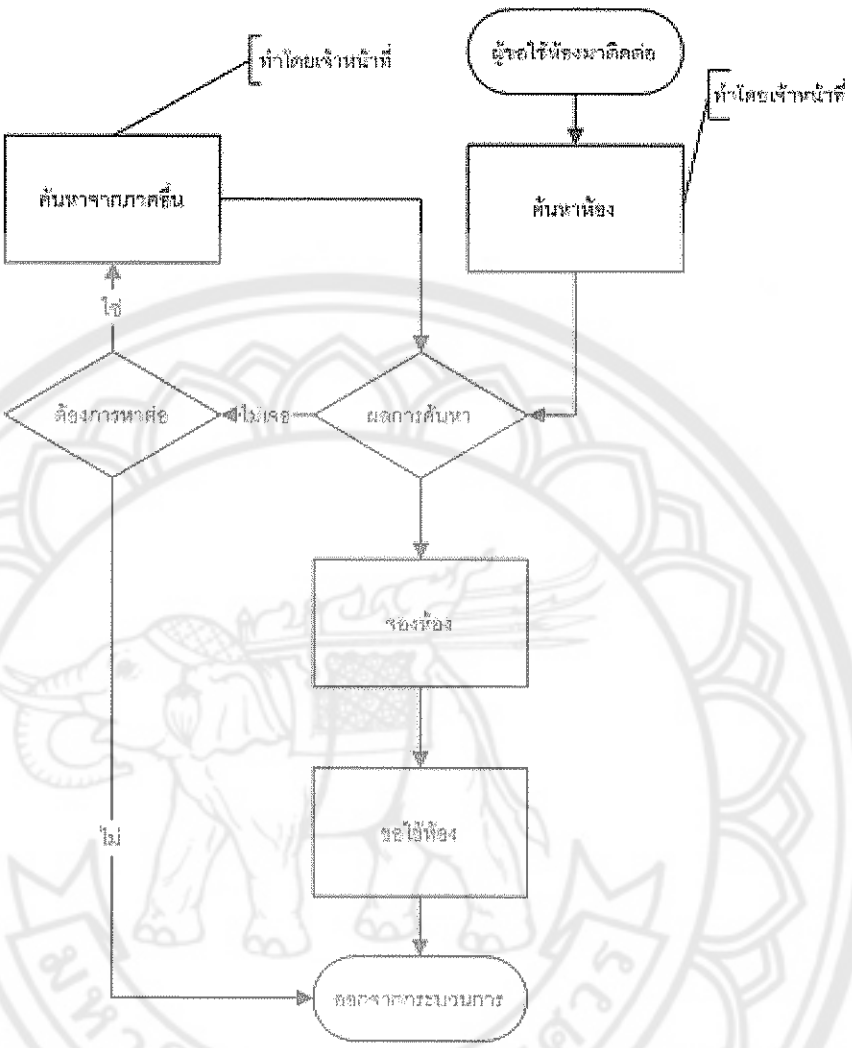
1. เจ้าหน้าที่วิชาการคณะวิทยาศาสตร์เข้าสู่ระบบ e-Registrar (1) (2) (3)
2. พิมพ์ตารางการใช้ห้อง (4) (5)
3. นำเอกสารที่พิมพ์มาแจกจ่ายให้แต่ละหน่วยงานย่อย (6)
4. เมื่อมีผู้ขอใช้ห้องมาติดต่อขอใช้ห้องมาติดต่อขอใช้ก็ต้องมาติดต่อที่เจ้าหน้าที่เพื่อค้นหาห้อง (13) (14) ในกรณีที่ไม่มีห้องจะต้องไปหาที่หน่วยงานอื่น (10) (9) และแจ้งห้องที่สามารถหาให้ได้ (15) (12) ผู้ที่ต้องการใช้ห้องก็ต้องมาทำเอกสาร (16) เพื่อยื่น

ขอใช้ห้องแล้วนำไปยื่นกับเจ้าหน้าที่ (17) (18) เจ้าหน้าที่จำนำเอกสารไปยื่นให้กับผู้บริหารเพื่อรอการอนุมัติ

5. เมื่อผู้บริหารต้องการรายงานการใช้ห้องจะต้องให้เจ้าหน้าที่รวบรวมจากเอกสารตารางการใช้ห้อง



ภาพ 1 ขั้นตอนของการขอใช้ห้อง



ภาพ 2 Flow Chart ของระบบเก่า

จากภาพจะเห็นได้ว่ามีขั้นตอนค่อนข้างมากซึ่งสามารถแยกเป็นงานต่างๆ ได้ดังนี้

2.1.1 การค้นหาห้อง

2.1.2 การจองห้อง

2.1.3 การขอใช้ห้อง

2.1.4 การอนุมัติใช้ห้อง

2.1.1 การค้นหาห้อง

กรณีบุคคลภายในคณะต้องการหาห้องอยู่ในการดูแลของภาควิชา

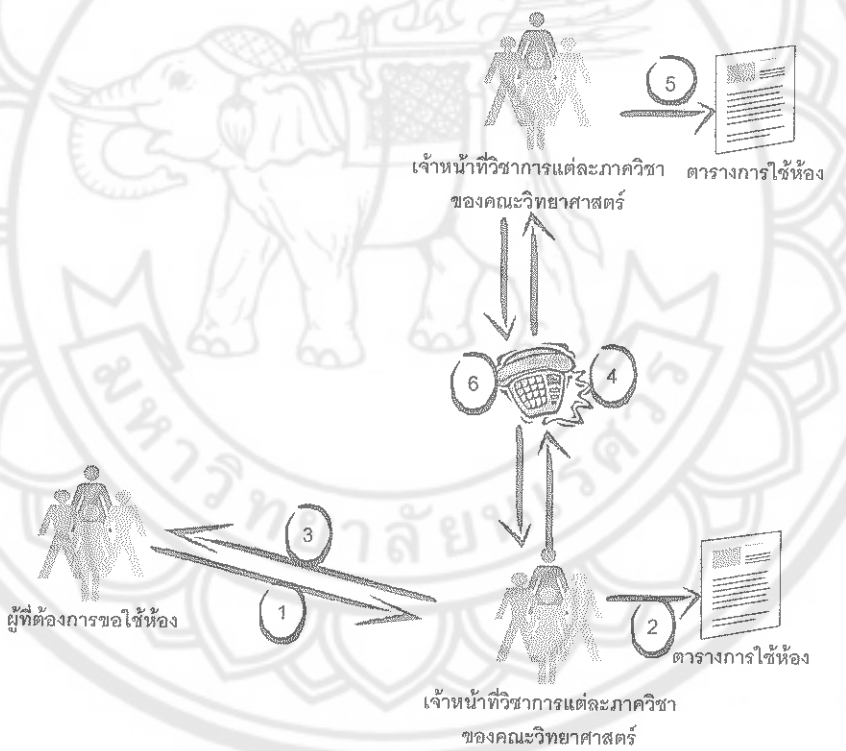
- ผู้ที่ต้องการขอใช้ห้องต้องมาติดต่อกับเจ้าหน้าที่วิชาการแต่ละภาควิชา
- เจ้าหน้าที่วิชาการแต่ละภาควิชาจะค้นหาห้องจากเอกสารให้
- เจ้าหน้าที่วิชาการแต่ละภาควิชาจะต้องผลว่ามีห้องว่างหรือไม่

กรณีบุคคลภายในคณะต้องการหาห้องอยู่ในการดูแลของสำนักงานเลขานุการคณะ

- ผู้ที่ต้องการขอใช้ห้องต้องมาติดต่อกับเจ้าหน้าที่วิชาการคณะ
- เจ้าหน้าที่วิชาการคณะจะค้นหาห้องจากเอกสารให้
- เจ้าหน้าที่วิชาการคณะจะต้องตรวจสอบและแจ้งผลว่ามีห้องว่างหรือไม่

กรณีบุคคลภายนอกคณะต้องการหาห้องว่างภายในคณะ

- ผู้ที่ต้องการขอใช้ห้องต้องมาติดต่อกับเจ้าหน้าที่วิชาการของคณะวิทยาศาสตร์
- เจ้าหน้าที่วิชาการคณะจะค้นหาห้องจากเอกสารให้
- กรณีที่ห้องที่คณะไม่ว่างเจ้าหน้าที่วิชาการคณะจะสอบถามห้องที่เจ้าหน้าที่วิชาการแต่ละภาค
- เจ้าหน้าที่วิชาการของคณะวิทยาศาสตร์จะต้องตรวจสอบและแจ้งผลว่ามีห้องว่างหรือไม่



ภาพ 3 ขั้นตอนของการค้นหาห้อง

2.1.2 การจองห้อง

— เมื่อได้ห้องตามต้องการแล้วจะทำการจองห้องไว้โดยจดบันทึกลงในตาราง

2.1.3 การขอใช้ห้อง

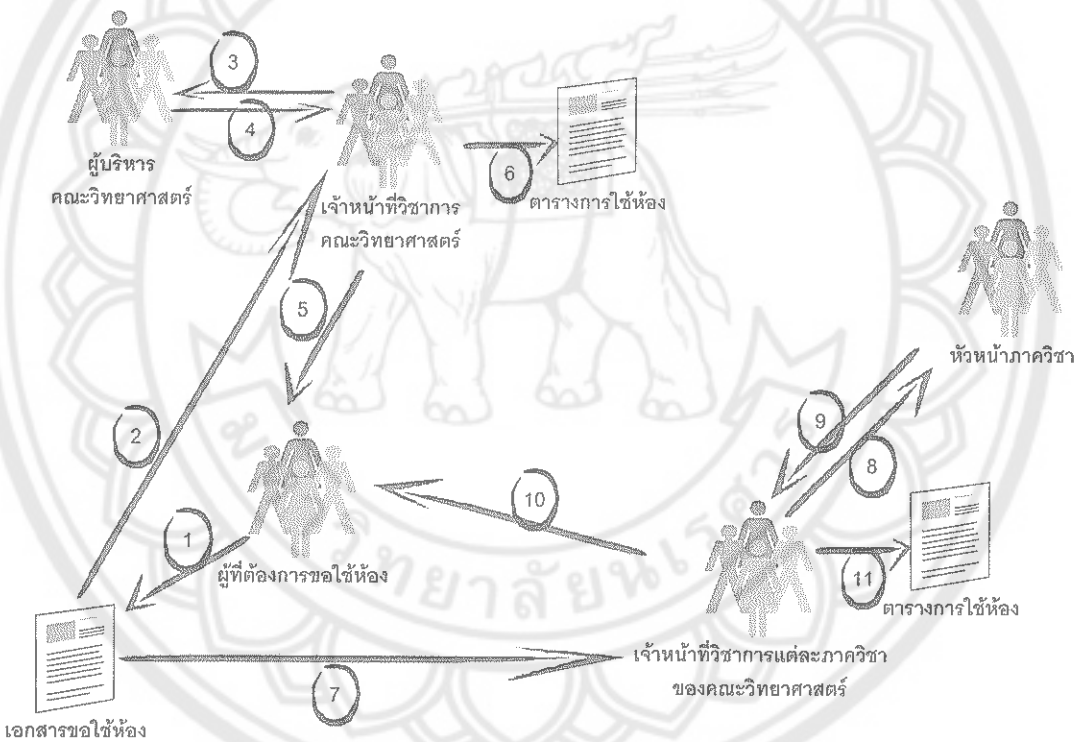
เมื่อทำการจองห้องแล้วจะต้องขออนุมัติใช้ห้อง

กรณีบุคคลภายในคณะ

- ทำหนังสือถึงหัวหน้าหน่วยงานที่ใช้ห้องได้ทันทีโดยผ่านเจ้าหน้าที่ (1) (7)
- เจ้าหน้าที่ตรวจสอบ (11)
- รอผลการอนุมัติ
- เจ้าหน้าที่แจ้งผล (10)

กรณีบุคคลภายนอกคณะ

- ทำหนังสือถึงคณบดีคณะวิทยาศาสตร์โดยผ่านเจ้าหน้าที่ (1) (2)
- เจ้าหน้าที่ตรวจสอบ (6)
- รอผลการอนุมัติ
- เจ้าหน้าที่แจ้งผล (5)



ภาพ 4 ขั้นตอนการขอใช้ห้องและการอนุมัติใช้ห้อง

2.1.4 การอนุมัติใช้ห้อง

หัวหน้าแต่ละหน่วยงานและคณบดีคณะวิทยาศาสตร์จะเป็นผู้อนุมัติการใช้ห้องทุกครั้งเมื่อมีการขอใช้ห้อง

การใช้ห้องเรียนและห้องปฏิบัตินั้นมีค่าใช้จ่ายต่างๆ เกิดขึ้นเช่น ค่าไฟฟ้า เครื่องค่าปรับอากาศ ค่าเครื่องคอมพิวเตอร์ และค่าแม่บ้าน ในการขอใช้ห้องต่างๆ เพื่อให้การเรียนการสอนนั้น

ถ้าเป็นบุคคลภายในคณะ ทางคณะจะบริการโดยไม่คิดค่าใช่จ่าย แต่ถ้าเป็นบุคคลภายนอกคณะจะคิดค่าใช้จ่าย โดยไม่หวังผลกำไรดังจะได้กล่าวต่อไป

2.2 การคิดค่าใช้จ่าย

ตามประกาศมหาวิทยาลัยนเรศวร เรื่อง กำหนดอัตราค่าบริการชุดคอมพิวเตอร์ ของคณะวิทยาศาสตร์ ได้กำหนดอัตราการค่าบริการคอมพิวเตอร์ ของคณะวิทยาศาสตร์ ดังนี้

- อัตราการค่าบริการคอมพิวเตอร์หน่วยงานภายในมหาวิทยาลัยคิดอัตราการค่าบริการคอมพิวเตอร์เป็นรายชั่วโมงๆ ละ 5 บาท/ชุด
- อัตราการค่าบริการคอมพิวเตอร์หน่วยงานภายนอกมหาวิทยาลัยคิดอัตราการค่าบริการคอมพิวเตอร์เป็นรายชั่วโมงๆ ละ 10 บาท/ชุด

หมายเหตุ

- อัตราการค่าบริการคอมพิวเตอร์จะคิดอัตราค่าบริการตามจำนวนคอมพิวเตอร์ที่ใช้งานจริง
- ค่าบริการคอมพิวเตอร์นั้นจะไม่จะไม่รวมค่าตอบแทนผู้ปฏิบัติงานและค่าบิการการใช้ห้องเรียนคณะวิทยาศาสตร์

ตาราง 2 อัตราค่าบริการห้องเรียน

ขนาดห้องเรียน (ที่นั่ง)	เจ้าหน้าที่โสตฯ (คน) *	แม่บ้าน (คน) *	ชั่วโมงละ (บาท)	หมายเหตุ
100-120	1	1	650	เครื่องปรับอากาศ
150-400	1	1-2	1,100	เครื่องปรับอากาศ

จากตาราง 2 การกำหนดจำนวนเจ้าหน้าที่โสตฯ และแม่บ้านไม่ได้มีการกำหนดตายตัวแต่ขึ้นอยู่กับความเหมาะสมกับงาน

หมายเหตุ ถ้าเป็นหน่วยงานภายในคณะขอใช้ห้องจะไม่คิดค่าใช้จ่ายในการบริการห้อง แต่จะต้องรับผิดชอบเรื่องค่าปฏิบัติงานของเจ้าหน้าที่ที่เกี่ยวข้อง

ในการศึกษาครั้งนี้ ผู้ศึกษาจะใช้อัตราตามประกาศนี้

2.3 แนวคิดการพัฒนาระบบงาน

จุดกำเนิดของระบบงานโดยปกติจะกำเนิดขึ้นจากผู้ใช้ระบบ เนื่องจากผู้ใช้ระบบเป็นผู้ใกล้ชิดกับกิจกรรมของธุรกิจมากที่สุด ดังนั้นกิจกรรมทางธุรกิจได้ดำเนินไปอย่างต่อเนื่องนั้น ความต้องการที่จะพัฒนาปรับปรุงกิจการต่างๆ ย่อมเกิดขึ้น นักวิเคราะห์ระบบจึงเริ่มเข้ามามีบทบาทในการพัฒนาปรับปรุงแก้ไขระบบงาน

James Wetherbe (2527) โดยใช้ชื่อว่า "System Analysis and Design: Traditional, Structured and Advanced Concepts and Techniques."โดยให้แนวความคิดในการแจกแจงกลุ่มของปัญหาออกเป็น 6 หัวข้อตามความต้องการของผู้ใช้ ซึ่งแทนด้วยอักษร 6 ตัวคือ PIECES อ่านว่า "พีซ-เซส" โดยมีรายละเอียดดังนี้คือ

1) Performance หมายถึงความต้องการที่จะให้มีการปรับปรุงทางด้านการปฏิบัติงาน

2) Information หมายถึงความต้องการที่จะให้มีการปรับปรุงและควบคุมทางด้านข้อมูล

3) Economics หมายถึงความต้องการที่จะให้มีการปรับปรุงและควบคุมทางด้านต้นทุน

4) Control หมายถึงความต้องการที่จะให้มีการปรับปรุงระบบงานข้อมูลเพื่อให้มีการควบคุมและระบบรักษาความปลอดภัยที่ดียิ่งขึ้น

5) Efficiency หมายถึงความต้องการที่จะให้มีการปรับปรุงประสิทธิภาพของคนและเครื่องจักร

6) Service หมายถึงความต้องการที่จะให้มีการปรับปรุงการบริการต่างๆ ให้ดีขึ้น เช่น การบริการลูกค้าหรือการให้บริการต่อพนักงานภายในธุรกิจเองเป็นต้น

ในแต่ละโครงการของระบบงานข้อมูลนั้น จะมีลักษณะที่จะตอบสนองความต้องการที่ได้ระบุอยู่ในพีซเซสอันใดอันหนึ่งหรือมากกว่าหนึ่งก็ได้ ดังนั้นพีซเซสจึงมีความสำคัญต่อนักวิเคราะห์ระบบในการใช้ เพื่อพิจารณาถึงปัญหาและความต้องการของผู้ใช้อย่างมีหลักเกณฑ์

วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC)

อำไพ วรรณสินธุ์ (ม.ป.ป.) ได้กล่าวว่า ระบบสารสนเทศทั้งหลายมีวงจรชีวิตที่เหมือนกัน ตั้งแต่เกิดจนตายวงจรนี้จะเป็นขั้นตอนที่เป็นลำดับตั้งแต่ต้นจนเสร็จเรียบร้อย เป็นระบบที่ใช้งานได้ ซึ่งนักวิเคราะห์ระบบต้องทำความเข้าใจให้ดีว่าในแต่ละขั้นตอนจะต้องทำอะไร และทำอย่างไร ขั้นตอนการพัฒนาระบบมีอยู่ด้วยกัน 7 ขั้น ด้วยกัน คือ 1) เข้าใจปัญหา (Problem Recognition), 2) ศึกษาความเป็นไปได้ (Feasibility Study), 3) วิเคราะห์ (Analysis), 4) ออกแบบ (Design),

5) สร้างหรือพัฒนาระบบ (Construction), 6) การปรับเปลี่ยน (Conversion), 7) บำรุงรักษา (Maintenance)

ขั้นที่ 1 : เข้าใจปัญหา *(Problem Recognition)*

การที่จะแก้ไขระบบเดิมที่มีอยู่แล้วไม่ใช่เรื่องที่ง่ายนัก หรือแม้แต่การสร้างระบบใหม่ ดังนั้น ควรจะมีการศึกษาเสียก่อนว่าความต้องการของผู้ใช้ระบบ (เจ้าหน้าที่,ผู้ขอใช้ และผู้บริหาร) มีความต้องการให้ระบบใหม่ช่วยแก้ปัญหาที่เกิดขึ้นอย่างไร ควรมีฟังก์ชั่นอะไร ขั้นตอนควรเป็นอย่างไร ในขั้นนี้คือทำความเข้าใจขั้นตอนการทำงานและปัญหาของระบบงานเดิม

ขั้นตอนที่ 2 : ศึกษาความเป็นไปได้ *(Feasibility Study)*

จุดประสงค์ของการศึกษาความเป็นไปได้ก็คือ การกำหนดว่าปัญหาคืออะไรและตัดสินใจว่าการพัฒนาสร้างระบบสารสนเทศ หรือการแก้ไขระบบสารสนเทศเดิมมีความเป็นไปได้หรือไม่ โดยเสียค่าใช้จ่ายและเวลาน้อยที่สุด และได้ผลเป็นที่น่าพอใจ

ปัญหาต่อไปคือ นักวิเคราะห์ระบบจะต้องกำหนดให้ได้ว่าการแก้ไขปัญหาดังกล่าวมีความเป็นไปได้ทางเทคนิคและบุคลากร ปัญหาทางเทคนิคก็จะเกี่ยวข้องกับเรื่องคอมพิวเตอร์ และเครื่องมือเก่าๆ ถ้ามี รวมทั้งเครื่องคอมพิวเตอร์ซอฟต์แวร์ด้วย ตัวอย่างคือ คอมพิวเตอร์ที่ใช้อยู่ในบริษัทเพียงพอหรือไม่ คอมพิวเตอร์อาจจะมีเนื้อที่ของฮาร์ดดิสก์ไม่เพียงพอ รวมทั้งซอฟต์แวร์ ว่าอาจจะต้องซื้อใหม่ หรือพัฒนาขึ้นใหม่ เป็นต้น ความเป็นไปได้ทางด้านบุคลากร คือ บริษัทมีบุคคลที่เหมาะสมที่จะพัฒนาและติดตั้งระบบเพียงพอหรือไม่ ถ้าไม่มีจะหาได้หรือไม่ จากที่ใด เป็นต้น นอกจากนั้นควรจะให้ความสนใจว่าผู้ใช้ระบบมีความคิดเห็นอย่างไรกับการเปลี่ยนแปลง รวมทั้งความเห็นของผู้บริหารด้วย (อำไพ วรรณสินธุ์ , ม.ป.ป.)

สุดท้ายนักวิเคราะห์ระบบต้องวิเคราะห์ได้ว่า ความเป็นไปได้เรื่องค่าใช้จ่าย รวมทั้งเวลาที่ใช้ในการพัฒนาระบบ และที่สำคัญคือ ผลประโยชน์ที่จะได้รับ เรื่องเวลาเป็นสิ่งสำคัญ เช่น การเปลี่ยนแปลงระบบเพื่อรองรับผู้ขายให้ได้มากกว่า 1,000 บริษัทนั้น ควรใช้เวลาไม่เกิน 1 ปี ตั้งแต่เริ่มต้นจนใช้งานได้ ค่าใช้จ่ายเริ่มตั้งแต่พัฒนาจนถึงใช้งานได้จริงได้แก่ เงินเดือน เครื่องมืออุปกรณ์ ต่างๆ เป็นต้น พูดถึงเรื่องผลประโยชน์ที่ได้รับอาจมองเห็นได้ไม่ง่ายนัก แต่นักวิเคราะห์ระบบควรมองและตีออกมาในรูปเงินให้ได้ เช่น เมื่อนำระบบใหม่เข้ามาใช้อาจจะทำให้ ค่าใช้จ่ายบุคลากรลดลง หรือกำไรเพิ่มมากขึ้น เช่น ทำให้ยอดขายเพิ่มมากขึ้น เนื่องจากผู้บริหารมีข้อมูลพร้อมที่จะช่วยในการตัดสินใจที่ดีขึ้น

การคาดคะเนทั้งหลายเป็นไปอย่างหยาบๆ เราไม่สามารถหาตัวเลขที่แน่นอนตายตัวได้ เนื่องจากทั้งหมดยังไม่ได้เกิดขึ้นจริง หลังจากเตรียมตัวเลขเรียบร้อยแล้ว นักวิเคราะห์ระบบก็นำตัวเลข ค่าใช้จ่ายและผลประโยชน์ (Cost-Benefit) มาเปรียบเทียบกัน

ขั้นตอนที่ 3 การวิเคราะห์ (Analysis)

เริ่มเข้าสู่การวิเคราะห์ระบบ การวิเคราะห์ระบบเริ่มตั้งแต่การศึกษาระบบการทำงานของธุรกิจนั้น ในกรณีที่ระบบเราศึกษานั้นเป็นระบบสารสนเทศอยู่แล้วจะต้องศึกษาว่าทำงานอย่างไร เพราะเป็นการยากที่จะออกแบบระบบใหม่โดยที่ไม่ทราบว่าระบบเดิมทำงานอย่างไร หรือธุรกิจดำเนินการอย่างไร หลังจากนั้นกำหนดความต้องการของระบบใหม่ ซึ่งนักวิเคราะห์ระบบจะต้องใช้เทคนิคในการเก็บข้อมูล (Fact-Gathering Techniques) ได้แก่ ศึกษาเอกสารที่มีอยู่ ตรวจสอบวิธีการทำงานในปัจจุบัน สัมภาษณ์ผู้ใช้และผู้จัดการที่มีส่วนเกี่ยวข้องกับระบบ เอกสารที่มีอยู่ได้แก่ คู่มือการใช้งาน แผนผังใช้งานขององค์กร รายงานต่างๆ ที่หมุนเวียนในระบบการศึกษาวิธีการทำงานในปัจจุบันจะทำให้นักวิเคราะห์ระบบรู้ว่าระบบจริงๆ ทำงานอย่างไร ซึ่งบางครั้งค้นพบข้อผิดพลาดได้ ตัวอย่าง เช่น เมื่อบริษัทได้รับใบเรียกเก็บเงินจะมีขั้นตอนอย่างไรในการจ่ายเงิน ขั้นตอนที่เสมียนป้อนใบเรียกเก็บเงินอย่างไร เฝ้าสังเกตการทำงานของผู้เกี่ยวข้อง เพื่อให้เข้าใจและเห็นจริงๆ ว่าขั้นตอนการทำงานเป็นอย่างไร ซึ่งจะทำให้นักวิเคราะห์ระบบค้นพบจุดสำคัญของระบบว่าอยู่ที่ใด (อำไพ วรรณสินธุ์, ม.ป.ป.)

การสัมภาษณ์เป็นศิลปะอย่างหนึ่งที่นักวิเคราะห์ระบบควรจะต้องมีเพื่อเข้ากับผู้ใช้ได้ง่ายและสามารถดึงสิ่งที่ต้องการจากผู้ใช้ได้ เพราะว่าความต้องการของระบบคือ สิ่งสำคัญที่จะใช้ในการออกแบบต่อไป ถ้าเราสามารถกำหนดความต้องการได้ถูกต้อง การพัฒนาระบบในขั้นตอนต่อไปก็จะง่ายขึ้น เมื่อเก็บรวบรวมข้อมูลแล้วจะนำมาเขียนรวมเป็นรายงานการทำงานของระบบซึ่งควรแสดงหรือเขียนออกมาเป็นรูปแทนที่จะร่ายยาวออกมาเป็นตัวหนังสือ การแสดงแผนภาพจะทำให้เราเข้าใจได้ดีและง่ายขึ้น หลังจากนั้นนักวิเคราะห์ระบบ อาจจะนำข้อมูลที่รวบรวมได้นำมาเขียนเป็น "แบบทดลอง" (Prototype) หรือตัวต้นแบบ แบบทดลองจะเขียนขึ้นด้วยภาษาคอมพิวเตอร์ต่างๆ และที่ช่วยให้ง่ายขึ้นได้แก่ ภาษายุคที่ 4 (Fourth Generation Language) เป็นการสร้างโปรแกรมคอมพิวเตอร์ขึ้นมาเพื่อใช้งานตามที่เราต้องการได้ ดังนั้นแบบทดลองจึงช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้

เมื่อจบขั้นตอนการวิเคราะห์แล้ว นักวิเคราะห์ระบบจะต้องเขียนรายงานสรุปออกมาเป็นข้อมูลเฉพาะของปัญหา (Problem Specification) ซึ่งมีรายละเอียดดังนี้

รายละเอียดของระบบเดิม ซึ่งควรจะเขียนมาเป็นรูปภาพแสดงการทำงานของระบบ พร้อมคำบรรยาย กำหนดความต้องการของระบบใหม่รวมทั้งรูปภาพแสดงการทำงานพร้อมคำบรรยาย ข้อมูลและไฟล์ที่จำเป็น คำอธิบายวิธีการทำงาน และสิ่งที่จะต้องแก้ไข

ขั้นตอนที่4 : การออกแบบ (Design)

ในระยะแรกของการออกแบบ นักวิเคราะห์ระบบจะนำการตัดสินใจของฝ่ายบริหารที่ได้จากขั้นตอนการวิเคราะห์การเลือกซื้อคอมพิวเตอร์ ฮาร์ดแวร์และซอฟต์แวร์ด้วย (ถ้ามีหรือเป็นไปได้) หลังจากนั้นนักวิเคราะห์ระบบจะนำแผนภาพต่างๆ ที่เขียนขึ้นในขั้นตอนการวิเคราะห์มาแปลงเป็นแผนภาพลำดับขั้น (แบบต้นไม้) เพื่อให้มองเห็นภาพลักษณ์ที่แน่นอนของโปรแกรมว่ามีความสัมพันธ์กันอย่างไร และโปรแกรมอะไรบ้างที่จะต้องเขียนในระบบ หลังจากนั้นก็เริ่มตัดสินใจว่าควรจะจัดโครงสร้างจากโปรแกรมอย่างไร การเชื่อมระหว่างโปรแกรมควรจะทำอย่างไร ในขั้นตอนการวิเคราะห์นักวิเคราะห์ระบบต้องหาว่า "จะต้องทำอะไร (What) " แต่ในขั้นตอนการออกแบบต้องรู้ว่า" จะต้องทำอย่างไร (How)"

ในการออกแบบโปรแกรมต้องคำนึงถึงความปลอดภัย (Security) ของระบบด้วย เพื่อป้องกันการผิดพลาดที่อาจจะเกิดขึ้น เช่น "รหัส" สำหรับผู้ใช้ที่มีสิทธิ์สำรองไฟล์ข้อมูลทั้งหมด เป็นต้น

นักวิเคราะห์ระบบจะต้องออกแบบฟอร์มสำหรับข้อมูลขาเข้า (Input Format) ออกแบบรายงาน (Report Format) และการแสดงผลบนจอภาพ (Screen Format) หลักการการออกแบบฟอร์มข้อมูลขาเข้าคือ ง่ายต่อการใช้งาน และป้องกันข้อผิดพลาดที่อาจจะเกิดขึ้น

ถัดมาระบบจะต้องออกแบบวิธีการใช้งาน เช่น กำหนดว่าการป้อนข้อมูลจะต้องทำอย่างไร จำนวนบุคลากรที่ต้องการในหน้าที่ต่างๆ แต่ถ้านักวิเคราะห์ระบบตัดสินใจว่าการซื้อซอฟต์แวร์ดีกว่าการเขียนโปรแกรม ขั้นตอนการออกแบบก็ไม่จำเป็นเลย เพราะสามารถนำซอฟต์แวร์สำเร็จรูปมาใช้งานได้ทันที สิ่งที่นักวิเคราะห์ระบบออกแบบมาทั้งหมดในขั้นตอนที่กล่าวมาทั้งหมดจะนำมาเขียนรวมเป็นเอกสารชุดหนึ่งเรียกว่า "ข้อมูลเฉพาะของการออกแบบระบบ " (System Design Specification) เมื่อสำเร็จแล้วโปรแกรมเมอร์สามารถใช้เป็นแบบในการเขียนโปรแกรมได้ทันที**สำคัญก่อนที่จะส่งถึงมือโปรแกรมเมอร์เราควรจะตรวจสอบกับผู้ใช้ว่าพอใจหรือไม่ และตรวจสอบกับทุกคนในทีมว่าถูกต้องสมบูรณ์หรือไม่ และแน่นอนที่สุดต้องส่งให้ฝ่ายบริหารเพื่อตัดสินใจว่าจะดำเนินการ ต่อไปหรือไม่ ถ้าอนุมัติก็ผ่านเข้าสู่ขั้นตอนการสร้างหรือพัฒนาระบบ** (Construction) (อำไพ วรรณสินธุ์, ม.ป.ป.)

ในการศึกษาครั้งนี้ผู้ศึกษาได้เลือกใช้ UML เป็นเครื่องมือช่วยในการออกแบบและให้สอดคล้องกับการเขียนโปรแกรมเชิงวัตถุ ดังจะได้กล่าวต่อไป

ขั้นตอนที่ 5 : การพัฒนาระบบ (Construction)

ในขั้นตอนนี้โปรแกรมเมอร์จะเริ่มเขียนและทดสอบโปรแกรมว่า ทำงานถูกต้องหรือไม่ ต้องมีการทดสอบกับข้อมูลจริงที่เลือกแล้ว ถ้าทุกอย่างเรียบร้อย เราจะได้โปรแกรมที่พร้อมที่จะนำไปใช้งานจริงต่อไป หลังจากนั้นต้องเตรียมคู่มือการใช้และการฝึกอบรมผู้ใช้งานจริงของระบบ

ระยะแรกในขั้นตอนนี้นักวิเคราะห์ระบบต้องเตรียมสถานที่สำหรับเครื่องคอมพิวเตอร์แล้วจะต้องตรวจสอบว่าคอมพิวเตอร์ทำงานเรียบร้อยดี

โปรแกรมเมอร์เขียนโปรแกรมตามข้อมูลที่ได้จากเอกสารข้อมูลเฉพาะของการออกแบบ (Design Specification) ปกติแล้วนักวิเคราะห์ระบบไม่มีหน้าที่เกี่ยวข้องในการเขียนโปรแกรม แต่ถ้าโปรแกรมเมอร์คิดว่าการเขียนอย่างอื่นดีกว่าจะต้องปรึกษานักวิเคราะห์ระบบเสียก่อน เพื่อที่ว่านักวิเคราะห์จะบอกได้ว่าโปรแกรมที่จะแก้ไขนั้นมีผลกระทบกับระบบทั้งหมดหรือไม่ โปรแกรมเมอร์เขียนเสร็จแล้วต้องมีการทบทวนกับนักวิเคราะห์ระบบและผู้ใช้งาน เพื่อค้นหาข้อผิดพลาด วิธีการนี้เรียกว่า "Structure Walkthrough " การทดสอบโปรแกรมจะต้องทดสอบกับข้อมูลที่เลือกแล้วชุดหนึ่ง ซึ่งอาจจะเลือกโดยผู้ใช้ การทดสอบเป็นหน้าที่ของโปรแกรมเมอร์ แต่นักวิเคราะห์ระบบต้องแน่ใจว่า โปรแกรมทั้งหมดจะต้องไม่มีข้อผิดพลาด (อำไพ วรรณสินธุ์, ม.ป.ป.)

ในการศึกษาครั้งนี้ผู้ศึกษาได้เลือกใช้ PHP ในการเขียนโปรแกรม และ MySQL มาใช้เป็นฐานข้อมูล

ขั้นตอนที่ 6 : การปรับเปลี่ยน (Construction)

ขั้นตอนนี้บริษัทนำระบบใหม่มาใช้แทนของเก่าภายใต้การดูแลของนักวิเคราะห์ระบบ การป้อนข้อมูลต้องทำให้เรียบร้อย และในที่สุดบริษัทเริ่มต้นใช้งานระบบใหม่นี้ได้

การนำระบบเข้ามาควรจะทำอย่างค่อยเป็นค่อยไปทีละน้อย ที่ดีที่สุดคือ ใช้ระบบใหม่ควบคู่ไปกับระบบเก่าไปสักระยะหนึ่ง โดยใช้ข้อมูลชุดเดียวกันแล้วเปรียบเทียบผลลัพธ์ว่าตรงกันหรือไม่ ถ้าเรียบร้อยก็เอาระบบเก่าออกได้ แล้วใช้ระบบใหม่ต่อไป

ขั้นตอนที่ 7 : บำรุงรักษา (Maintenance)

การบำรุงรักษาได้แก่ การแก้ไขโปรแกรมหลังจากการใช้งานแล้ว สาเหตุที่ต้องแก้ไขโปรแกรมหลังจากใช้งานแล้ว สาเหตุที่ต้องแก้ไขระบบส่วนใหญ่มี 2 ข้อ คือ

1. มีปัญหาในโปรแกรม (Bug)
2. การดำเนินงานในองค์กรหรือธุรกิจเปลี่ยนไป

จากสถิติของระบบที่พัฒนาแล้วทั้งหมดประมาณ 40% ของค่าใช้จ่ายในการแก้ไขโปรแกรม เนื่องจากมี "Bug" ดังนั้นนักวิเคราะห์ระบบควรให้ความสำคัญกับการบำรุงรักษา ซึ่งปกติจะคิดว่าไม่มีความสำคัญมากนัก

เมื่อธุรกิจขยายตัวมากขึ้น ความต้องการของระบบอาจจะเพิ่มมากขึ้น เช่น ต้องการรายงานเพิ่มขึ้น ระบบที่ดีควรจะแก้ไขเพิ่มเติมสิ่งที่ต้องการได้

การบำรุงรักษาระบบ ควรจะอยู่ภายใต้การดูแลของนักวิเคราะห์ระบบ เมื่อผู้บริหารต้องการแก้ไขส่วนใดนักวิเคราะห์ระบบต้องเตรียมแผนภาพต่างๆ และศึกษาผลกระทบต่อระบบและให้ผู้บริหารตัดสินใจต่อไปว่าควรจะแก้ไขหรือไม่ (อำไพ วรรณสินธุ์, ม.ป.ป.)

ในการศึกษา ผู้ศึกษาจะดำเนินการตามหลักการพัฒนาระบบแบบวงจรการพัฒนาระบบ (SDLC)

2.4 แนวคิดเชิงวัตถุ (Object-Oriented concept)

แนวคิดเชิงวัตถุมุ่งเน้นสิ่งต่างๆ ที่ใกล้เคียงกับโลกแห่งความจริงในลักษณะรูปธรรม โดยจะมองระบบเป็นกลุ่มของออบเจ็กต์ที่มีปฏิกิริยาต่อกัน มีการนำข้อมูลและฟังก์ชันการทำงานรวมเข้าด้วยกันในออบเจ็กต์ ทำให้ข้อมูลที่เป็นออบเจ็กต์นั้นสามารถอธิบายคุณสมบัติรวมทั้งมีฟังก์ชันการทำงานในตัวเองได้

แนวคิดเชิงวัตถุได้เข้าไปปรากฏในภาษาโปรแกรมปลายทศวรรษ 1960 เพื่อให้สามารถทำงานแบบเชิงวัตถุอันประกอบไปด้วยความสามารถหลักที่เกี่ยวกับการห่อหุ้ม (Encapsulation) การรับทอดคุณสมบัติ (Inheritance) และการเปลี่ยนรูป (Polymorphism) แนวคิดในการสร้างโปรแกรมเชิงวัตถุที่เกิดขึ้นนี้ ใช้เวลาค่อนข้างมาก กว่าจะเป็นที่ยอมรับโดยทั่วไป เพราะเป็นการปฏิวัติแนวทางการพัฒนาโปรแกรมในหลายด้าน ส่วนสำคัญที่ทำให้วิธีการเขียนโปรแกรมเชิงวัตถุเข้ามามีบทบาทมากขึ้นอย่างเห็นได้ชัด คือ การพัฒนาภาษาซีพลัสพลัส ในช่วงต้นทศวรรษที่ 1980 ทำให้มีภาษาโปรแกรมเชิงวัตถุที่ทำงานได้ดีและสะดวกขึ้น อีกทั้งทำให้ผู้เขียนภาษาซีที่มีอยู่แล้วจำนวนมาก สามารถเรียนรู้ แนวทางใหม่ของการโปรแกรมเชิงวัตถุได้ง่าย

ระบบคอมพิวเตอร์ใหม่ๆ ที่เกิดขึ้นในช่วงทศวรรษที่ 1980 เป็นเครื่องแสดงให้เห็นถึงความสำคัญของแนวคิดเชิงวัตถุ ซึ่งจะทำให้ผู้เขียนโปรแกรมสามารถสร้างโปรแกรมโดยการใช้ชิ้นส่วนสำเร็จภาพมาประกอบกันเป็นโปรแกรม อันเป็นวิธีที่ทำให้การสร้างโปรแกรมขนาดใหญ่เป็นไปได้ง่ายขึ้น เชื่อถือได้มากขึ้น แก้ไขเปลี่ยนแปลงได้ง่ายขึ้นและสร้างได้รวดเร็วขึ้นกว่าวิธีแบบเดิมที่ใช้กันมาแม้ว่าวิธีการเชิงวัตถุจะไม่ใช่แนวทางที่จะแก้ปัญหาทางซอฟต์แวร์ได้ทุกอย่าง

แต่ได้แสดงให้เห็นความสามารถอย่างสำคัญในการช่วยลดภาระ ลดปัญหา และลดค่าใช้จ่ายในการสร้างซอฟต์แวร์ขนาดใหญ่ได้เป็นอย่างดี (รอม หิรัญพฤกษ์, 2536, หน้า 191-192)

ออบเจ็กต์

ออบเจ็กต์เป็นสิ่งที่มีทั้งข้อมูลและพฤติกรรม ออบเจ็กต์เป็นสิ่งที่ใช้แทนสิ่งที่มีอยู่ในโลกความจริง (อภิเนตร อูนากูล, 2543, หน้าที่ 21) เช่น สุนัข ระบบควบคุมการไหลของอากาศ เซนเซอร์หรือเครื่องยนต์หรืออาจจะใช้แทนสิ่งที่เป็นนามธรรม เช่น ความเป็นเจ้าของ เที่ยวบิน การวิ่ง หรือออบเจ็กต์สามารถเป็นสิ่งที่มองเห็นได้เช่น จักรยาน รถ องค์กร ใบรายการสินค้า เป็นต้น โดยออบเจ็กต์ทั้งหมดนี้จะเป็นสิ่งที่ผู้พัฒนาสนใจ ซึ่งออบเจ็กต์แบบใดก็ตามล้วนมีลักษณะ ดังนี้

(1) แอททริบิวต์ คือ คุณสมบัติของออบเจ็กต์ที่เก็บไว้เพื่อที่จะให้ผู้อื่นเรียกไปใช้งาน เก็บไว้ใช้ในการประมวลผล เพื่อตอบสนองกับเหตุการณ์ภายนอกหรืออาจจะเก็บไว้เพื่อเป็นสถานะของตัวออบเจ็กต์เอง โดยทั่วไปแอททริบิวต์ของออบเจ็กต์จะถูกซ่อนไว้จากผู้ที่จะใช้งานออบเจ็กต์ ทั้งนี้เนื่องจากการเข้าใช้งานแอททริบิวต์โดยตรงเปรียบเสมือนเป็นการเข้าถึงโครงสร้างของ ตัวอย่างออบเจ็กต์ที่พบเห็นอยู่บ่อยๆ เช่น ออบเจ็กต์รถยนต์มีแอททริบิวต์ เช่น ความกว้าง ความยาว สี น้ำหนัก เป็นต้นวัตถุ (อภิเนตร อูนากูล, 2543, หน้าที่ 26)

(2) เมธอด (Method) คือ พฤติกรรม (Behavior) หรือบริการ ที่ออบเจ็กต์สามารถกระทำให้ได้ เช่น ออบเจ็กต์รถยนต์ อาจจะมีเมธอด สตาร์ทเครื่องยนต์ ออกวิ่ง เบรก ดับเครื่องยนต์เปิดไฟ ยกเลี้ยว เป็นต้น ทั้งนี้จุดประสงค์หลักของเมธอดก็เพื่อใช้ในการจัดการกับแอททริบิวต์นั่นเอง (ชาลี และเทพฤทธิ์, 2544, หน้าที่ 16)

(3) สถานะ (State) คือ สถานะของออบเจ็กต์ ณ เวลาหนึ่ง โดยที่สถานะของออบเจ็กต์สามารถเปลี่ยนแปลงได้จากเดิม เมื่อมีเงื่อนไขบางประการ เช่น ออบเจ็กต์รถยนต์ มีสถานะของรถยนต์ เช่น กำลังออกตัว กำลังลดความเร็ว เป็นต้น

(4) เอกลักษณ์เฉพาะตัวของออบเจ็กต์หรือโอไอดี (Object Identity: OID) คือ คุณลักษณะบางอย่างที่ทำให้ออบเจ็กต์แต่ละออบเจ็กต์แตกต่างกัน โดยระบบจะกำหนดโอไอดีให้กับออบเจ็กต์แต่ละตัว โอไอดีไม่ขึ้นอยู่กับค่าแอททริบิวต์ใดๆ ในออบเจ็กต์ ณ เวลาที่ออบเจ็กต์ถูกสร้างขึ้นมาและไม่สามารถเปลี่ยนแปลงได้ จะถูกลบได้ก็ต่อเมื่อออบเจ็กต์นั้นถูกล้างออกไปเท่านั้น และไม่มีการนำโอไอดีนั้นกลับมาใช้อีก มีลักษณะแตกต่างจากคีย์หลักที่ใช้ในแบบจำลองเชิงสัมพันธ์ตรงที่ คีย์หลักขึ้นอยู่กับการกำหนดค่าจากผู้ใช้ที่จะเลือกแอททริบิวต์ใดเป็นคีย์ก็ได้ และสามารถเปลี่ยนแปลงได้ตลอดเวลา (อัจฉราและคณะ, 2544, หน้า 58)

(5) ข้อความ (Message) คือ สารที่ส่งจากออบเจ็กต์หนึ่งไปยังอีกออบเจ็กต์หนึ่ง เพื่อให้ออบเจ็กต์อื่นทำงานตามที่ต้องการหรือการเรียกใช้งานเมธอดของออบเจ็กต์นั่นเอง

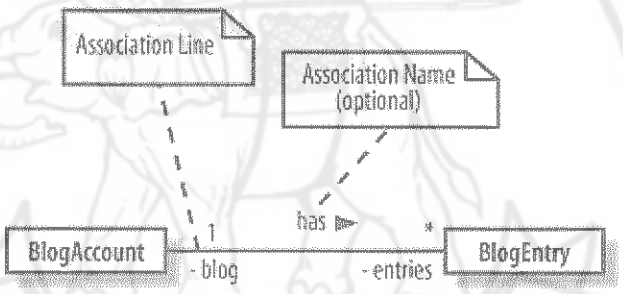
คลาส

กลุ่มของออบเจ็กต์ที่มีโครงสร้างพื้นฐานคุณสมบัติและพฤติกรรมเดียวกัน ดังนั้นออบเจ็กต์ที่มีคุณสมบัติลักษณะเดียวกันนี้จะรวมกลุ่มอยู่ในคลาสเดียวกัน หรืออาจกล่าวได้ว่าคลาสคือต้นแบบข้อมูลที่มีไว้เพื่อสร้างออบเจ็กต์

ความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาส สามารถแบ่งได้ออกเป็น 3 ลักษณะด้วยกันคือ

(1) แอสโซซิเอชัน (Association) คือ ความสัมพันธ์ระหว่างคลาสหนึ่งกับอีกคลาสหนึ่ง แต่ไม่ได้มีความสัมพันธ์แบบเป็นส่วนหนึ่งส่วนใด เช่นความสัมพันธ์ระหว่างนักบินกับเครื่องบิน (โอภาส เอี่ยมสิริวงศ์, 2544, หน้าที่ 174)

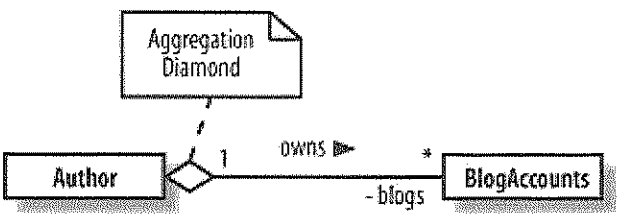


ภาพ 5 ตัวอย่างความสัมพันธ์แบบแอสโซซิเอชัน

(ที่มา: Learning UML 2.0 By Kim Hamilton, Russell Miles, 2006)

(2) อะกรีเกชัน (Aggregation) คือ ความสัมพันธ์ในลักษณะที่ออบเจ็กต์หนึ่งเป็นส่วนหนึ่งของอีกออบเจ็กต์หนึ่ง (สุนทริน วงศ์ศิริกุล, 2545, หน้าที่ 13) แบ่งออกเป็น 2 รูปแบบคือ

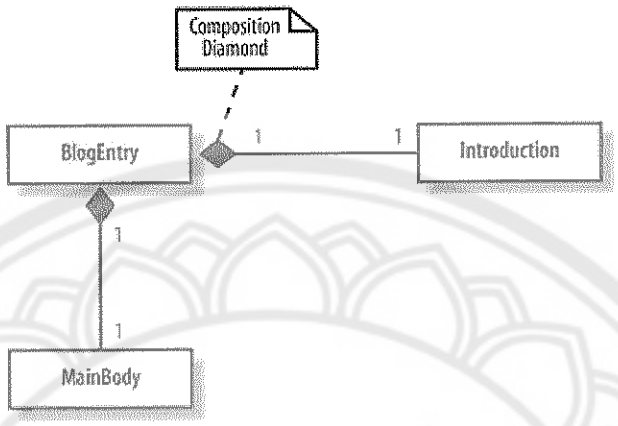
แชร์ อะกรีเกชัน (Shared Aggregation) คือ ความสัมพันธ์แบบเป็นส่วนหนึ่งของ (a part of) โดยจะมีคลาสที่ใหญ่ที่สุดเป็นออบเจ็กต์หลัก และมีคลาสหรือออบเจ็กต์ส่วนอื่นๆ เป็นส่วนหนึ่งของออบเจ็กต์หลัก เช่น รถยนต์ มีส่วนประกอบคือ เครื่องยนต์ ที่นั่ง และล้อ เป็นต้นดังตัวอย่างภาพ 4



ภาพ 6 ตัวอย่างความสัมพันธ์แบบอะกรีเกชัน

(ที่มา: Learning UML 2.0 By Kim Hamilton, Russell Miles, 2006)

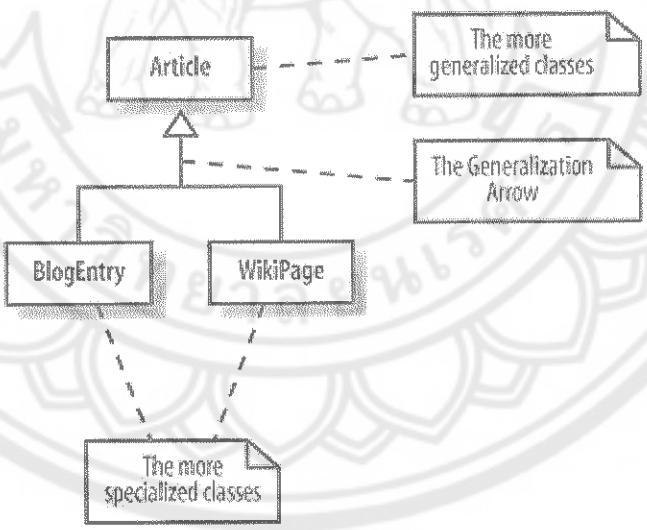
คอมโพสิต อะกรีเกชัน (Composite Aggregation) คือ ความสัมพันธ์ที่มีความข้องเกี่ยวกันเสมอ เช่น คลาสปุ่มไม่สามารถมีได้ หากไม่มีคลาสวินโดว์ เป็นต้น



ภาพ 7 ตัวอย่างความสัมพันธ์แบบคอมโพสิต อะกรีเกชัน

(ที่มา: Learning UML 2.0 By Kim Hamilton, Russell Miles, Publisher: 2006)

(3) เจเนอรัลไลเซชัน (Generalization) คือ ความสัมพันธ์ระหว่างคลาสในลักษณะของการรับทอดคุณสมบัติ จากโครงสร้างคลาสหนึ่งไปยังโครงสร้างอีกคลาสหนึ่ง (โอภาส เอี่ยมสิริวงศ์, 2544, หน้าที่ 175-176)



ภาพ 8 ตัวอย่างความสัมพันธ์แบบเจเนอรัลไลเซชัน

(ที่มา: Learning UML 2.0 By Kim Hamilton, Russell Miles, 2006)

คุณลักษณะของวิธีการเชิงวัตถุ

(1) การกำหนดสาระสำคัญ (Abstraction) คือ การตีกรอบแนวความคิดเพื่อให้มองเห็นสิ่งที่เกี่ยวข้องในระบบที่สนใจ และลดความซ้ำซ้อนของสิ่งที่จะนำมาใช้ระบบ เช่น เมื่อพูดถึงคอมพิวเตอร์ บางคนอาจจะคิดถึงเครื่องคอมพิวเตอร์ บางคนอาจจะคิดถึงเครื่องโน้ตบุ๊ก เป็นต้น

IS. IT
ส 379 ป
2551
ฉ.2

(2) การห่อหุ้ม คือ การรวบรวมเมธอดและแอททริบิวต์เข้าเป็นหน่วยเดียวกันเพื่อที่แอททริบิวต์สามารถถูกเปลี่ยนแปลงได้อย่างเหมาะสมโดยผ่านเมธอด และเราจะเรียกผลที่เกิดจากการใช้งานการห่อหุ้มว่า การซ่อนข้อมูล (Information Hiding)

(3) การรับทอดคุณสมบัติ คือ วิธีการในการสร้างคลาสใหม่จากคลาสเดิมที่มีอยู่ ทั้งนี้ คลาสที่สร้างขึ้นใหม่จะมีวัตถุประสงค์ในการทำงานที่เฉพาะเจาะจงมากยิ่งขึ้น ซึ่งคลาสที่ถ่ายทอดคุณสมบัติจะเรียกว่าซูเปอร์คลาสหรือคลาสแม่ และเรียกคลาสที่ได้รับการสืบทอดคุณสมบัติว่าซับคลาสหรือคลาสลูก เช่น เมื่อพูดถึงห้อง อาจแบ่งเป็น 3 ประเภทคือ ห้องเรียน ห้องปฏิบัติการ และห้องประชุม ในส่วนของห้องเรียนก็อาจจำแนกได้เป็น ห้องเรียนปกติ, ห้องสโลป ในส่วนของห้องปฏิบัติการก็อาจจำแนกเป็นห้องปฏิบัติการคอมพิวเตอร์ ห้องปฏิบัติการเคมี ห้องปฏิบัติการฟิสิกส์ เป็นต้น

(4) การเปลี่ยนรูป คือ การที่เมธอดเดียวกันมีพฤติกรรมที่แตกต่างกันเมื่อถูกใช้กับออบเจ็กต์ที่เกิดจากคนละคลาสกัน กล่าวอีกนัยหนึ่งคือออบเจ็กต์ที่เกิดจากต่างคลาสกันสามารถที่จะมีปฏิกิริยาตอบสนองต่อเมธอดชื่อเดียวกันได้อย่างแตกต่างกัน เช่น เมธอดที่ออกแบบไว้สำหรับการรับค่าพารามิเตอร์จำนวน 3 ค่า แต่สามารถเรียกใช้งานได้แม้วาจำนวนพารามิเตอร์จะไม่ครบก็ตาม นอกจากนั้นแล้วการโปรแกรมเชิงวัตถุยังยอมให้ผู้ใช้สามารถกำหนดใช้เมธอดที่มีชื่อเดียวกันแต่มีจำนวนพารามิเตอร์ที่แตกต่างกันได้ (Edward Yourdon, 2547, หน้า 6-8)

ข้อดีของการประยุกต์ใช้แนวคิดเชิงวัตถุ

(1) สนับสนุนการพัฒนาระบบที่ซับซ้อน ในการพัฒนาโปรแกรมรูปแบบเดิม นักพัฒนาโปรแกรมจะทำงานกันที่ระดับฟังก์ชัน กล่าวคือตั้งแต่เริ่มการพัฒนาจนจบ จะเป็นการสร้างปรับปรุง แก้ไข รายละเอียดของฟังก์ชันต่างๆ ภายในตัวโปรแกรม แต่สำหรับแนวคิดเชิงวัตถุนักพัฒนาจะทำการวิเคราะห์ออกแบบระบบงานกันในระดับของออบเจ็กต์ซึ่งประกอบด้วยทั้งข้อมูลและเมธอดภายในแต่ละออบเจ็กต์ ดังนั้นจึงกล่าวได้ว่าการอาศัยแนวคิดเชิงวัตถุจะช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้ง่ายขึ้นมาก

(2) สนับสนุนการนำกลับมาใช้งานซ้ำอีกครั้ง เนื่องจากแต่ละคลาสหรือออบเจ็กต์ที่กำหนดขึ้นนั้นจะมีความสมบูรณ์อยู่ในตัวมันเองบนพื้นฐานแนวคิดของแต่ละออบเจ็กต์เอง รวมทั้งยังเป็นอิสระจากสภาพแวดล้อมอื่น ดังนั้นแต่ละคลาสจึงง่ายต่อการนำกลับมาใช้งานใหม่ การนำกลับมาใช้งานอาจอยู่ในรูปแบบของการสืบทอดคุณสมบัติระหว่างออบเจ็กต์ หรือการใช้งานซอฟต์แวร์คอมโพเนนต์ก็เช่นกัน

300551
i. 4439399

(3) ปรับปรุงแก้ไขและบำรุงรักษาง่าย เนื่องจากข้อมูลและเมธอดการทำงานที่เกี่ยวข้องกับออบเจ็กต์หนึ่งๆ จะถูกรวบรวมอยู่ที่เดียวกัน การทำงานภายในของแต่ละออบเจ็กต์จะไม่เกี่ยวข้องพึ่งพิงกับโค้ดที่อยู่ภายนอกออบเจ็กต์ ดังนั้นนักพัฒนาสามารถทำการแก้ไขปรับปรุงรายละเอียดภายในของแต่ละคลาสได้โดยไม่กระทบต่อส่วนที่เรียกใช้งานภายนอกแต่อย่างใด นอกจากนี้ในการขยายระบบก็สามารถทำได้ง่าย โดยการสร้างออบเจ็กต์หรือคลาสเพิ่มเติมลงไปในตัวโปรแกรมนั่นเอง (ชาลีและเทพฤทธิ์, 2544, หน้าที่ 18)

ข้อจำกัดของการประยุกต์ใช้แนวคิดเชิงวัตถุ

เนื่องจากแนวคิดเชิงวัตถุเป็นการปฏิวัติแนวทางการพัฒนาโปรแกรมในหลายด้าน ไม่ว่าจะเป็นในด้านแนวความคิด การวิเคราะห์และการออกแบบ ขั้นตอนการทำงาน สิ่งเหล่านี้เป็นเรื่องใหม่สำหรับนักพัฒนาที่เคยชินกับการพัฒนาระบบด้วยวิธีการแบบดั้งเดิม ทำให้ต้องใช้เวลาในการเรียนรู้พอสมควร

ในการศึกษาครั้งนี้ ผู้ศึกษาจะเลือกใช้เพียงแค่สามไดอะแกรม คือ Usecase Diagram, Class Diagram และ Sequence Diagram เท่านั้นในการออกแบบระบบ

2.4.1 แบบจำลองเชิงวัตถุด้วยยูเอ็มแอล

เทคโนโลยีในการพัฒนาระบบด้วยวิธีการเชิงวัตถุเป็นหลักการทางวิศวกรรม ซึ่งบรรจุไว้ด้วยองค์ประกอบต่างๆ ที่เรียกว่าแบบจำลองเชิงวัตถุ แบบจำลองเชิงวัตถุจะอธิบายถึงออบเจ็กต์ซึ่งเป็นที่รวมของข้อมูล ความสัมพันธ์ระหว่างออบเจ็กต์และการประมวลผลในระบบที่สนใจสำหรับงานวิจัยนี้เลือกใช้คลาสไดอะแกรมของยูเอ็มแอล สำหรับใช้ในการแสดงผลลัพธ์ของเค้าร่างฐานข้อมูลเชิงวัตถุ

ยูเอ็มแอล (Unified Modeling Language: UML)

(1) แบบจำลองในวลี "ภาษาแบบจำลอง" (Modeling Language) คือ ความพยายามในการที่จะอธิบายปัญหาของซอฟต์แวร์ที่จะดำเนินการพัฒนาขึ้นมา ตัวแบบจำลองจะแสดงให้เห็นถึงออบเจ็กต์ต่างๆ ที่เกี่ยวข้อง และความสัมพันธ์ระหว่างออบเจ็กต์เหล่านั้น นอกจากนั้นแบบจำลองยังแสดงให้เห็นถึงวิธีการที่จะแก้ไขปัญหา เราอาจจะใช้ไดอะแกรม เนื้อความ (Text) หรือรูปแบบอื่นๆ ซึ่งเป็นที่ยอมรับกันระหว่างผู้พัฒนาและผู้ใช้ระบบในการนำเสนอแบบจำลองหนึ่งๆ ดังนั้นเมื่อกล่าวถึงภาษาแบบจำลอง จะหมายถึงภาษาที่เอาไว้อธิบายแบบจำลองนั่นเอง ภาษาแบบจำลองทั้งหลายมักจะใช้ไดอะแกรมหรือเนื้อความในการอธิบายถึงออบเจ็กต์และความสัมพันธ์ระหว่างออบเจ็กต์เหล่านั้น (บรรจงและญาณวรรณ, 2542, หน้าที่ 185)

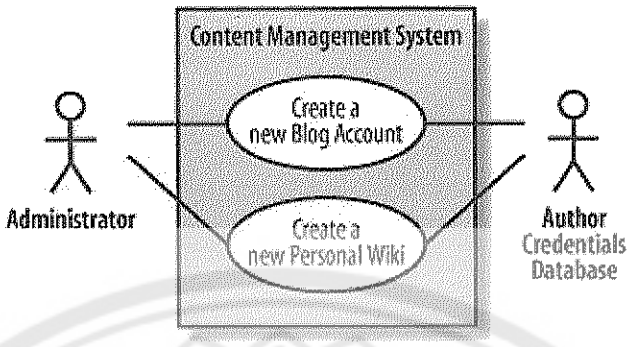
(2) ยูเอ็มแอล เป็นภาษาลักษณ์รูปภาพมาตรฐานสำหรับใช้ในการสร้างแบบจำลองเชิงวัตถุและจัดการเอกสารต่างๆ ในการผลิตระบบซอฟต์แวร์ โดยทำให้การออกแบบซอฟต์แวร์หรือการสร้างพิมพ์เขียวของซอฟต์แวร์ทำได้ง่าย ผู้คิดค้นยูเอ็มแอลเกิดจากการพัฒนาของทีมบุคลากรแห่งบริษัท เรชันนอลซอฟต์แวร์และความร่วมมือจากบริษัทซอฟต์แวร์และคอมพิวเตอร์ต่างๆ เช่น ไอบีเอ็ม ไมโครซอฟต์ ออราเคิล ฮิวเล็ตแพ็กการ์ด ออบเจ็กต์ไทม์และยูนีซีส ยูเอ็มแอล เป็นภาษาแบบจำลองภาษาหนึ่ง ซึ่งสามารถใช้ในการแก้ไขปัญหาในการดำเนินงานโครงการซอฟต์แวร์ ในการแก้ปัญหาหนึ่งๆ ยูเอ็มแอลจะใช้แบบจำลองที่มีรูปแบบแตกต่างกัน แบบจำลองที่ยูเอ็มแอลใช้จะมีลักษณะต่อเนื่องกันไป คือแบบจำลองหนึ่งจะอาศัยแบบจำลองที่สร้างขึ้นมาก่อนหน้านี้เพื่อทำการสร้างแบบจำลองต่อไป โดยแต่ละแบบจำลองจะมีมุมมองของปัญหาในแง่ที่แตกต่างกันออกไป แต่เมื่อเอาแบบจำลองเหล่านั้นมาประกอบกันเข้า ก็จะสามารถดำเนินการวิเคราะห์ ออกแบบ และพัฒนาซอฟต์แวร์ได้อย่างมีประสิทธิภาพ (บรรจงและญาณวรรณ, 2542, หน้าที่ 185)

ประเภทแผนภาพของยูเอ็มแอล

ยูเอ็มแอล ประกอบด้วยแผนภาพทั้งหมด 9 แผนภาพ เพื่อใช้ในการจำลองระบบงานเปรียบได้กับการมองในแง่มุมต่างๆ (แผนภาพแต่ละชนิด) เพื่อให้สามารถเข้าใจระบบงานให้มากที่สุด ผู้ใช้จำลองแบบไม่จำเป็นต้องใช้ทุกแผนภาพ สามารถเลือกใช้แผนภาพที่เหมาะสมยูเอ็มแอลประกอบด้วยแผนภาพต่างๆ ดังนี้(บรรจงและญาณวรรณ, 2542, หน้าที่ 185)

(1) ยูสเคสไดอะแกรม (Use-case Diagram) ใช้มองภาพรวมของระบบและความต้องการต่างๆ ซึ่งคล้ายกับแผนภาพการไหลของข้อมูล ซึ่งสามารถบอกได้ว่า ใครเกี่ยวข้องกับงานใดระบบอะไรและมีงานหลักๆ อะไรบ้าง และใช้สำหรับกำหนดความต้องการของระบบผ่านมุมมองของผู้ใช้ โดยเน้นไปที่ความสัมพันธ์ที่เกิดขึ้นในระบบ

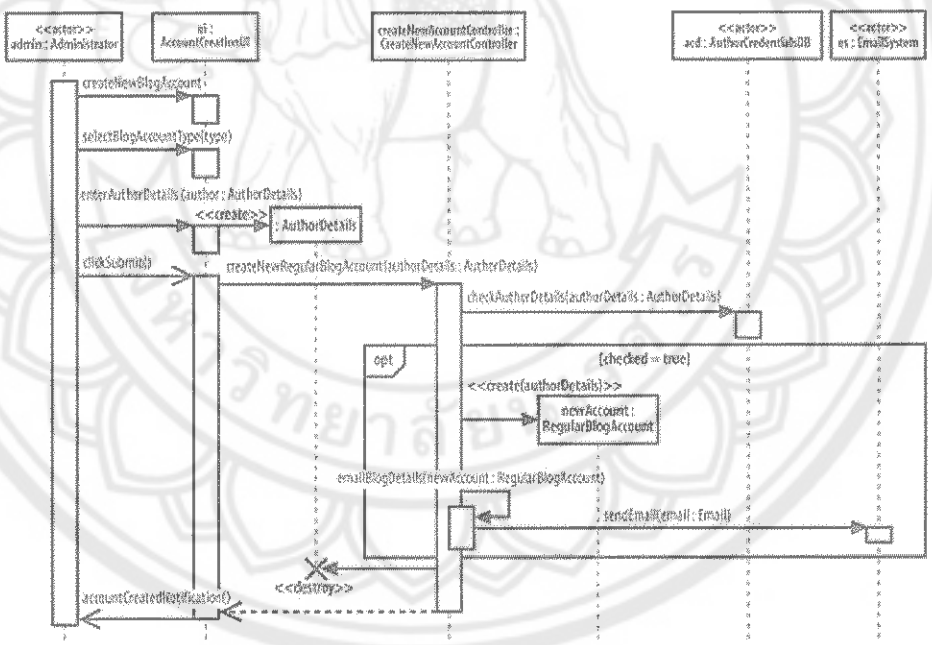
ยูสเคสไดอะแกรม มีองค์ประกอบ 2 ส่วนคือ แอ็กเตอร์ (actor) และยูสเคส (Use-case) แอ็กเตอร์ คือ สิ่งที่อยู่นอกระบบแต่เป็นผู้ให้อะไรบางอย่างแก่ระบบ ส่วนยูสเคสจะแสดงถึงขอบเขตของระบบที่เรากำลังสนใจ



ภาพ 9 ตัวอย่างยูสเคสไดอะแกรม

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

(2) ซีเควนไดอะแกรม (Sequence Diagram) เป็นแผนภาพที่แสดงขั้นตอนการทำงานของแต่ละยูสเคสระหว่างออบเจ็กต์ต่างๆ ที่ส่งข้อความถึงกันและกัน โดยแผนภาพนี้จะช่วยให้โปรแกรมเมอร์เห็นภาพรวม ทำให้ง่ายต่อความเข้าใจในการเขียนและควบคุมโปรแกรมตามที่ออกแบบไว้ อย่างไรก็ตามซีเควนไดอะแกรม จะไม่ได้แสดงความสัมพันธ์ระหว่างออบเจ็กต์

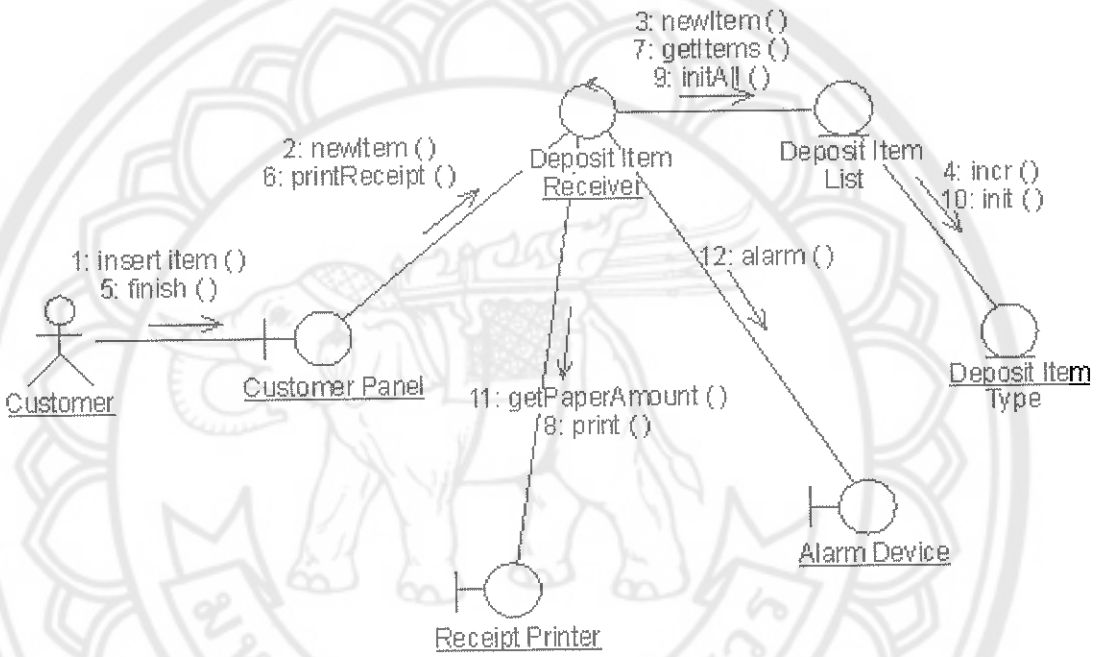


ภาพ 10 ตัวอย่างซีเควนไดอะแกรม

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

(3) ออบเจ็กต์ไดอะแกรม (Object Diagram) เป็นแผนภาพที่แสดงความสัมพันธ์ระหว่างออบเจ็กต์ที่เชื่อมโยงกันในช่วงเวลาหนึ่งเท่านั้น โดยสัญลักษณ์ของออบเจ็กต์ไดอะแกรมจะมีลักษณะเดียวกับคลาสไดอะแกรม ต่างกันที่ชื่อของออบเจ็กต์ไดอะแกรมจะมีการขีดเส้นใต้เอาไว้ด้วย

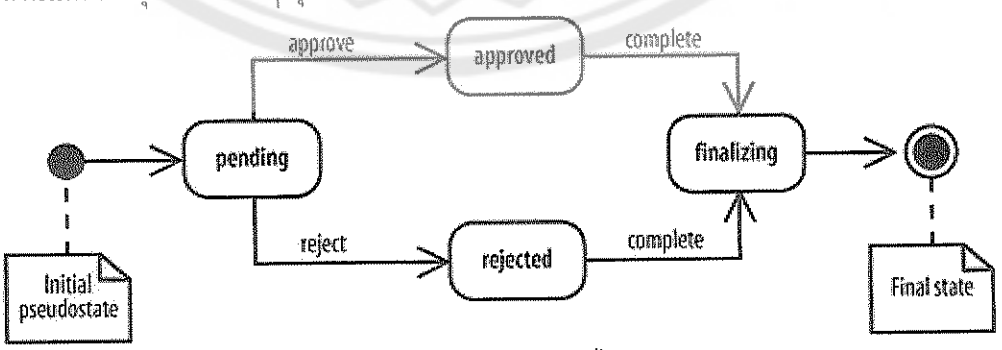
(4) คอลแลบอเรชันไดอะแกรม (Collaboration Diagram) เป็นแผนภาพชนิดเดียวกันกับซีเควทไดอะแกรม โดยซีเควนไดอะแกรมเป็นแผนภาพที่แสดงถึงการแลกเปลี่ยนข่าวสาร แต่คอลแลบอเรชันไดอะแกรมแสดงความสัมพันธ์ระหว่างออบเจ็กต์และการปฏิสัมพันธ์ โดยจะแสดงลำดับการทำงานก่อนและหลัง วิธีการเลือกใช้ คือ ถ้าเป็นการกำหนดช่วงเวลาที่แน่นอนและใช้เวลาเป็นสิ่งสำคัญให้เลือกใช้ซีเควนไดอะแกรม แต่ถ้าเป็นการให้ความสำคัญภายในออบเจ็กต์ให้เลือกใช้คอลแลบอเรชันไดอะแกรม



ภาพ 11 ตัวอย่างซีเควนไดอะแกรม

(ที่มา:http://venus.imp.mx/hilario/personal/RUP2001.03.00.01/process/modguide/md_coldm.htm)

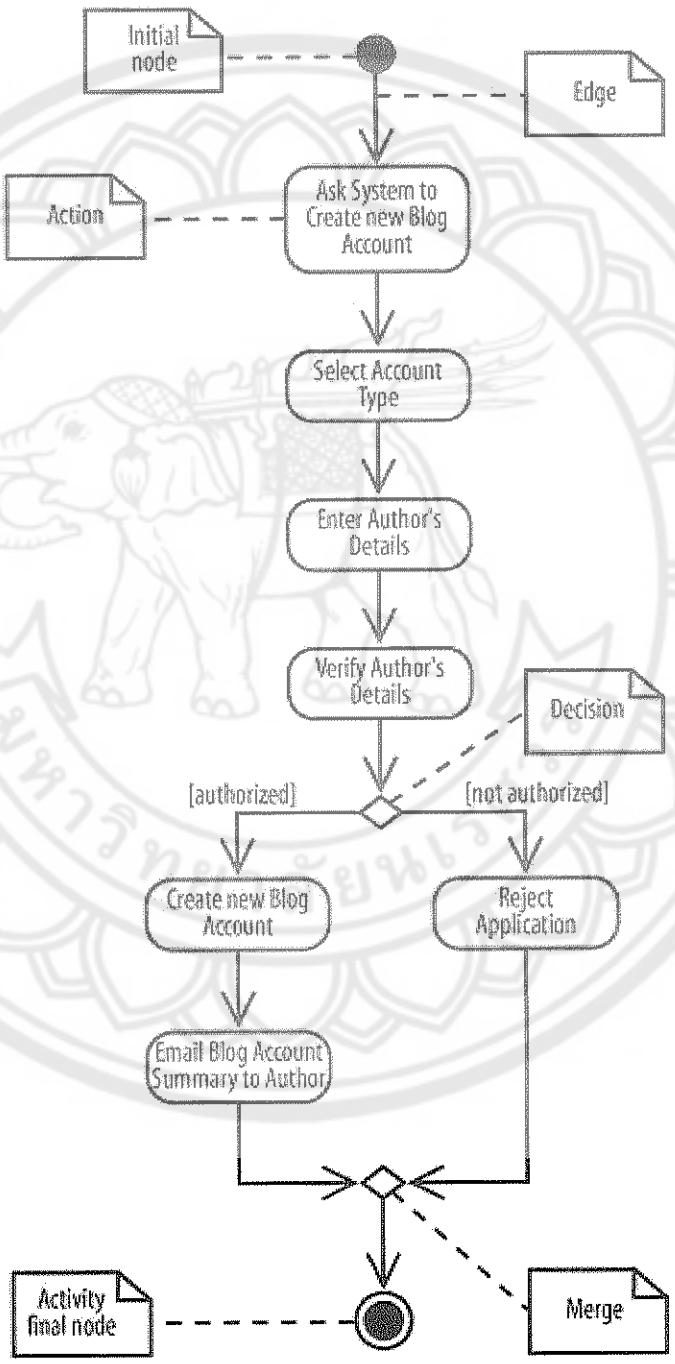
(5) สเตทชาร์ทไดอะแกรม (Statechart Diagram) เป็นแผนภาพที่แสดงถึงวงจรชีวิตของออบเจ็กต์โดยแสดง เหตุการณ์ต่างๆ (event) ที่มีผลกระทบต่อสถานะของออบเจ็กต์ ซึ่งอาจจะมีจุดเริ่มต้นและสิ้นสุดได้หลายๆ จุด



ภาพ 12 ตัวอย่างสเตทชาร์ทไดอะแกรม

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

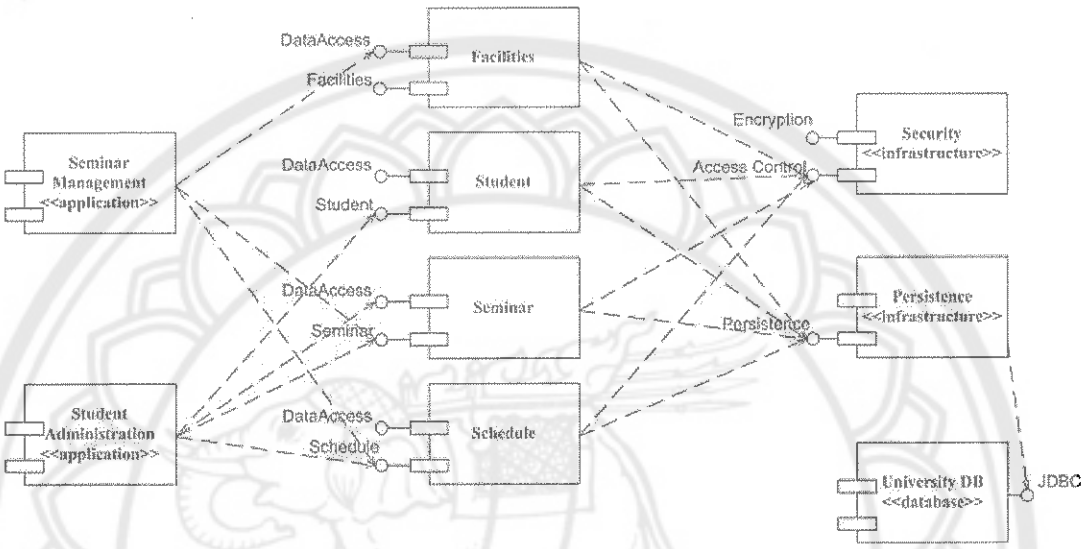
(6) แอ็กทิวิตีไดอะแกรม (Activity Diagram) เป็นแผนภาพที่แสดงขั้นตอนของการปฏิบัติงานและจุดที่ต้องมีการตัดสินใจที่เกิดภายในออบเจ็กต์หรือภายในกระบวนการทำงาน โดยแต่ละขั้นตอน (Activity) จะแสดงอยู่ภายในวงรี และจุดที่มีการตัดสินใจจะแทนด้วยรูปสี่เหลี่ยมขนมเปียกปูน



ภาพ 13 ตัวอย่างแอ็กทิวิตีไดอะแกรม

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

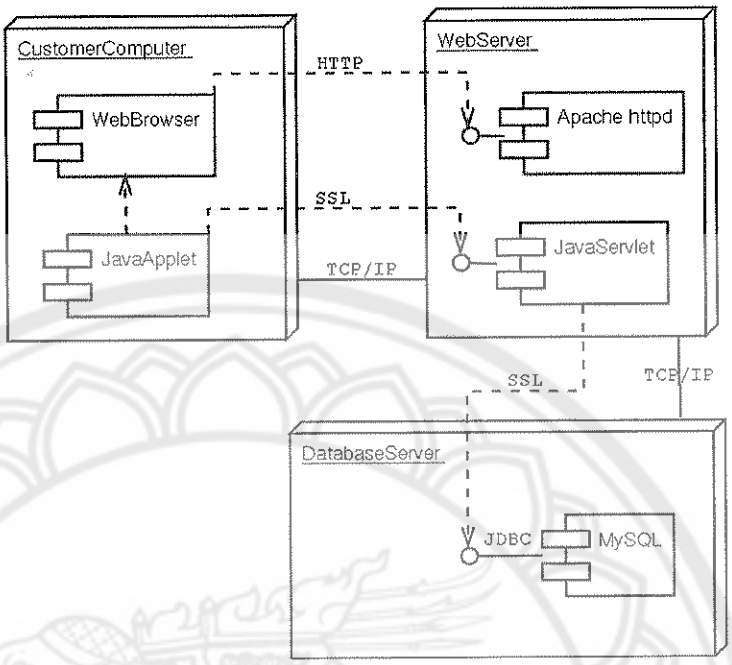
(7) คอมโพเนนต์ไดอะแกรม (Component Diagram) เป็นแผนภาพแสดงโครงสร้างทางกายภาพ ในส่วนของซอฟต์แวร์คอมโพเนนต์ เช่น ตัวโปรแกรม โปรแกรมที่ถูกคอมไพล์แล้วและส่วนติดต่อผู้ใช้งาน องค์ประกอบต่างๆ ของระบบที่เชื่องโยงกันจะเป็นความสัมพันธ์ในลักษณะขึ้นต่อกัน โดยจะแสดงในรูปของเส้นประที่มีหัวลูกศรชี้จากคอมโพเนนต์ลูกไปยังคอมโพเนนต์หลัก



ภาพ 14 ตัวอย่างคอมโพเนนต์ไดอะแกรม

(ที่มา: The Object Primer 3rd Edition Agile Model Driven Development with UML 2 by Scott W. Ambler, 2004)

(8) ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram) เป็นแผนภาพสำหรับแสดงการเชื่อมต่อของอุปกรณ์ฮาร์ดแวร์ในระบบและมักใช้ร่วมกับคอมโพเนนต์ไดอะแกรม โดยข้างในฮาร์ดแวร์อาจประกอบด้วยซอฟต์แวร์คอมโพเนนต์ ดังภาพ 15 ดีพลอยเมนต์ไดอะแกรม จะแสดงอยู่ในรูปออบเจ็กต์และแสดงในช่วงเวลาของการรันหรือระหว่างการเอ็กซิคิวต์ ดังนั้นไฟล์คอมโพเนนต์ของระบบที่ไม่ได้ใช้สำหรับรัน เช่น ตัวโปรแกรม จะไม่ปรากฏในแผนภาพประเภทนี้

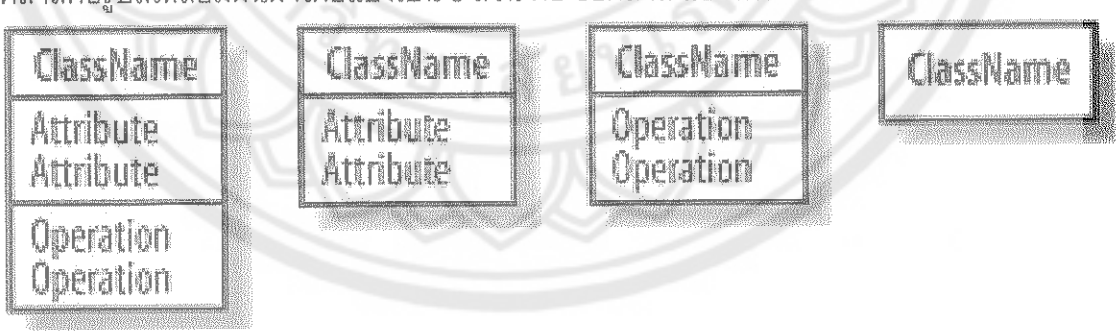


ภาพ 15 ตัวอย่างดีพลอยเมนต์ไดอะแกรม

(ที่มา: http://www.vinci.org/uml/deploymentx.html by Ralph L. Vinciguerra)

(9) คลาสไดอะแกรม

เป็นแผนภาพที่ประกอบไปด้วยคลาสต่างๆ ที่มาประกอบกันกลายเป็นระบบหรือซอฟต์แวร์ตัวหนึ่ง โดยคลาสต่างๆ นี้มีที่มาจากซีเควนไดอะแกรมหรือคอลแลบอเรชันไดอะแกรม แต่ละคลาสจะประกอบไปด้วย 3 ส่วน คือ ชื่อคลาส แอททริบิวต์และเมธอดสัญลักษณ์ที่ใช้แทนคลาสคือรูปสี่เหลี่ยมผืนผ้าโดยแบ่งเป็น 3 ส่วน คือ ชื่อคลาส แอททริบิวต์และเมธอด

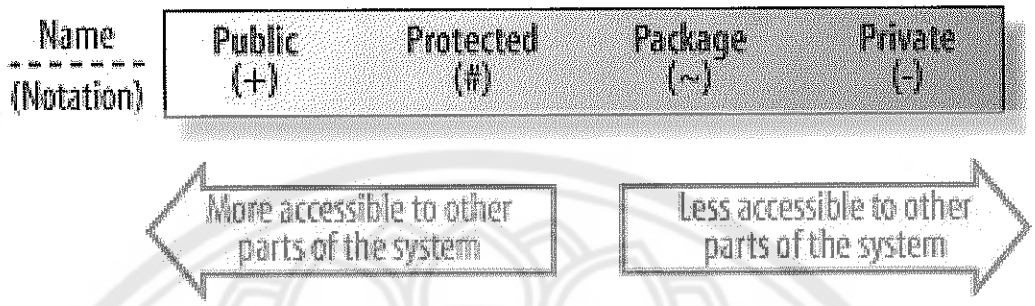


ภาพ 16 สัญลักษณ์ของคลาส

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

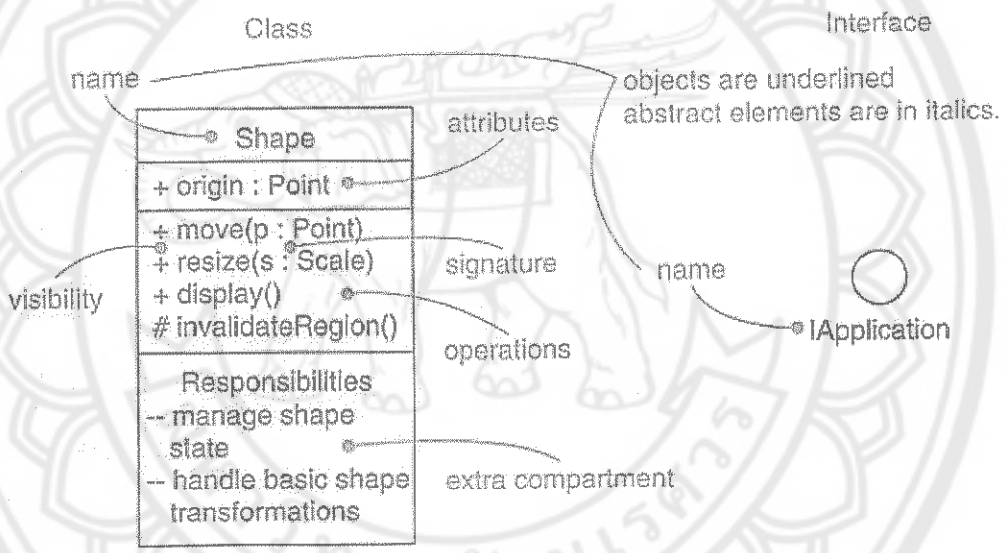
ในส่วนของสัญลักษณ์ที่อยู่ด้านหน้าแอททริบิวต์หรือเมธอด จะเป็นส่วนที่ใช้แสดงความสามารถเข้าถึงข้อมูลในคลาสของแอททริบิวต์หรือเมธอดนั้น ได้แก่ Public จะใช้สัญลักษณ์เป็นเครื่องหมายบวก (+) ซึ่งจะอนุญาตให้ทุกๆ คลาส สามารถเข้าถึงแอททริบิวต์หรือเมธอดของคลาสนั้นๆ ได้ private จะใช้สัญลักษณ์เป็นเครื่องหมายลบ (-) ซึ่งจะไม่อนุญาตให้เข้าถึงแอททริ

บิวต์หรือเมธอดของคลาสนั้นๆ ได้ protected จะใช้สัญลักษณ์เป็นเครื่องหมายชาร์ป (#) ซึ่งจะอนุญาตเฉพาะลูกของคลาสนั้นๆ เท่านั้น



ภาพ 17 ลำดับของการเข้าถึง

(ที่มา: Learning UML 2.0 by Kim Hamilton, Russell Miles, 2006)

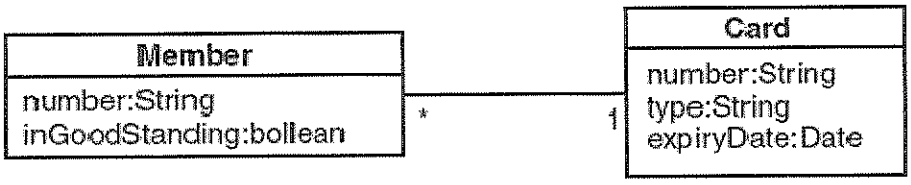


ภาพ 18 สัญลักษณ์ของคลาส

(ที่มา: Grady Booch, James Rumbaugh and Ivar Jacobson, the Unified Modeling Language User Guide, 1999, 435)

คลาสไดอะแกรมนอกจากจะประกอบไปด้วยคลาสต่างๆ แล้ว แต่ละคลาสที่เกิดขึ้นในระบบจะมีความสัมพันธ์กันระหว่างคลาส ซึ่งความสัมพันธ์เหล่านี้สามารถแบ่งได้ดังนี้

(1) แอสโซซิเอชัน แสดงความสัมพันธ์ระหว่างคลาสในลักษณะที่ใช้บริการของอีกคลาสหนึ่ง แต่ไม่ได้มีความสัมพันธ์แบบเป็นส่วนหนึ่ง มีความสัมพันธ์ได้ทั้งทางเดียวและสองทางการกำหนดชื่อและทิศทางของความสัมพันธ์ ในคลาสไดอะแกรมสามารถแทนความสัมพันธ์นี้ได้ด้วยเส้นตรงระหว่างคลาสและมีชื่อความสัมพันธ์กำกับอยู่ ในการกำหนดทิศทางของชื่อความสัมพันธ์ทำได้โดยวาดสามเหลี่ยมทึบไว้ด้านซ้ายหรือด้านขวาของชื่อ ทั้งนี้ขึ้นอยู่กับทิศทางของความสัมพันธ์ซึ่งลูกศรจะช่วยในการอ่านความสัมพันธ์ให้เป็นไปได้อย่างถูกต้อง



ภาพ 19 สัญลักษณ์ของคลาส

(ที่มา: Mike O'Docherty, Object-Oriented Analysis and Design Understanding System Development with UML 2.0 2005 หน้าที่ 291)

มัลติพลิซิตี (Multiplicity) คือ การพิจารณาจำนวนออบเจ็กต์ของคลาสหนึ่ง ที่สามารถเชื่อมโยงกับออบเจ็กต์ของคลาสที่เกี่ยวข้อง

สำหรับความสัมพันธ์แบบแอสโซซิเอชัน สามารถแบ่งได้เป็นดังนี้

- ยูนารี แอสโซซิเอชัน (Unary Association) เป็นความสัมพันธ์แบบที่ประกอบด้วยหนึ่งคลาสเท่านั้นหรือเกิดความสัมพันธ์ภายในคลาส

- ไบนารี แอสโซซิเอชัน (Binary Association) เป็นความสัมพันธ์ที่มีคลาสที่เกิดความสัมพันธ์ 2 คลาสเท่านั้น

- แอสโซซิเอชัน คลาส (Association Class) เป็นความสัมพันธ์ที่มีคลาสที่เกิดความสัมพันธ์มากกว่า 1 คลาสขึ้นไป และความสัมพันธ์ที่เกิดขึ้นมีแอททริบิวต์ สามารถแทนความสัมพันธ์นี้ได้ด้วยเส้นปะระหว่างคลาส

- เอ็นนารี แอสโซซิเอชัน (N-ary Association) เป็นความสัมพันธ์ที่มีคลาสเกิดความสัมพันธ์มากกว่า 2 คลาสขึ้นไป

- ควอลิไฟ แอสโซซิเอชัน (Qualified Association) เป็นความสัมพันธ์ที่ถูกทำให้ชัดเจนโดยแอททริบิวต์ที่เรียกว่า ควอลิไฟเออร์ (qualifier) พิจารณาควอลิไฟ แอสโซซิเอชันเช่นเดียวกับแนวคิดของเอนทิตีแบบอ่อนในแบบจำลองความสัมพันธ์เอนทิตี

(2) เจเนอรัลไลเซชัน เป็นความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาส โดยซับคลาสรับถ่ายทอดคุณสมบัติ ได้แก่ แอททริบิวต์และเมธอดมาจากซูเปอร์คลาสของตัวเอง ความสัมพันธ์แบบนี้ถูกเรียกว่า “Is-a” นอกจากนี้ถ้าแอททริบิวต์หรือเมธอดของซูเปอร์คลาสถูกกำหนดให้เป็นค่าใหม่ในซับคลาสจะเป็นการทับคำนิยามที่กำหนดไว้ในซูเปอร์คลาส นำสู่แนวคิดการเปลี่ยนรูป ความสัมพันธ์ระหว่างซูเปอร์คลาสและซับคลาสสามารถแทนได้ด้วยสัญลักษณ์รูปสามเหลี่ยม

- ซิงเกิล เจเนอรัลไลเซชัน (Single Generalization) คือ โครงสร้างลำดับชั้นที่แต่ละซับคลาส มีเพียงหนึ่งซุปเปอร์คลาส

- มัลติเพิล เจเนอรัลไลเซชัน (Multiple Generalization) คือ โครงสร้างลำดับ

ชั้นที่แต่ละซับคลาส มีมากกว่าหนึ่งซุปเปอร์คลาส

(3) อะกรีเกชัน เป็นความสัมพันธ์ระหว่างคลาสหรือออบเจ็กต์ในแง่ของการรวมกัน ความสัมพันธ์ระหว่างคลาสที่เกิดอะกรีเกชันสามารถแทนด้วยสี่เหลี่ยมข้าวหลามตัดติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาสที่หมายถึงสิ่งที่ใหญ่กว่า

(4) คอมโพสิตชัน เป็นความสัมพันธ์การประกอบกันเป็นความสัมพันธ์ระหว่างWhole และ part บางครั้งถูกเรียกว่า "Has a" คือ Whole จะประกอบไปด้วย parts ต่างๆ ดังนั้นการคงอยู่ของ parts จะต้องขึ้นอยู่กับ Whole สามารถแทนด้วยสี่เหลี่ยมข้าวหลามตัดทึบติดอยู่ระหว่างปลายเส้นความสัมพันธ์กับคลาสอื่นมีหมายถึงสิ่งที่ใหญ่กว่า

ในการศึกษาครั้งนี้จะเลือกใช้ยูสเคสไดอะแกรม, คลาสไดอะแกรม และซีเควนไดอะแกรม เป็นหลักในการออกแบบระบบ เนื่องจาก ยูสเคสไดอะแกรมและคลาสไดอะแกรมถูกบังคับด้วยให้ใช้ด้วยการออกแบบด้วย UML อยู่แล้ว และสาเหตุที่เลือกใช้ ซีเควนไดอะแกรมอีกไดอะแกรมหนึ่ง เพราะซีเควนไดอะแกรมสามารถบอกกระบวนการทำงานของระบบได้ครบถ้วน

2.4.2 ภาษาเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุได้นำเอาส่วนที่ดีที่สุดของการโปรแกรมแบบโปรซีเยอร์มาพัฒนาร่วมกับแนวความคิดใหม่ๆ ที่ประยุกต์มาจากสิ่งที่เกิดขึ้นในชีวิตประจำวันของมนุษย์ ซึ่งผลที่ได้จะนำไปสู่แนวทางการเขียนโปรแกรมรูปแบบใหม่ที่มีประสิทธิภาพดีกว่าแบบเดิม โดยปกติแล้วการเขียนโปรแกรมเชิงวัตถุจะทำการแบ่งขอบเขตของงานที่ต้องการออกเป็นส่วนย่อยๆ ที่มีความสัมพันธ์กันรวมไว้ด้วยกันเป็นกลุ่มจากนั้นกลุ่มดังกล่าวจะถูกนำมาเรียงลำดับในรูปของโครงสร้างที่มีลักษณะเป็นลำดับชั้น เพื่อเพิ่มประสิทธิภาพในการเรียกใช้งานนั่นเอง โดยภาษาที่จะเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุได้นั้นต้องมีคุณสมบัติ 3 ข้อดังนี้

1. จะต้องเป็นภาษาที่ใช้ออบเจ็กต์ในการแก้ปัญหาเป็นหลัก เช่น ถ้าต้องการเขียนโปรแกรมติดต่อกับโมเด็ม หากเป็นการเขียนโปรแกรมแบบเก่าจะแบ่งฟังก์ชันการทำงานออกเป็นส่งรับหรือตรวจสอบข้อมูล แต่ถ้าเป็นภาษาเชิงวัตถุแล้วผู้เขียนต้องมองโมเด็มให้เป็นออบเจ็กต์ตัวหนึ่งโดยในออบเจ็กต์โมเด็มก็อาจประกอบไปด้วยออบเจ็กต์ย่อยๆ และออบเจ็กต์ย่อยเหล่านี้ก็จะมีการสื่อสารซึ่งกันและกัน

2. ออบเจ็กต์ที่ปรากฏในการเขียนโปรแกรมต้องเป็นอินสแตนซ์ของคลาส

3. คลาสสามารถรับทอดคุณสมบัติการใช้งานไปยังคลาสอื่นได้

ถ้าภาษาใดที่ขาดคุณสมบัติในสามข้อนี้ไปก็จะไม่เรียกภาษานั้นว่าเป็นภาษาในการเขียนโปรแกรมเชิงวัตถุ เช่นถ้าภาษาใดสนับสนุนการสร้างคลาสและออบเจ็กต์ แต่ไม่สนับสนุนการรับ

ทอดคุณสมบัติของคลาสแล้ว จะเรียกภาษานั้นว่า ภาษาที่ทำงานบนชนิดข้อมูลนามธรรม (Abstract Data Type) หรืออาจเรียกว่าภาษาในการเขียนโปรแกรมบนพื้นฐานของวัตถุ ไม่ใช่ภาษาในการเขียนโปรแกรมเชิงวัตถุ (อภิเนตร อูนากูล, 2543 หน้าที่ 18-19)

ภาษาที่สามารถเขียนเชิงวัตถุได้มีหลากหลายภาษาแต่ในการศึกษาครั้งนี้ผู้ศึกษาเลือกใช้ภาษา PHP ในการพัฒนา

2.5 แนวคิดแบบOpen source

ปัจจุบันการใช้โอเพนซอร์สซอฟต์แวร์กันมากมายและใช้งานได้ผลดี มีความเสถียรสูงมาก สิ่งนี้เป็นสิ่งที่ยืนยันได้ว่า โอเพนซอร์สซอฟต์แวร์เป็นทางเลือกที่เป็นไปได้ มิใช่เป็นเรื่องน่ากลัวเหมือนที่มีการกล่าวถึงกัน กระแสทางเลือกของโอเพนซอร์ส เป็นเส้นทางหนึ่งที่หลายองค์กรต้องเริ่มให้ความสนใจ โดยเฉพาะในบางประเทศได้เริ่มดำเนินการในระดับนโยบายของประเทศ เช่น ประเทศจีนได้มีนโยบายและการประกาศจุดยืนของตนเองเด่นชัดที่จะพัฒนาและนำแนวทางโอเพนซอร์สมาใช้

ริชาร์ด สตอลแมน เป็นผู้หนึ่งที่มีบทบาทที่สำคัญมากในเรื่องของคำว่าโอเพนซอร์ส เขาได้ก่อตั้งองค์กรหนึ่งชื่อ FSF - Free Software Foundation ในปี ค.ศ. 1983 ด้วยเหตุผลที่ว่า ธุรกิจซอฟต์แวร์ทำขึ้นเพื่อการค้าจะเป็นอุปสรรคใหญ่ต่อการพัฒนาซอฟต์แวร์ เพราะการพัฒนาที่ไม่เปิดเผยซอร์สโค้ด และการที่มีปัญหาเรื่องลิขสิทธิ์ต่างๆ ทำให้การพัฒนาซอฟต์แวร์ต้องติดขัดและขาดความก้าวหน้า เขาจึงเสนอรูปแบบการพัฒนาและลิขสิทธิ์ซอฟต์แวร์ที่เป็นแนวความคิดให้และให้ชื่อว่า GPL - General Public License โดยมีองค์กรที่เขาเป็นผู้ก่อตั้งให้มาเป็นผู้พิจารณาและดูแลในเรื่องกฎหมาย

เขาได้เริ่มโครงการโดยที่เอาซอฟต์แวร์มาเผยแพร่โดยใช้ชื่อโครงการว่า กะนู GNU (www.gnu.org) ซึ่งเป็นที่มาของซอฟต์แวร์ที่เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ต่อไปอีกหลายตัว และหนึ่งในที่รู้จักดีคือ GNU C ซึ่งเป็นคอมไพเลอร์ที่ใช้กันมาก

เริ่มแรกที่เขาใช้เรียกว่า ฟรีซอฟต์แวร์ (Free Software) แต่ทำให้คนเข้าใจไขว้เขวในเรื่องลิขสิทธิ์ ต่อมาเขาจึงเปลี่ยนเสียใหม่และใช้ชื่อว่า โอเพนซอร์ส (Open Source)

โดยที่คนส่วนใหญ่เข้าใจคำว่า ฟรี หมายถึง ไม่มีลิขสิทธิ์ซึ่งไม่ถูกต้อง และไม่ใช่แนวทางที่เขาต้องการ เพราะเขาให้ความหมายของคำว่า ฟรี คือ อิสระ เสรีภาพในการที่จะยินยอมให้ผู้อื่นใช้งาน ทำการคัดลอก แจกจ่าย และได้สิทธิ์อื่นๆ เช่น การพัฒนาต่อเนื่อง

GPL จึงมีลิขสิทธิ์และเป็นลิขสิทธิ์แบบ GPL ซึ่งเป็นลิขสิทธิ์ที่ถูกต้องตามกฎหมาย และมีกฎหมายรองรับ แต่มีข้อแตกต่างกับลิขสิทธิ์ซอฟต์แวร์ เพื่อการค้าอื่นคือ GPL ให้สิทธิแก่ผู้ใช้มากกว่า เพราะให้สิทธิ์ในการใช้ทำสำเนา ในการแจกจ่ายต่อ หรือได้ซอร์สโค้ดมาเพื่อพัฒนาแก้ไขดัดแปลง ไม่ว่าการกระทำพื้นๆ มีวัตถุประสงค์เพื่อการศึกษา หรือเพื่อธุรกิจเชิงพาณิชย์ แต่ได้เน้นข้อพิเศษไว้คือ ห้ามเปลี่ยนแปลงหลักการ เงื่อนไข หรือข้อตกลงใดๆ ที่กำหนดไว้ในซอฟต์แวร์ที่ใช้ลิขสิทธิ์แบบ GPL กล่าวคือ ถ้าผู้ใช้นำลิขสิทธิ์แบบ GPL ไปใช้ และพัฒนาต่อต้องคงลิขสิทธิ์แบบ GPL ไปใช้ และพัฒนาต่อ ต้องคงลิขสิทธิ์แบบ GPL ไว้ต่อไป ทั้งนี้นั้นเป็นข้อดีที่เป็นแนวความคิดที่ป้องกันไว้ในกรณีที่บุคคลต่อไปจะไปสร้างสิทธิหรือจำกัดสิทธิ์แก่ผู้อื่นที่เปลี่ยนไปจากเดิม

ทุกวันนี้เกือบทุกองค์กรเริ่มตระหนักและเห็นจริงแล้วว่า ค่าใช้จ่ายทางด้านซอฟต์แวร์แนวโน้มที่จะสูงขึ้น อีกทั้งวงจรชีวิตของซอฟต์แวร์ที่ใช้งานอยู่นี้มีวงจรชีวิตที่สั้นมาก ซอฟต์แวร์ที่จัดซื้อหามาด้วยราคาแพง มีวงจรชีวิตเพียงแค่ไม่เกินสองปีต้องอัพเกรดหรือปรับเปลี่ยนรุ่นใหม่ หรือต้องซื้อหาสิทธิ์มาใช้ใหม่ การดำเนินการในเรื่องการใช้ซอฟต์แวร์จึงเป็นต้นทุนที่เป็นปัญหาใหญ่ตามมา

องค์กรหลายองค์กรจึงต้องหันมาทบทวนและดูว่ามีการลดค่าใช้จ่ายในเรื่องเหล่านี้ได้อย่างไร หากพิจารณาที่เซิร์ฟเวอร์ที่ใช้กันบนอินเทอร์เน็ต ทั้งเมล์เซิร์ฟเวอร์ เว็บเซิร์ฟเวอร์ หรือเซิร์ฟเวอร์บริการอื่นๆ อีกมากมาย ซึ่งพบว่ามีการใช้โอเพนซอร์สซอฟต์แวร์กันมากมาย และใช้งานได้ผลดี มีความเสถียรสูงมาก สิ่งนี้เป็นสิ่งที่ยืนยันได้ว่า โอเพนซอร์สซอฟต์แวร์เป็นทางเลือกที่เป็นไปได้ มิใช่เป็นเรื่องน่ากลัวเหมือนที่มีการกล่าวถึงกัน (ศูนย์กลางความรู้แห่งชาติ, ม.ป.ป.)

ในการศึกษาครั้งนี้ผู้ดำเนินการศึกษา ได้เลือกใช้ซอฟแวร์ที่เป็น Open source มาใช้ในการพัฒนาโปรแกรมคือ

2.5.1 โปรแกรมในการจัดการฐานข้อมูล MySQL

2.5.2 ภาษาในการพัฒนาระบบ PHP

2.5.1 โปรแกรมในการจัดการฐานข้อมูล MySQL

โปรแกรมในการจัดการฐานข้อมูล MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ MySQL สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael "Monty" Widenius. รุ่นของผลิตภัณฑ์นั้นแบ่งออกมาได้สามสายการผลิต คือเวอร์ชันที่เป็นรุ่นที่ใช้ฟรี (คอมมิวนิตี้ เวอร์ชัน)

เวอร์ชันการค้า (คอมเมอร์เชียลเวอร์ชัน) และ เวอร์ชันที่สนับสนุนกับผลิต ภัณฑ์ SAP (MAX DB) ความแตกต่างคือเวอร์ชันคอมมิวนิตี้นั้นสามารถนำไปใช้งานได้ฟรีแต่ขาดการสนับสนุนหรือการช่วยเหลือเมื่อมีปัญหาเกิดขึ้น, เวอร์ชันที่เป็นคอมเมอร์เชียลนั้นให้บริการด้านความสนับสนุนเมื่อมีปัญหา (ซื้อบริการ) สรุปคร่าวๆ ประเภทดาต้าเบสให้เลือกใช้มี

- MySQL เอ็นเตอร์ไพรส์ Enterprise
- MySQL คลัสเตอร์ Cluster
- MySQL Embedded
- MySQL Community (open source เวอร์ชัน)

MySQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ phpBB และนิยมใช้งานร่วมกับภาษาโปรแกรม PHP ซึ่งมักจะได้ชื่อว่าเป็นคู่ จะเห็นได้จากคู่มือคอมพิวเตอร์ต่างๆ ที่จะสอนการใช้งาน MySQL และ PHP ควบคู่กันไป นอกจากนี้ หลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูล MySQL ซึ่งรวมถึง ภาษาซี ซีพลัสพลัส ปาสคาล ซีชาร์ป ภาษาจาวา ภาษาเพิร์ล พีเอชพี ไพทอน รูบี และภาษาอื่น ใช้งานผ่าน API สำหรับโปรแกรมที่ติดต่อผ่าน ODBC หรือ ส่วนเชื่อมต่อกับภาษาอื่น (database connector) เช่น เอเอสพี สามารถเรียกใช้ MySQL ผ่านทาง MyODBC,ADO,ADO.NET เป็นต้น (MySQL AB, ม.ป.ป.)

2.5.2 ภาษาในการพัฒนาระบบ PHP

PHP เป็นโปรแกรมประเภท Server Side Scripting มาจากคำว่า Personal Home Page PHP (Hypertext Preprocessor) สร้างจากภาษา Perl เพื่อติดต่อกับผู้ใช้หรือผู้เข้าชมเว็บเพจ Rasmus Lerdorfs online ช่วยจัดการระบบข้อมูล SQL ผู้ที่ได้เขียนขึ้นมาได้ทำการสอบถามจากคนที่เข้ารับบริการและผู้ที่เขียนเว็บไซต์ ในเวอร์ชัน 2.0 ออกแบบให้ทำงานร่วมกับ Html ได้ โดยได้ทำการเพิ่มส่วนของฟอร์ม ให้ผู้ใช้กดปุ่ม Submit และเข้าถึงฐานข้อมูลได้ง่ายกว่า เป็นแบบ PHP/FI และมีการพัฒนาอย่างต่อเนื่องและรวดเร็ว PHP นำมาใช้ในเว็บไซต์ทางธุรกิจในปี 1996 คือ www.supercuts.com ซึ่งใช้ในการสร้างฐานข้อมูลสำหรับลูกค้า ต่อมาเป็นการพัฒนา API สำหรับ PHP3 เป็นภาษาที่สมบูรณ์ เพราะได้รับการแก้ไขข้อบกพร่องจากรุ่นก่อนหน้านี้ มีการเขียนแบบกระจาย สนับสนุนการทำงานกับเซิร์ฟเวอร์ และออกเป็นเวอร์ชัน PHP4 ซึ่งการทำงานจะรวดเร็วกว่า PHP3 ปี 2542 มีเว็บไซต์ประมาณ 100,000 เว็บไซต์ที่ใช้ PHP ช่วยในการดำเนินธุรกิจบนเว็บของตน และ PHP ในปัจจุบันตั้งอยู่บนพื้นฐานของ PHP Hypertext Processor เพื่อเตรียมพร้อมกับความเปลี่ยนแปลงในอนาคต โค้ดโปรแกรมที่ใช้รันที่ฝั่งเซิร์ฟเวอร์ เช่น ASP, CGI, PHP ช่วยในการสร้างเว็บไซต์แบบไดนามิก เขียนสคริปต์ให้ติดต่อกับฐานข้อมูลได้ PHP รวมเอา

ความสามารถของ C, Java, Perl ไว้ด้วยกัน จากการสำรวจได้มีการลงความเห็นว่า ปัจจุบัน PHP เป็นโมดูลที่ได้รับความนิยมเป็นอย่างมากสำหรับการทำงานบน Apache server และมีจำนวนเว็บไซต์ที่ใช้ PHP และทำงานกับ Apache server ประมาณ 2 ล้านเว็บไซต์ จึงเป็นที่น่าสงสัยว่าทำไม PHP จึงได้รับความนิยมมาก และเป็นภาษาที่มีการทำงานแบบ Server Side Scripting (นิรุธ อำนวยศิลป์, 2544, หน้าที่ 51)

การเขียนเว็บเพจจะมีนามสกุล ซึ่งผู้เขียนสามารถกำหนดนามสกุลให้กับแต่ละเว็บเพจได้จากภาษาที่ใช้เขียน การเขียนเว็บเพจด้วยภาษาสคริปต์ php จะมีนามสกุลเป็น .php ซึ่งจะเป็นการบ่งบอกเพื่อติดต่อกับเว็บเซิร์ฟเวอร์ การติดตั้งในระบบปฏิบัติการ คือ Windows และ Linux ทำได้โดยจะมีการใช้โปรแกรมเว็บเซิร์ฟเวอร์ที่หลากหลาย ส่วนใหญ่จะเป็น PWS หรือ IIS ซึ่งเป็นของไมโครซอฟต์ อาจใช้เป็น Apache ต้องมีการติดตั้งไฟล์ที่เกี่ยวข้องกับการทำงานของ php คือ php-3_0_11-win32.zip ซึ่งใช้ได้กับเว็บเซิร์ฟเวอร์ที่หลากหลาย เมื่อดาวน์โหลดแล้วให้คลี่ไฟล์ลง C:\PHP3 และคัดลอกไฟล์ php3.ini-dist ไปเก็บไว้ที่ C:\WINDOWS ถ้าเป็น PWS หรือ IIS ที่ไม่ใช่ IIS เวอร์ชัน 4 จะต้องแก้ registry จึงจะใช้งาน php ร่วมกับเว็บเซิร์ฟเวอร์ได้ ถ้าใช้ Apache ไม่ต้องแก้ register พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ ออราเคิล DBase PostgreSQL IBM DB2 MySQL Informix ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย คุณสามารถเชื่อมต่อกับฐานข้อมูลต่างๆ ที่รองรับมาตรฐานโลกนี้ได้ (กิตติภูมิ วรฉัตร, 2543)

พีเอชพียังสามารถรองรับการสื่อสารกับการบริการในโปรโตคอลต่างๆ เช่น LDAP IMAP SNMP NNTP POP3 HTTP COM (บนวินโดวส์) และอื่นๆ อีกมากมาย คุณสามารถเปิด Socket บนเครือข่ายโดยตรง และ ตอบโต้โดยใช้ โปรโตคอลใดๆ ก็ได้ PHP มีการรองรับสำหรับการแลกเปลี่ยนข้อมูลแบบ WDDX Complex กับ Web Programming อื่นๆ ทั่วไปได้ พูดถึงในส่วน Interconnection, พีเอชพีมีการรองรับสำหรับ Java objects ให้เปลี่ยนมันเป็น PHP Object แล้วใช้งาน คุณยังสามารถใช้รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้เช่นกัน (สมศักดิ์ โชคชัยชุติกุล, 2547)

ในการศึกษาครั้งนี้ผู้ศึกษา ได้เลือกใช้ MySQL เป็นโปรแกรมในการจัดการฐานข้อมูล และ PHP เป็นภาษาในการพัฒนาระบบ เพื่อให้เป็นไปแนวทางเดียวกับคณะวิทยาศาสตร์

2.6 งานวิจัยที่เกี่ยวข้อง

Sueychyun (Roger) Fang (2548) ได้ทำการวิจัยเกี่ยวกับ ระบบการจัดตารางเรียนของมหาวิทยาลัยโดยใช้ UML (UNIVERSITY COURSE SCHEDULING SYSTEM: UCSS-A UML APPLICATION WITH DATABASE AND VISUAL PROGRAMMING) ได้กล่าวว่า ตารางเรียนของมหาวิทยาลัยมีขนาดใหญ่ ซับซ้อน และเป็นงานที่ต้องเสียเวลามาก ช่วงเวลาในห้องเรียนจะถูกแบ่งเป็นช่วงเพื่อจะจัดสรรให้ผู้สอนได้ ห้องเรียนเหล่านี้ของภาควิชาต่างๆ จะไม่กระทบกับกฎที่กำหนดไว้ล่วงหน้าแล้ว งานวิจัยนี้ใช้ UML ช่วยในการออกแบบเพื่อแก้ปัญหาตารางเวลาของผู้ใช้ในทรัพยากรที่จำกัด ระบบพัฒนาด้วย Visual Basic .NET และ SQL Server 2000 เมื่อพัฒนาเสร็จจะนำไปใช้เพื่อสนับสนุน 2 ตารางเรียน ใน Arkansas Tech University คณะผู้ทำการศึกษาได้ออกแบบ โครงสร้างของระบบตารางเรียนของมหาวิทยาลัยนั้นพัฒนาใน Arkansas Tech University โดยใช้ UML ในการออกแบบซึ่งประกอบด้วย use case diagram, activity diagram, package diagram, class diagram, statechart diagram และ deployment diagram ในการเริ่มการจัดตารางเรียนคณะผู้ทำการศึกษาไม่ได้ตั้งเป้าหมายว่าจะเป็นระบบสร้างตารางเรียนอัตโนมัติแต่เน้นการตัดสินใจบนข้อจำกัด คณะผู้ทำการศึกษาได้ใช้ UCSS ด้วยวัตถุประสงค์สองอย่างคือ ปรับปรุงกระบวนการทำด้วยมือและระบบที่สมบูรณ์สำหรับการสอน

M. Dimopoulou และคณะ (2546) ได้ทำการวิจัยเกี่ยวกับระบบจัดตารางเรียนของมหาวิทยาลัยอัตโนมัติ โดยการพัฒนาแบบกระจาย (An automated university course timetabling system develop in a distributed environment: A case study) ในการศึกษานี้คณะผู้ศึกษาได้ทดสอบการออกแบบและการนำระบบเครือข่ายคอมพิวเตอร์มาช่วยจัดตารางเรียนของมหาวิทยาลัย เสนอระบบที่ใช้ฐานข้อมูลรวมกัน เพื่อความเหมาะสมของสารสนเทศและการทำการคำนวณ ความแตกต่างของแต่ละหน่วยงานในมหาวิทยาลัยได้รวบรวมความเหมาะสมของฐานข้อมูลในการซึ่งและปรับปรุงข้อมูลของตนเอง นอกจากนั้น การปรับปรุงฐานข้อมูลเป็นกระบวนการโดยผู้จัดตารางเวลาเพื่อสร้างตารางเวลาของแต่ละหน่วยงานซึ่งแยกออกจากกัน ซึ่งจะเหมาะสมของระบบเครือข่าย รูปแบบการร่วมกันของตารางเรียนทำให้การค้นหาง่าย ระบบจะใช้รูปแบบการเขียนโปรแกรมแบบตัวเลขเต็มหน่วย ในการกำหนดวิชาลงช่วงเวลาและห้องเรียนของตารางเรียนแต่ละหน่วยงาน วิธีการทำโดยอัตโนมัติเป็นการทำให้เกิดความเชื่อมโยงรูปแบบการเขียนโปรแกรมแบบตัวเลขเต็มหน่วยด้วยข้อมูลของแต่ละหน่วยงานและแก้ไขปัญหาความขัดแย่งของแต่ละหน่วยงานโดยการกระจายกระบวนการทำงาน รูปแบบตัวอย่างนี้เป็นการช่วยเหลือให้มีความยืดหยุ่น โดยเฉพาะการสร้างข้อจำกัดให้เหมือนกับข้อสันนิษฐานของผู้ใช้และผู้ทำ

รายงาน ทำให้ง่ายในการแสดงผลการจัดตารางเวลา มีความยืดหยุ่นและมีการทดสอบตารางเวลาอื่นๆ ซึ่งเป็นเงื่อนไขตามความต้องการของผู้ใช้ และภายใต้เงื่อนไขของ Athens University of Economics and Business

รังสรรค์ เนียมสนิท และคณะ (2528) ได้ทำการวิจัยเกี่ยวกับการวิเคราะห์รูปแบบที่เหมาะสมในการจัดตารางเรียนตารางสอนสำหรับวิชาพื้นฐานของคณะวิทยาศาสตร์ ที่ให้บริการแก่คณะต่างๆ ในมหาวิทยาลัยขอนแก่น ได้ทำการวิเคราะห์รูปแบบที่เหมาะสมในการจัดต่างเรียนตารางสอน พบว่ามีรูปแบบมูลฐานอยู่ 2 รูปแบบที่สามารถนำไปใช้และประยุกต์ใช้ได้ในอนาคต คณะผู้ศึกษาคาดว่าจะได้ประโยชน์ซึ่งเป็นส่วนหนึ่งในการช่วยให้การจัดตารางเรียนตารางสอนของมหาวิทยาลัยขอนแก่นเป็นไปด้วยดีรวมทั้ง ทำให้ได้ข้อมูลพื้นฐานและทราบรูปแบบที่เหมาะสมของการจัดตารางเรียน ตารางสอน ซึ่งจะเป็นข้อมูลสำหรับการว่างแผนระบบการศึกษา โดยการปฏิบัติในรูปแบบของกรรมการร่วมระหว่างผู้แทนของคณะวิทยาศาสตร์ ผู้แทนจากคณะต่างๆ กับเจ้าหน้าที่ของกองบริการการศึกษาในการปฏิบัติงานระยะเริ่มแรก