บทที่ 2
ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ระบบสารสนเทศ

2.1.1.1 นิยาม

ชุมพล ศฤงคารศิริ (2538: 55) ให้ความหมายของคำว่าสารสนเทศไว้ว่า สารสนเทศ คือ ข้อมูลที่ได้ผ่านการประมวลผล และถูกจัดให้อยู่ในรูปที่มีความหมาย และเป็นประโยชน์ ต่อการตัดสินใจของผู้รับ (recipient)

ในระบบสารสนเทศ จะมีคำนิยามที่ใช้อยู่โดยทั่วไป คือ ข้อมูล สารสนเทศ และระบบสารสนเทศ (ณัฏฐพันธ์ เขจรนันทร์ และไพบูลย์ เกียรติโกมล, 2542)

ข้อมูล (Data) หมายถึง ข้อเท็จจริงต่างๆ ที่มีอยู่ในธรรมชาติของสิ่งที่ได้รับการสนใจ ไม่ว่าจะเป็นบุคคล สัตว์ ผลิตภัณฑ์ สถานการณ์ เหตุการณ์ หรือ อื่นๆ โดยอาจจะอยู่ในรูปแบบที่เป็นตัวเลข ข้อความ หรือรายละเอียดในรูปแบบต่างๆ ซึ่งใช้แทนข้อเท็จจริงนั้น

สารสนเทศ (Information) หมายถึง เรื่องราวต่าง ๆ ที่ได้จากการนำข้อมูลมาประมวลผล ด้วยวิธีการใดๆ ให้เกิดเป็นความรู้ที่ต้องการสำหรับนำไปใช้ประโยชน์ ทั้งนี้ความสัมพันธ์ระหว่าง ข้อมูล และสารสนเทศ

ระบบสารสนเทศ (Information System) หมายถึง ระบบที่ประกอบด้วยคน เครื่องคอมพิวเตอร์ และอุปกรณ์ต่างๆ ที่ทำงานประสานกัน เพื่อจัดทำสารสนเทศสำหรับสนับสนุนการปฏิบัติงาน การจัดการ และการตัดสินใจในหน่วยงานหรือองค์กร

2.1.1.2 ขอบข่ายความรู้ที่จำเป็น (A Necessary Knowledge Framework)

เพื่อทำความเข้าใจระบบสารสนเทศ รวมถึงรู้จักวิธีการนำเอาไปประยุกต์ใช้นั้น ผู้ใช้ระบบสารสนเทศจำเป็นต้องมีความรู้ความเข้าใจในเรื่องเหล่านี้ คือ :

1. รากฐานแนวความคิด (Foundation Concepts) หมายถึงพื้นฐานพฤติกรรม หรือแนวความคิดเกี่ยวกับเทคนิค เพื่อที่จะทำให้มีความเข้าใจว่า ระบบสารสนเทศจะช่วยสนับสนุนการปฏิบัติการ, การบริหารการตัดสินใจ, การใช้กลยุทธ์ทางธุรกิจ ให้ประสบผลสำเร็จได้อย่างไร
2. เทคโนโลยี (Technology) เรื่องหลักของแนวความคิดนี้คือ เรื่องของการพัฒนาและการจัดการเทคโนโลยีสารสนเทศ ซึ่งนั่นก็คือ ต้องมีความรู้เกี่ยวกับฮาร์ดแวร์, ซอฟท์แวร์, เครือข่ายคอมพิวเตอร์, การจัดการฐานข้อมูล หรือเทคโนโลยีการประมวลผลข้อมูลต่างๆ
3. การประยุกต์ (Application) การประยุกต์ในความหมายนี้คือ หลักการใช้ระบบสารสนเทศเพื่อนำไปปฏิบัติการ, บริหารจัดการ, และใช้เพื่อเป็นข้อได้เปรียบในการแข่งขันขององค์กร ซึ่งก็รวมทั้ง พาณิชย์อิเล็กทรอนิกส์ และการใช้ Internet, Intranet, Extranet ในองค์กร
4. การพัฒนา (Development) หมายถึงว่า ทำอย่างไรที่ผู้ใช้ หรือผู้เชี่ยวชาญสารสนเทศนี้ จะสามารถพัฒนาเงื่อนไขของระบบสารสนเทศไปช่วยในการแก้ปัญหาทางธุรกิจได้ และนอกจากนี้ยังหมายถึงทำอย่างไรที่จะพัฒนาทฤษฎีใหม่ๆ ขึ้นมาใช้ได้
5. การจัดการ (Management) หมายถึงประสิทธิภาพในการจัดการเรื่องทรัพยากร และการใช้กลยุทธ์ทางธุรกิจ เพื่อนำมาจัดการเกี่ยวกับเทคโนโลยีสารสนเทศในองค์กร รวมถึงประเภทของธุรกิจในแต่ละระดับด้วย

2.1.1.3 กิจกรรมระบบสารสนเทศ (Information Systems Activities)

กิจกรรมของระบบสารสนเทศ ได้แก่ กระบวนการต่างๆ ที่เกิดขึ้นในระบบสารสนเทศ ซึ่งมีตั้งแต่การนำเข้าข้อมูล, การประมวลผล, การนำออกข้อมูล, การจัดเก็บ และการควบคุมกิจกรรมต่างๆ ซึ่งสามารถจำแนกรายละเอียดได้ดังนี้ คือ

1. หน่วยนำเข้าของทรัพยากรข้อมูล (Input of Data Resources) หมายถึงการนำเอาข้อมูลต่างๆ ป้อนเข้าไปสู่คอมพิวเตอร์ส่วนใหญ่เป็นข้อมูลการประมวลผลรายการ หรือบางครั้งอาจจะเป็นการส่งข้อมูลจากแผ่นดิสก์เก็ตเข้าสู่เครื่องเพื่อทำการประมวลผล ข้อมูลที่ป้อนเข้าเหล่านั้นจะเกี่ยวข้องกับประมวลผลการขาย ซึ่งถูกจัดเก็บเอาไว้ในลักษณะเอกสารต้นฉบับ (Source document) เช่น แบบฟอร์มการขาย-การสั่งซื้อ เป็นต้น

2. การประมวลผลข้อมูลสารสนเทศ (Processing of Data into Information) ข้อมูลเป็นสิ่งจำเป็นที่ป้อนเข้าไปเพื่อทำการประมวลผล ในหลายลักษณะ เช่น คำนวณ เปรียบเทียบ ค้นหา แยกแยะ และสรุป กิจกรรมเหล่านี้ต้องการเปลี่ยนแปลงให้เป็นสารสนเทศที่ดีสำหรับผู้ใช้ คุณภาพและการจัดเก็บของระบบสารสนเทศนั้น จะต้องมีการดูแลบำรุงรักษา ตลอดจนปรับปรุงให้เป็นข้อมูลที่มีความทันสมัยอยู่เสมอ ยกตัวอย่างเช่น ข้อมูลที่รับเข้ามาซึ่งเกี่ยวกับการจัดซื้อจะต้องมีการเพิ่ม (Added), เปรียบเทียบ (Compared), จัดเก็บ (Stored), แยกแยะ (Classified), สรุป (Summarized) และปรับปรุง (Update)
3. หน่วยนำออกของการผลิตภัณฑ์สารสนเทศ (Output of Information Products) ได้แก่ สารสนเทศประเภทต่างๆ ที่ถูกส่งมายังผู้ใช้ เป้าหมายของระบบสารสนเทศคือการผลิตสารสนเทศหรือจัดทำผลิตภัณฑ์สารสนเทศ (Information Products) ให้มีความสะดวกเหมาะสมกับผู้ใช้ ผลิตภัณฑ์สารสนเทศที่รู้จักกันโดยทั่วไป คือ ข้อความ, รายงาน, แบบฟอร์ม, ภาพกราฟิกส์ หรืออาจแสดงออกมาในลักษณะของวิดีโอ, เสียงเพลงที่ออกมาจากลำโพง, ผลิตภัณฑ์กระดาษ และมัลติมีเดีย ส่วนอีกเรื่องหนึ่งที่ต้องคำนึง คือ คุณภาพสารสนเทศ (Information Quality) ต้องมีความถูกต้องแน่นอน เชื่อถือได้ เป็นประโยชน์ และเข้าใจไม่ยากสำหรับผู้ใช้
4. การจัดเก็บทรัพยากรข้อมูล (Storage of Data Resources) การจัดเก็บข้อมูลทางคอมพิวเตอร์นั้น จะมีการจัดเก็บข้อมูลเป็นกลุ่มของตัวอักษร (Field), ระเบียน (record), แฟ้มข้อมูล (File), และฐานข้อมูล (Database) ทั้งนี้ เพื่อให้เกิดความสะดวกต่อการประมวลผล, และการเรียกออกมาใช้เมื่อถึงคราวจำเป็น
5. การควบคุมระบบการทำงาน (Control of System Performance) นั่นคือมีกระบวนการควบคุมการปฏิบัติงานให้ดีนั่นเอง เพื่อให้ผลย้อนกลับ (Feedback) ออกมาในทางที่ดี ผลย้อนกลับนี้ ช่วยในเรื่องของการติดตามงาน, การประเมินผลความต้องการได้ดี เช่น ผู้จัดการฝ่ายขายทราบว่ายอดขายลดลง เขาอาจใช้วิธีการป้อนข้อมูลเข้าไป และทำการประมวลผลเพื่อให้ทราบตัวเลขที่แน่นอนและนำมาช่วยติดตามการขายให้ก้าวหน้าต่อไป

2.1.1.4 ประเภทของระบบสารสนเทศ (Type of Information Systems)

ระบบสารสนเทศที่ใช้กันอยู่ทั่วโลกนั้น สามารถจำแนกได้เป็นหลายๆ ชนิด แตกต่างกัน ไม่ว่าจะเป็นระบบสารสนเทศสำหรับการปฏิบัติการ หรือระบบสารสนเทศเพื่อการจัดการ ซึ่งจุดประสงค์หลักของระบบสารสนเทศมี 2 หลักการใหญ่คือ สนับสนุนการปฏิบัติการทางธุรกิจ และสนับสนุนการจัดการดังมีรายละเอียดดังต่อไปนี้

1. ระบบสนับสนุนการปฏิบัติการ (Operations Support Systems)

ระบบสนับสนุนการปฏิบัติการนี้ เป็นระบบที่ผลิตระบบสารสนเทศต่างๆ สำหรับใช้เกี่ยวกับผลิตภัณฑ์ต่างๆ ทั้งภายในและภายนอก ซึ่งบทบาทก็คือ เพื่อประมวลผลรายการอำนวยความสะดวกแก่กระบวนการทางธุรกิจ, ควบคุมกระบวนการในโรงงานอุตสาหกรรม, สนับสนุนการสื่อสารภายในองค์กร, การเขียนเอกสารร่วมกัน และปรับปรุงฐานข้อมูลในองค์กร โดยสามารถแจงรายละเอียดได้อีกดังนี้ คือ

- *ระบบประมวลผลรายการ (Transaction Processing System : TPS)* เป็น ระบบสนับสนุนการปฏิบัติงานประจำวัน ซึ่งได้แก่ กระบวนการขาย (Process sales), การสั่งซื้อ (Purchases), และการเปลี่ยนแปลงสินค้าคงคลัง (Inventory changes) สำหรับตัวแบบของการประมวลผลรายการนั้นมี 5 ระบบได้แก่ (1) การประมวลผล, (2) เครื่องจักรกลคอมพิวเตอร์, (3) อุปกรณ์ที่ต่างชนิดกัน, (4) ข้อมูล และ (5) องค์ประกอบของขนาดองค์กรการทำงานที่อาจต่างกันได้ ส่วนระบบประมวลผลรายการแยกย่อยออกเป็น 2 อย่าง คือ
    - การประมวลผลแบบชุดหรืองวด (Batch Processing) เป็นการเก็บรวบรวมข้อมูลเอาไว้เป็นชุดหรือเป็นงวด แล้วนำมาประมวลผลพร้อมกันทีเดียว เช่นระบบเงินเดือน
    - การประมวลผลตามเวลาจริง หรือออนไลน์ (Real-time or Online Processing) เป็นการประมวลผลที่สามารถประมวลผลข้อมูลออกมาได้ในทันทีทันใด หลังจากมีการประมวลผลรายการเกิดขึ้น เช่น การประมวลผล ณ จุดขาย (Point of sale : POS) จะเห็นได้ตามร้านอาหาร Fast Food เช่น KFC, Pizza Hut เป็นต้น หรือ ระบบ ATM
- *ระบบควบคุมการประมวลผล (Process Control Systems)* เพื่ออำนวยความสะดวกและรวดเร็วในการทำงาน ตัวอย่างของระบบนี้ เช่น โรงงานกลั่นน้ำมันปิโตรเลี่ยม ใช้

อิเล็กทรอนิกส์เซ็นเซอร์ (Electronic sensors) เชื่อมโยงไปยังเครื่องคอมพิวเตอร์เพื่อติดตามดูกระบวนการทางเคมี เป็นต้น

- *ระบบการเขียนเอกสารร่วมกันในองค์กร(Enterprise Collaboration Systems)* ระบบนี้ช่วยให้คนที่ทำงานร่วมกันในองค์กรสามารถเขียนเอกสารร่วมกันหรือทำงานร่วมกันได้ นอกจากนี้ยังช่วยในเรื่องของการสื่อสารทางความคิด, การใช้ทรัพยากรร่วมกัน, การประสานความร่วมมือกันระหว่างสมาชิกในกลุ่มทั้งในกระบวนการที่เป็นทางการและไม่เป็นทางการ ซึ่งส่วนใหญ่มักเป็นงานโครงการ (Project) ตัวอย่างเช่น กระบวนการทางธุรกิจที่มีกลุ่มของวิศวกร,ผู้เชี่ยวด้านการตลาด, และคนที่มีความรู้สูงช่วยกันพัฒนาผลิตภัณฑ์ใหม่ขึ้นมา กรณีอย่างนี้ อาจเรียกอีกอย่างหนึ่งว่า กลุ่มทำงานเสมือนจริง (Virtual team)

2. ระบบสนับสนุนการจัดการ(Management Support Systems)

ระบบสนับสนุนการจัดการ สามารถเรียกเรียกอีกอย่างหนึ่งว่า ระบบสารสนเทศเพื่อการบริหาร (Management Information System) หรือ MIS คือ ระบบที่มีการจัดอย่างเป็นระเบียบ และรวมเข้าเป็นกลุ่มโครงสร้างที่ ประกอบขึ้นมาจากบุคคลจำนวนมาก เครื่องมือ และระเบียบวิธีการต่าง ๆ ที่ช่วยให้มีข้อมูลที่ถูกต้องทั้งจากแหล่งภายในและภายนอก กล่าวคือ ระบบสารสนเทศเพื่อการบริหารเป็นระบบที่รวม (Integrate) ผู้ใช้และเครื่อง (User-Machine) เข้าไว้ด้วยกัน โดยทำหน้าที่ในการจัดหาสารสนเทศ หรือข่าวสารเพื่อช่วยในการตัดสินใจของผู้บริหารในเรื่องของกระบวนการจัดการองค์กร เช่น การวางแผน การจัดองค์กร และการควบคุม เพื่อให้องค์กรสามารถดำเนินการไปตามวัตถุประสงค์ที่ตั้งไว้

ระบบสารสนเทศเพื่อการบริหารจะต้องมีการประสานร่วมกับหน่วยงานหรือระบบย่อยอื่นๆ ในองค์กร โดยมีลักษณะการจัดตั้งที่เป็นระบบ และง่ายแก่การประสานงานกับระบบย่อยอื่นๆ ในองค์กรด้วย

ระบบสารสนเทศเพื่อการบริหารสามารถดำเนินการได้โดยไม่ต้องอาศัยคอมพิวเตอร์เข้ามาช่วย แต่เนื่องจากความสามารถของคอมพิวเตอร์ ในอันที่จะประมวลผลข้อมูลได้จำนวนมากในเวลาอันรวดเร็ว ดังนั้นในปัจจุบันระบบสารสนเทศเพื่อการบริหารจึงมักจะผ่านกระบวนการประมวลผลด้วยคอมพิวเตอร์

หน้าที่หลักของสารสนเทศเพื่อการบริหาร ประกอบด้วย

- ให้สารสนเทศเพื่อช่วยในการตัดสินใจของผู้บริหารได้
- ให้สารสนเทศแก่ผู้บริหารทุกระดับได้
- ให้สารสนเทศเพื่อช่วยในการแก้ไขปัญหาทุกรูปแบบของปัญหา
- ให้สารสนเทศที่รวดเร็วและเหมาะสมกับการใช้งาน

ประโยชน์ที่ผู้บริหารจะได้รับจากระบบสารสนเทศเพื่อการบริหาร สามารถสรุปได้ดังนี้

- ช่วยให้ผู้บริหารมองเห็นปัญหาและโอกาสได้รวดเร็วขึ้น
- ช่วยให้ผู้บริหารมีเวลาสำหรับการวางแผนได้มากขึ้น
- ช่วยให้ผู้บริหารใช้เวลาในการพิจารณาปัญหาที่มีความซับซ้อนได้มากขึ้น
- ช่วยให้ผู้บริหารควบคุมการดำเนินการได้ดีขึ้น

ระบบสารสนเทศเพื่อการบริหารนั้น แม้จะสร้างขึ้นให้กับผู้บริหารใช้ก็จริงอยู่ แต่ผลลัพธ์ของระบบ หรือรายงานที่จะจัดทำให้ผู้บริหารแต่ละระดับนั้นมีความแตกต่างกัน เพราะขึ้นอยู่กับหน้าที่ของผู้บริหารแต่ละคนซึ่งจะบังคับให้ต้องการสารสนเทศที่ต่างกัน สารสนเทศที่ต้องใช้จึงมักจะเป็นสารสนเทศที่เกี่ยวกับสภาพของตลาดและสถานการณ์ภายนอกบริษัทมากกว่าจะเป็นสารสนเทศจากภายในบริษัท ในทางตรงกันข้าม ผู้บริหารระดับล่างซึ่งต้องควบคุมการปฏิบัติงานภายในให้ดำเนินไปตามเป้าหมายและวัตถุประสงค์ที่วางไว้ก็ต้องการสารสนเทศจากภายในมากกว่าภายนอก ดังแสดงได้ตารางข้างล่าง

ตารางที่ 2.1 ตารางแสดงความสัมพันธ์ของระดับการบริหาร และคุณลักษณะสารสนเทศที่ต้องการ

ผู้บริหาร	คุณลักษณะสารสนเทศ
ระดับสูง	มาจากภายนอกเกินกว่าครึ่ง เป็นสารสนเทศสรุปแสดงแนวโน้มระยะยาว ไม่จำเป็นต้องเป็นปัจจุบัน
ระดับกลาง	มาจากภายนอกประมาณครึ่ง เป็นข้อมูลและสารสนเทศสรุปแนวโน้มระยะสั้น ควรเป็นสารสนเทศปัจจุบัน
ระดับล่าง	มาจากภายในเป็นส่วนใหญ่ เป็นข้อมูลแสดงรายละเอียด เป็นเรื่องปัจจุบัน

การออกแบบระบบสารสนเทศเพื่อการบริหาร เป็นการจัดวางระบบสารสนเทศเพื่อการบริหารใหม่ทั้งหมด หรือเป็นการปรับปรุงระบบสารสนเทศเพื่อการบริหารเดิมเพียงบางส่วน โดยการออกแบบนี้จะขึ้นกับผลที่ได้จากการศึกษาและวิเคราะห์ระบบสารสนเทศเพื่อการบริหารเดิม และผลการตัดสินใจของผู้บริหารว่าต้องการระบบสารสนเทศเพื่อการบริหารใหม่เป็นอย่างไร ทั้งนี้กระบวนการดังกล่าว จะประกอบด้วยขั้นตอนที่สำคัญดังนี้

1) การออกแบบรายงาน
2) การออกแบบข้อมูลเพื่อนำเข้าระบบประมวลผล
3) การออกแบบระบบประมวลผล

การออกแบบรายงาน รายงานเป็นส่วนที่สำคัญสำหรับผู้บริหารที่จะไปใช้ประโยชน์ ดังนั้น ถ้ารายงานเป็นไปตามความต้องการของผู้บริหารแล้ว ก็ถือได้ว่าระบบที่ออกแบบบรรลุเป้าหมายไปได้ส่วนหนึ่ง สำหรับขั้นตอนโดยละเอียดของการออกแบบรายงานจะประกอบด้วย

- การกำหนดรายงานที่ต้องการ เป็นการกำหนดถึงรายงานที่ต้องการจากระบบ โดยนำผลจากขั้นตอนการศึกษาและวิเคราะห์ระบบมาทบทวนและพิจารณาร่วมกับความต้องการของผู้บริหารและผู้ปฏิบัติงาน หลักที่ใช้ในการพิจารณารายงานที่ต้องการจากระบบ ได้แก่ รายงานนั้นยังมีความต้องการหรือไม่รายงานนั้นมีความซ้ำซ้อนกับรายงานอื่นๆหรือไม่
- การกำหนดสารสนเทศในรายงาน ภายหลังจากที่ได้มีการกำหนดรายงานที่ต้องการแล้ว จะต้องมีการวิเคราะห์ร่วมกับผู้บริหารและผู้ปฏิบัติงาน เพื่อกำหนดรายละเอียดของสารสนเทศที่ต้องการในรายงาน
- การออกแบบรูปแบบรายงาน จะกระทำภายหลังจากที่ได้กำหนดรายละเอียดของสารสนเทศในรายงานแล้ว รูปแบบรายงานเหล่านี้จะแบ่งออกเป็นรายงานที่ใช้ภายในหน่วยงานและรายงานที่ใช้ภายนอกหน่วยงาน โดยรายงานที่ใช้ภายในหน่วยงานเป็นรายงานที่ใช้ในการปฏิบัติงานประจำ จึงมีรูปแบบที่เป็นไปตามความพอใจของหน่วยงานเอง ในขณะที่รายงานที่ใช้ภายนอกหน่วยงานจะมีรูปแบบที่ต้องคำนึงถึงวัตถุประสงค์ของผู้บริหารหน่วยงานต่าง ๆ ที่นำไปใช้ด้วย
- การจัดระบบในการออกรายงานนอกเหนือจากการออกแบบรูปแบบรายงานแล้ว จะต้องคำนึงถึงระบบในการออกรายงานด้วย เช่น จำนวนชุดของรายงานที่ต้องการ การไหลของรายงานถึงผู้รับสารสนเทศ และความถี่ในการออกรายงาน เป็นต้น

การออกแบบข้อมูลเพื่อนำเข้าระบบประมวลผล เป็นการพิจารณาลักษณะข้อมูลที่นำเข้าสู่ระบบประมวลผล เพื่อให้ได้รายงานจากระบบตามที่ต้องการ ซึ่งในขั้นตอนนี้มีสิ่งที่ต้องพิจารณาดังนี้

- ข้อมูลนำเข้าที่ต้องการ การพิจารณาว่าข้อมูลนำเข้าควรเป็นอะไรบ้าง ขึ้นกับรายงานที่ได้ออกแบบไว้ ซึ่งงานในขั้นตอนนี้จะนำเอาผลการวิเคราะห์รายงานที่ได้ออกแบบไว้มาพิจารณาถึงชนิด และขนาดของข้อมูลที่ใช้เป็นข้อมูลนำเข้า
- แหล่งข้อมูลนำเข้า ในการวิเคราะห์จำเป็นต้องหาแหล่งข้อมูลที่จำเป็นต้องใช้เพื่อกำหนดแหล่งข้อมูลนำเข้าของระบบ ทั้งนี้แหล่งข้อมูลที่ใช้ดังกล่าวเพื่อจัดทำรายงานอาจแบ่งออกได้เป็น
    - แหล่งข้อมูลจากเอกสารชิ้นเดียวกัน การใช้แหล่งข้อมูลนี้จะไม่ยุ่งยากมาก เนื่องจากข้อมูลทั้งหมดมาจากเอกสารในชิ้นเดียวกัน
    - แหล่งข้อมูลที่เกิดจากการคำนวณ บางรายงานอาจมีข้อมูลที่มาจากแหล่งเดียว และข้อมูลบางส่วนได้มาจากการนำข้อมูลไปทำการคำนวณ
    - แหล่งข้อมูลหลายแหล่ง ลักษณะแหล่งข้อมูลแบบนี้จะทำให้เกิดความยุ่งยากในการออกแบบระบบสารสนเทศ เนื่องจากข้อมูลที่นำเข้าจะมีหลายแบบ
    - แหล่งข้อมูลจากตารางที่ได้กำหนดขึ้น เป็นการกำหนดค่าไว้เป็นตารางอ้างอิง และนำมาประมวลผล ซึ่งเป็นวิธีการที่นิยมใช้กันโดยทั่วไป เนื่องจากเป็นการสรุปข้อมูลในรูปแบบที่สามารถนำเสนอได้ง่าย และการเตรียมข้อมูลนำเข้าก็สะดวก
- การกำหนดระยะเวลาของข้อมูลนำเข้า เป็นการกำหนดระยะเวลาและความถี่ของข้อมูลนำเข้า ทั้งนี้เพื่อให้ทันต่อความต้องการใช้ในการประมวลผลให้ได้เป็นรายงานตามที่ต้องการ

การออกแบบระบบประมวลผลจะครอบคลุมตั้งแต่ การเก็บรวบรวมข้อมูล การจดบันทึก การเก็บรักษา การคำนวณ การประมวลผล การวิเคราะห์และการเรียกกลับมาใช้ในภายหลัง ทั้งนี้เพื่อที่จะประมวลผลข้อมูลให้ได้เป็นสารสนเทศและรายงานตามที่ต้องการ

2.1.1.5 เป้าหมายของระบบสารสนเทศ

ระบบสารสนเทศสำหรับองค์กรต่าง ๆ โดยส่วนใหญ่แล้วมักมีเป้าหมายที่สำคัญ (ประสงค์ ปราณีตพลกรัง และคณะ, 2541) ดังนี้

1. เพิ่มประสิทธิภาพในการทำงาน (Operational Efficiency)
2. เพิ่มประสิทธิภาพของหน้าที่งาน (Functional Effectiveness)
3. เพิ่มคุณประโยชน์ในเชิงการแข่งขัน (Competitive Advantage)

การเพิ่มประสิทธิภาพในการทำงาน (Operational Efficiency) เป็นการช่วยให้งานที่ทำอยู่นั้นสามารถทำได้เร็วขึ้น มีความถูกต้องมากขึ้น ทำให้พนักงานมีเวลาในการเรียนรู้งานใหม่ ๆ เพื่อเพิ่มประสิทธิภาพในการทำงาน ลักษณะที่เห็นได้ คือ เป็นการทำสิ่งที่มีอยู่ให้ดีขึ้น (Do things better)

การเพิ่มประสิทธิภาพของหน้าที่งาน (Functional Effectiveness) เป็นการช่วยให้ผู้บริหารมีมุมมองที่มากขึ้นและกว้างขึ้น ได้รับทราบถึงข้อมูลที่หลากหลาย ช่วยในการตัดสินใจ รวมทั้งสามารถบริหารควบคุมหน่วยงานได้ดีขึ้น ลักษณะที่เห็นได้ คือ เป็นการทำในสิ่งที่ดีกว่า (Do better things)

การเพิ่มคุณประโยชน์ในเชิงการแข่งขัน (Competitive Advantage) เป็นการสร้างความได้เปรียบในการแข่งขันเมื่อเทียบกับคู่แข่ง ไม่ว่าจะเป็นในเรื่องของการตอบสนองความต้องการของลูกค้า การผลิตสินค้าใหม่ ๆ เข้าสู่ตลาด การสร้างโอกาสทางธุรกิจ เป็นต้น ประโยชน์ในข้อนี้ ถือได้ว่าเป็นปัจจัยที่สำคัญอย่างยิ่งสำหรับองค์กรต่าง ๆ ในปัจจุบัน ลักษณะที่เห็นได้ คือ เป็นการทำในสิ่งที่ดีและสิ่งที่ใหม่ (Do better and new things)

2.1.1.6 การพัฒนาระบบสารสนเทศ

องค์กรใดๆก็ตามโดยทั่วไปจะมีระบบสารสนเทศที่ใช้งานอยู่และได้รับการนำไปใช้งานโดยผู้บริการแต่เมื่อดำเนินเการไประยะหนึ่งอาจจำเป็นต้องมีการปรับปรุงและพัฒนาระบบสารสนเทศ

เหตุที่มาของการพัฒนาระบบสารสนเทศ มักจะเกิดขึ้นจากสาเหตุดังนี้

1. เพื่อแก้ไขปัญหาที่เกิดขึ้น ไม่ว่าจะเนื่องด้วย การวางระบบเดิมไม่เหมาะสมหรือสภาพการณ์เปลี่ยนแปลงไปเช่นองค์กรขยายใหญ่ขึ้น ปริมาณข้อมูลเพิ่มมากขึ้น เกิดความล่าช้าในการทำงานอย่างมาก
2. เพื่อให้สามารถตอบสนองความต้องการใหม่ เมื่อระบบเดิมที่มีอยู่ไม่สามารถเอื้ออำนวย หรือตอบสนองต่อความต้องการใหม่ที่เกิดขึ้นได้ ก็ต้องมีการปรับปรุงระบบสารสนเทศ
3. เพื่อนำความคิดและเทคโนโลยีใหม่มาใช้ การเกิดขึ้นของแนวคิดหรือเทคโนโลยีใหม่ซึ่งสามารถนำมาใช้ในการปรับปรุงระบบสารสนเทศที่มีอยู่ให้มีประสิทธิภาพเพิ่มขึ้นอย่างมาก เป็นหนึ่งในเหตุผลที่ทำให้เกิดการพัฒนาระบบสารสนเทศขึ้นใหม่
4. เพื่อพัฒนาระบบสารสนเทศทั้งระบบให้มีประสิทธิภาพมากขึ้น ในบางกรณีระบบสารสนเทศที่มีอยู่ใช้มาเป็นเวลานาน เกิดความล้าสมัย และทำงานได้ผลไม่ดีเท่าที่ควร ดังนั้นจึงอาจเกิดแนวคิดในการปรับปรุงทั้งระบบใหม่ให้มีประสิทธิภาพมากยิ่งขึ้น
5. วงจรการพัฒนาระบบสารสนเทศ เป็นขั้นตอนในการพัฒนาระบบสารสนเทศ ซึ่งประกอบด้วยขั้นตอนในการพัฒนา 3 ขั้นตอนหลัก คือ
   - การศึกษาเบื้องต้น
   - การศึกษาความเป็นไปได้
   - การพัฒนาและปรับใช้ระบบสารสนเทศ

รูปแบบของการพัฒนาระบบสารสนเทศมีรูปแบบและวิธีการที่ใช้อยู่ โดยทั่วไปในองค์กรต่าง ๆ ดังนี้

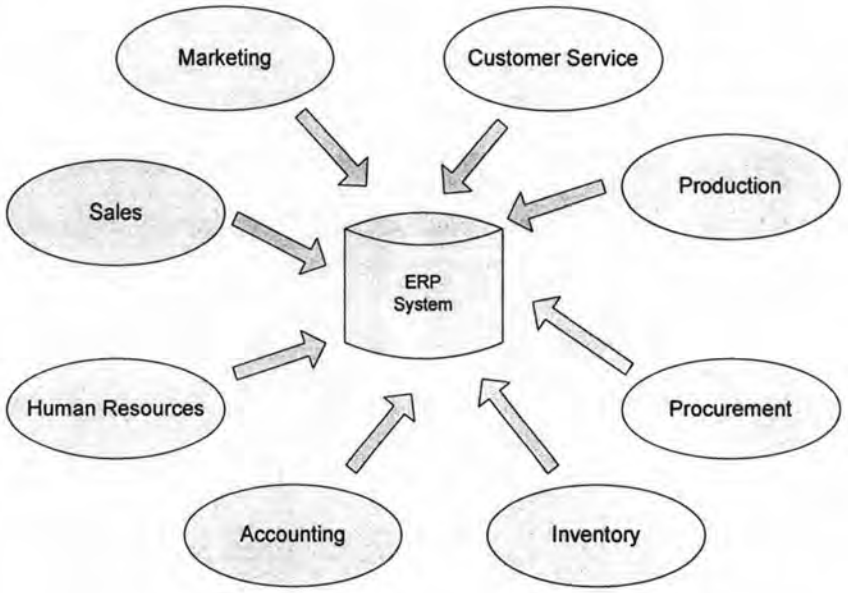
1. การพัฒนาระบบงานตามวงจรการพัฒนาระบบ (System Development Life Cycle)
2. การพัฒนาระบบงานโดยการสร้างระบบต้นแบบ (Prototyping)
3. การพัฒนาระบบงานโดยการนำชุดซอฟต์แวร์สำเร็จรูปมาใช้ (Application Software Package)
4. การพัฒนาระบบงานโดยผู้ใช้งานปลายทาง (End-User Development)
5. การพัฒนาระบบงานโดยการจ้างหน่วยงานภายนอก (Outsourcing)

2.1.2 ระบบ ERP (Enterprise Resource Planning)

(Khalid Sheikh, 2002)

2.1.2.1 ความหมายของระบบ ERP (Enterprise Resource Planning)

ระบบ ERP (Enterprise Resource Planning) หมายถึง ระบบที่ช่วยในการจัดการกระบวนการทางธุรกิจ ทั้งหมดในบริษัท ไม่ว่าจะเป็นระบบงานขาย ระบบตลาด ระบบจัดซื้อ ระบบคลังสินค้า ระบบผลิต ระบบบัญชี และอื่นๆ ภายใต้ฐานข้อมูลอันเดียวกันดังรูปด้านล่าง เพื่อให้การทำงานภายในบริษัทเป็นไปอย่างสอดคล้อง เกิดความรวดเร็ว ไม่ซ้ำซ้อน และลดต้นทุนทั้งระบบได้ รวมทั้งยังจัดการระบบข้อมูลสารสนเทศสำหรับ ผู้บริหารให้ทราบถึงผลการดำเนินงาน เพื่อใช้ในการกำหนดกลยุทธ์ในการบริหารได้อย่างเหมาะสม ถูกต้อง และรวดเร็วทันการณ์



รูปที่ 2.1 แสดงแบบจำลองแนวคิดระบบ ERP

ระบบ ERP ได้เตรียมไว้สำหรับทุกธุรกรรมทั้งแบบการซื้อมา-ขายไป โรงงาน โรงแรม โรงพยาบาล ธนาคาร ประกันภัย การขนส่ง การเช่าซื้อ และมูลนิธิ ระบบ ERP ได้เตรียมสำนักงานและพนักงานเพื่อให้บริษัทสามารถทำธุรกรรมได้ด้วยคนเพียงคนเดียว ข้อมูลในระบบงานจะไหลไปตามโครงสร้างของทางธุรกิจ ระบบจะเก็บข้อมูลที่เกิดขึ้นไว้บนฐานข้อมูลอันเดียวกัน และอนุญาตให้ผู้ใช้สามารถสร้าง แก้ไข หรือสอบถามดูข้อมูลได้ตลอดเวลาจากเครื่องคอมพิวเตอร์ของตนเองในทุกที่ที่ต้องการ ระบบ ERP สามารถแสดงรายการของข้อมูลได้หลาย

สกุลเงินพร้อมทั้งประมาณการราคาต้นทุนต่อหน่วยและแสดงต้นทุนในการขายสำหรับผู้บริหาร เนื่องจากฐานข้อมูลเป็นระบบที่มีประสิทธิภาพสูง และง่ายต่อการแก้ไขเปลี่ยนแปลงในด้านของการหน้าจอการ Interface

2.1.2.2 ประวัติความเป็นมาและอนาคตของระบบ ERP

ระบบ ERP เริ่มต้นจากซอฟท์แวร์แพ็คเกจขนาดใหญ่ ซึ่งเริ่มแพร่หลายตั้งแต่ทศวรรษ 1970 โดยเริ่มต้นจากการใช้งานในส่วนของการวางแผนความต้องการวัตถุดิบ (Material Requirement Planning : MRP) ในปี 1950 ซึ่งในขณะนั้นซอฟท์แวร์ลักษณะนี้ จะใช้สนับสนุนเพียงแค่เรื่องของการจัดการวัตถุดิบเท่านั้น ต่อมาในระหว่างทศวรรษ 1970 ได้มีการเพิ่มแอพพลิเคชันงานให้กับระบบ MRP มากขึ้นเรื่อยๆ จนเป็นผลให้เกิดระบบ MRP II ขึ้น และพัฒนาต่อเนื่องเรื่อยมาจนถึงทศวรรษ 1980 มีการเพิ่มหน้าที่งาน (function) ต่างๆ มากมายเข้าไปในระบบ MRP II จนกระทั่งในทศวรรษ 1990 ระบบ ERP ก็ได้ถูกเปิดตัวขึ้นเป็นครั้งแรก

วิสัยทัศน์ในเรื่องของ การรวบรวมระบบสารสนเทศของทุกกระบวนงานภายในบริษัทเข้าด้วยกัน หรือ "หนึ่งบริษัท หนึ่งระบบ" ได้ถูกนำเสนอขึ้นมาตั้งแต่ทศวรรษ 1970 ซึ่งในเวลาขณะนั้น ระบบสารสนเทศในส่วนงานต่างๆ แทบจะไม่ได้ถูกรวบรวมเข้าด้วยกันเลย อีกทั้งเมื่อมีแอพพลิเคชันใหม่ๆ เพิ่มขึ้นมา ก็จะถูกป้อนใส่เข้าไปเป็นเสมือนระบบสารสนเทศอีกหลายๆ ส่วนที่แยกออกมา ซึ่งส่งผลให้เกิดระบบส่วนเกิน หรือส่วนที่ไม่ต้องการเกิดขึ้น และทำให้โครงสร้างของระบบมีความซับซ้อนมากขึ้น ยกตัวอย่างเช่น เมื่อมีการป้อนข้อมูลใหม่ๆ เกิดขึ้นอย่างต่อเนื่อง มันจะเป็นการยากมากที่ระบบส่วนย่อยๆ ที่แยกออกมาเหล่านั้นจะถูกป้อนข้อมูลใหม่ๆ เหล่านั้นไปด้วย ซึ่งส่งผลให้การวิเคราะห์ข้อมูลขาดประสิทธิภาพ และคุณภาพ ซึ่งด้วยเหตุผลดังที่กล่าวนี้จึงทำให้มีระบบ ERP เกิดขึ้น โดยในช่วงแรกนั้นระบบจะถูกออกแบบมาเพื่อสนองตอบความต้องการของแต่ละบริษัท แต่เนื่องด้วยค่าใช้จ่ายในการบำรุงรักษาระบบนั้นเพิ่มสูงขึ้นอย่างรวดเร็ว และบ่อยครั้งที่ต้องลงทุนไปกับการสร้างระบบในส่วนงานใหม่ๆ ขึ้นมา จึงทำให้มี ERP Package เกิดขึ้นในทศวรรษที่ 1990 ซึ่ง ERP Package นี้เองที่สามารถช่วยแก้ปัญหาต่างๆ เหล่านั้นได้

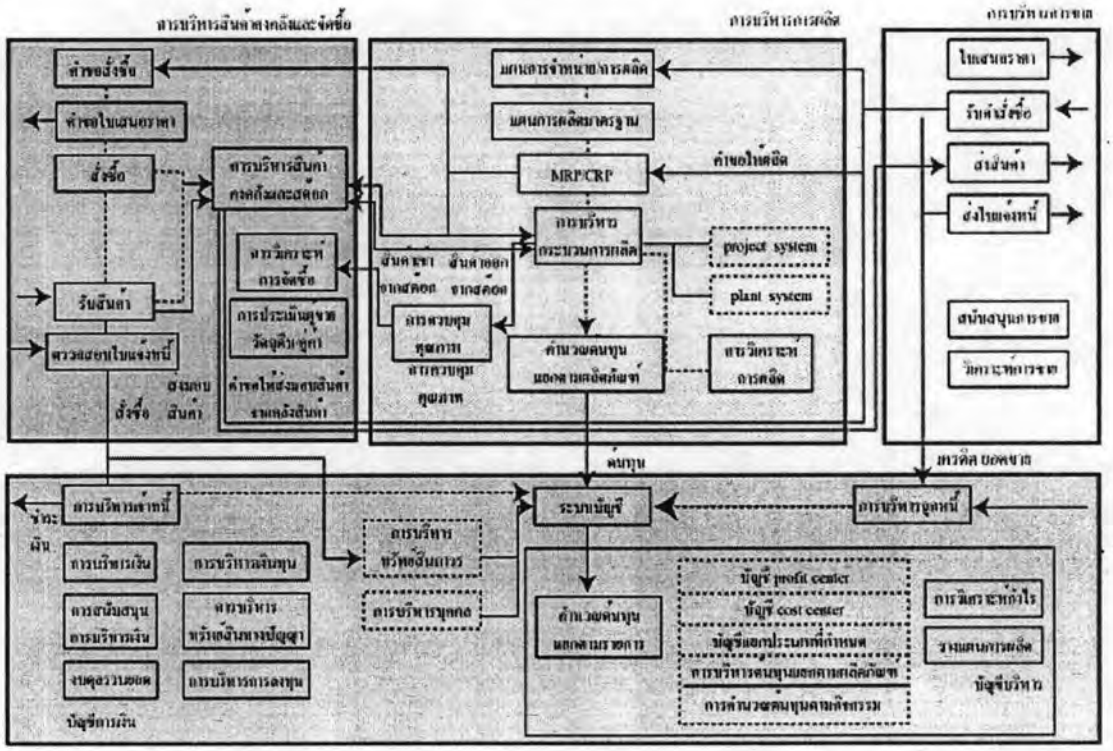
แต่อย่างไรก็ตาม ระบบ ERP ก็เหมือนกับระบบเทคโนโลยีสารสนเทศทั่วๆ ไป ที่เกิดการเปลี่ยนแปลงขึ้นอย่างรวดเร็ว เพราะในช่วงทศวรรษ 1980 ระบบ ERP ได้ถูกออกแบบมาสำหรับเมนเฟลมคอมพิวเตอร์ แต่พอย่างเข้าสู่ทศวรรษ 1990 กลับถูกแทนที่ด้วยโครงสร้างแบบเครือข่าย (Client-server) และในปัจจุบันนี้มีเวอร์ชันใหม่ออกมาให้ใช้งานบนเวปได้ และนอกเหนือจากนี้ฟังก์ชันงานในระบบ ERP เองก็เพิ่มขึ้นด้วย ยกตัวอย่างเช่น การจัดการห่วงโซ่

อุปทาน (Supply Chain Management: SCM) การจัดการด้านความสัมพันธ์กับลูกค้า (Customer Relation Management: CRM) และข้อมูลคลังสินค้า (Data warehousing) เป็นต้น

2.1.2.3 ลักษณะสำคัญของระบบ ERP

1. การบูรณาการระบบงานต่างๆ ของระบบ ERP

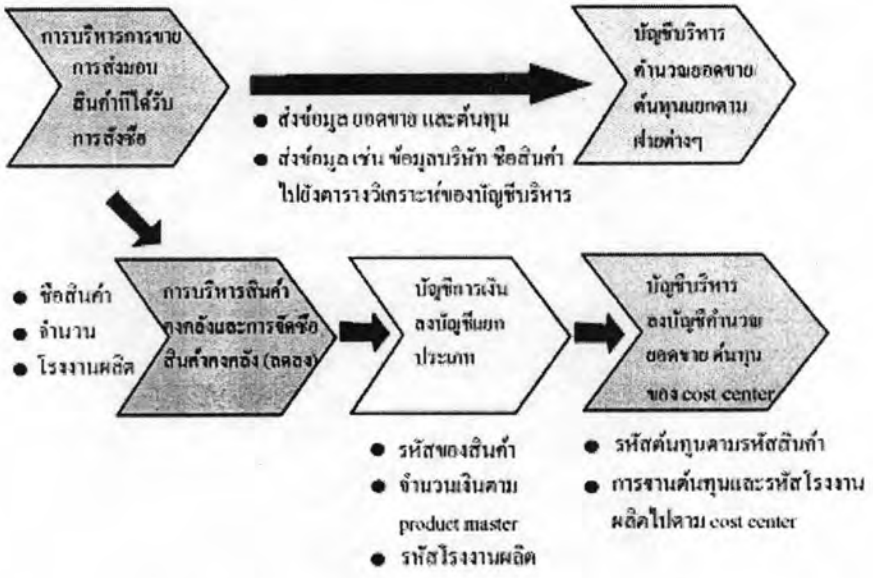
จุดเด่นของ ERP คือ การบูรณาการระบบงานต่างๆ เข้าด้วยกัน ตั้งแต่การจัดซื้อจัดจ้าง การผลิต การขาย บัญชีการเงิน และการบริหารบุคคล ซึ่งแต่ละส่วนงานจะมีความเชื่อมโยงในด้าน การไหลของวัตถุดิบสินค้า (Material flow) และการไหลของข้อมูล (Information flow) ERP ทำหน้าที่เป็นระบบการจัดการข้อมูล ซึ่งจะทำให้การบริหารจัดการงานในกิจกรรมต่างๆ ที่เชื่อมโยงกันให้ผลลัพธ์ออกมาดีที่สุด พร้อมกับสามารถรับรู้สถานการณ์และปัญหาของงานต่างๆ ได้ทันที ทำให้สามารถตัดสินใจแก้ปัญหาองค์กรได้อย่างรวดเร็ว



รูปที่ 2.2 การบูรณาการระบบงานต่างๆ ของระบบ ERP

2. รวมระบบงานแบบ real time ของระบบ ERP

การรวมระบบงานต่างๆ ของระบบ ERP จะเกิดขึ้นในเวลาจริง (Real time) อย่างทันที เมื่อมีการใช้ระบบ ERP ช่วยให้สามารถทำการปิดบัญชีได้ทุกวัน เป็นรายวัน คำนวณ ต้นทุนและกำไรขาดทุนของบริษัทเป็นรายวัน

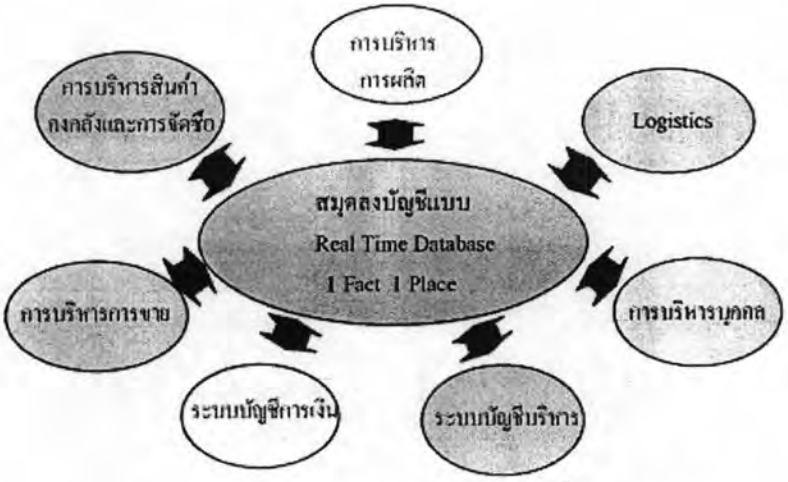


รูปที่ 2.3 การรวมระบบงานของ ERP แบบ Real Time

3. ระบบ ERP มีฐานข้อมูล (Database) แบบสมุดลงบัญชี

การที่ระบบ ERP สามารถรวมระบบงานต่าง ๆ เข้าเป็นระบบงานเดียวแบบ Real time ได้นั้น ก็เนื่องมาจากระบบ ERP มี database แบบสมุดลงบัญชี ซึ่งมีจุดเด่น คือ คุณสมบัติของการเป็น 1Fact 1 Place ซึ่งต่างจากระบบแบบเดิมที่มีลักษณะ 1 Fact Several Places ทำให้ระบบซ้ำซ้อน ขาดประสิทธิภาพ เกิดความผิดพลาดและขัดแย้งของข้อมูลได้ง่าย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.4 ERP มี database แบบสมุดลงบัญชี

2.1.3 การจำแนกชนิดและการกำหนดรหัส

2.1.3.1 นิยาม

Snead (1989)

การจำแนกชนิด (Classification) เป็นกระบวนการจัดหรือแบ่งแยกพัสดุออกเป็นกลุ่ม ตามคุณสมบัติ หรือตามสิ่งที่สนใจ เพื่อแยกสิ่งที่เหมือนกันออกเป็นกลุ่ม และบ่งชี้ความแตกต่างของชิ้นส่วนที่อยู่คนละกลุ่มได้

การกำหนดรหัส (Coding) เป็นกระบวนการกำหนดสัญลักษณ์ เพื่อใช้สื่อความหมายในการติดต่อสื่อสาร ซึ่งอาจจะแปลความหมายจากสัญลักษณ์ เพื่อใช้เป็นข้อมูลในการนำไปใช้งานต่อไป

2.1.3.2 โครงสร้างของระบบการจำแนกและการกำหนดรหัส

ระบบการจำแนกและการกำหนดรหัสมีมากกว่า 100 ระบบ ตามการพัฒนาเทคโนโลยีกลุ่ม (Group Technology) สามารถแบ่งประเภทได้เป็น 3 ประเภทใหญ่ๆ คือ

1. ระบบรหัสแบบตามลำดับชั้น (Hierarchical Code or Monocode)

รหัสประเภทนี้ความหมายของรหัสแต่ละตัว จะขึ้นกับความหมายของรหัสตัวก่อนหน้านั้นหมายความว่ารหัสแต่ละตัวจะขยายความข้อมูลหรือความหมายของรหัสตัวก่อนหน้า

การกำหนดรหัสตามลำดับขั้น สามารถให้ข้อมูลได้มาก ด้วยจำนวนหลักที่สัมพันธ์กันในจำนวนน้อย การกำหนดความหมายของแต่ละหลักสำหรับระบบนี้มีความยุ่งยาก แม้ว่าการใช้งานจะง่ายก็ตาม เริ่มจากส่วนหลักของโครงสร้างต้นไม้ หาคำตอบตามคำถามที่เกี่ยวกับหัวข้อของรหัสเรื่อยๆ ไปจนกระทั่งสิ้นสุดกิ่ง แล้วบันทึกแต่ละทางเลือกที่ได้ตอบคำถามไว้ก็จะได้รหัสตามที่ปรารถนา อย่างไรก็ตามการหาความหมายของรหัสแต่ละตัวก็จะมีความซับซ้อน เพราะว่ารหัสหลักก่อนหน้านั้นจะต้องถูกตีความหมายออกมาก่อน ฝ่ายออกแบบมักจะมักจะใช้ระบบรหัสแบบนี้ สำหรับชิ้นส่วนเพราะว่าสามารถให้ข้อมูลด้านรูปร่าง วัสดุ และขนาดได้อย่างมีประสิทธิภาพ ตรงข้ามกับฝ่ายผลิต ที่ต้องการทราบข้อมูล ขบวนการทางด้านกระบวนการผลิตเป็นหลัก ทำให้ยากแก่การแก้ไขและวิเคราะห์ถึงความสัมพันธ์ทางด้านข้อมูลกระบวนการ เมื่อมีการใช้โครงสร้างตามลำดับชั้นในการออกแบบและการผลิต

2. ระบบรหัสแบบคุณลักษณะ (Attribute or Polycode)

รหัสแบบคุณลักษณะ (Attribute Code) อาจะเรียกได้ว่า Polycode, รหัสลูกโซ่ (Chain Code), รหัสไม่ต่อเนื่อง (Discrete Code) หรือรหัสตำแน่งตัวเลขตายตัว (Fixed-digit Code) ความหมายของรหัสแต่ละตำแหน่งจะเป็นอิสระต่อกันไม่ขึ้นกับตำแหน่งอื่น ดังนั้นแต่ละคุณลักษณะของชิ้นส่วนสามารถระบุลงในแต่ละตำแหน่งของรหัสแบบลูกโซ่ได้

การใช้รหัสแบบลูกโซ่ในการกำหนดรหัสชิ้นส่วนต่างๆ และแก้ไขชิ้นส่วนจะสามารถทำได้ง่ายขึ้น ถ้ามีการใช้ระบบคอมพิวเตอร์มาช่วย ด้วยเหตุนี้ระบบรหัสแบบลูกโซ่จึงเป็นที่นิยมทั้งในระบบที่เกี่ยวข้องกับการผลิตหรือโรงงาน เพราะจะทำให้ง่ายต่อการระบุชิ้นส่วนที่มีคุณสมบัติที่คล้ายกัน ซึ่งต้องใช้กระบวนการผลิตคล้ายกัน ข้อเสียของรหัสแบบนี้ คือตำแหน่งของรหัสจะต้องถูกจองสำหรับแต่ละคุณลักษณะที่ต่างกันของชิ้นส่วน ดังนั้นรหัสที่ได้อาจจะยาวมาก

3. ระบบรหัสแบบผสม (Hybrid Code or Mixed code)

รหัสผสม คือ รหัสที่มีโครงสร้างผสมระหว่างรหัสแบบตามลำดับชั้นและรหัสแบบลูกโซ่ ซึ่งการกำหนดรหัสส่วนใหญ่มักจะใช้รหัสผสม เพราะนำข้อดีของรหัสตามลำดับชั้นและรหัสแบบลูกโซ่มาใช้ ดังตัวอย่างต่อไปนี้ ตำแหน่งที่ 1 จะบ่งชี้ประเภทชิ้นส่วน เช่น ประเภทเฟือง อีก 5 ตำแหน่งถัดไป บ่งชี้ถึงลักษณะของเฟือง และตำแหน่งที่ 7 จะระบุถึงรายละเอียดเพิ่มเติมเช่น วัสดุ ด้วยเหตุนี้รหัสแบบผสมจึงเป็นที่นิยมเพราะสามารถแสดงถึงความสัมพันธ์ที่ซับซ้อนของชิ้นส่วนได้ดีกว่ารหัสแบบลูกโซ่เพียงอย่างเดียวอีกทั้งยังง่ายต่อการจำแนกชิ้นส่วนที่มีคุณลักษณะพิเศษได้อีกด้วย

2.1.3.3 การเลือกหรือออกแบบระบบรหัส

ปัจจัยที่ควรคำนึงถึงในการเลือกหรือออกแบบระบบรหัส คือ

1. วัตถุประสงค์ (Objective) ซึ่งขึ้นอยู่กับผู้ใช้ว่า มีวัตถุประสงค์จะใช้ระบบใด ทั้งผู้ที่อยู่ในฝ่ายวิศวกรรม ฝ่ายผลิตหรือทั้งสองฝ่าย

ฝ่ายวิศวกรรม มีวัตถุประสงค์เพื่อ

- จัดหาระบบการแสดงข้อมูลที่มีประสิทธิภาพสำหรับชิ้นส่วนที่เหมือนกัน
- จัดหาข้อมูลของชิ้นส่วนที่มีรูปแบบมาตรฐาน
- จัดหาความหมายในการวางกำลังการผลิตผลที่ได้อย่างมีประสิทธิภาพ

ฝ่ายการผลิต มีวัตถุประสงค์เพื่อ

- จัดหาข้อมูลที่ต้องการจากรูปแบบของชิ้นส่วน
- จัดหาเพื่อการแก้ไขการวางแผนกระบวนการผลิตได้อย่างมีประสิทธิภาพ
- จัดหาความหมายของกลุ่มเครื่องจักรหรือกลุ่มชิ้นส่วนได้อย่างมีประสิทธิภาพ

2. ความหลากหลาย (Robustness) การเลือกระบบใดควรที่จะควบคุมชิ้นส่วนที่มีอยู่ทั้งหมดของบริษัท การวิเคราะห์ ควรพิจารณาการวางแผนการใช้งานเทคโนโลยีกลุ่ม และคุณสมบัติต่างๆ ของชิ้นส่วนที่สัมพันธ์กันในการนำไปใช้งาน
3. ความสามารถในการขยาย (Expandability) เพราะว่าการกำหนดรหัสเป็นสิ่งที่ยุ่งยากในการที่จะหาทุกสิ่งได้ตามที่มีการกำหนดรหัสไว้ ดังนั้นควรที่การสำรองรหัสเผื่อไว้สำหรับลักษณะต่างๆ ที่สำคัญๆ
4. ความแตกต่าง (Differentiation) ความแตกต่างของรหัสที่ถูกพัฒนาขึ้นมาภายหลัง จะต้องสามารถแยกแยะชิ้นส่วนทั้งหมดที่ผลิตขึ้นมาได้
5. ความเป็นอัตโนมัติ (Automation) ระบบการจำแนกและกำหนดรหัสส่วนใหญ่ที่ใช้ในปัจจุบันต้องสามารถนำไปใช้กับคอมพิวเตอร์ได้ เมื่อมีการประเมินศักยภาพของระบบทำให้ใช้เวลาน้อย ซึ่งการประเมินนี้ต้องไม่มีข้อจำกัดของการจำแนกและการกำหนดรหัส โดยการพิจารณาถึงฐานข้อมูล สิ่งที่ต้องการแก้ไขและวิเคราะห์หน้าที่การทำงาน

6. ประสิทธิภาพ (Efficiency) รหัสที่มีประสิทธิภาพ จะต้องพิจารณาถึงจำนวนตำแหน่งของรหัส ถ้าจำนวนรหัสสั้นเกินไป ควรพิจารณาการเพิ่มตำแหน่งของรหัสให้สามารถให้ข้อมูลได้มากขึ้น
7. ราคา (Cost) ควรพิจารณาถึงราคาเบื้องต้นของระบบที่ใช้ ราคาที่ใช้ในการปรับปรุงเปลี่ยนแปลงเพื่อให้เหมาะสมกับความต้องการ ราคาที่คอมพิวเตอร์มาใช้หับระบบและราคาที่เกิดจากการใช้ระบบ
8. ความซับซ้อน (Simplicity) ความง่ายต่อการใช้งานมีความสำคัญ เพราะบุคคลากรจำนวนมากที่ใช้จะไม่คุ้นเคยกัยระบบคอมพิวเตอร์เพื่อให้ผู้ใช้ยอมรับจึงต้องมีการอบรมบุคลากรที่ใช้งานนั้น

2.1.4 ระบบฐานข้อมูลของการควบคุมการผลิตและคลังสินค้า

(Smith B., 1989)

ข้อมูลหลักของระบบควบคุมการผลิตและคลังสินค้าในแต่ละองค์กร สามารถแบ่งได้ 5 ประเภทหลักดังนี้

- ข้อมูลพัสดุหลัก (Item Master File)
- ข้อมูลสูตรการผลิต (Bill of Material File)
- ข้อมูลเส้นทางการผลิต (Routing File)
- ข้อมูลสถานีงาน (Work Center File)
- ข้อมูลอุปกรณ์เครื่องมือ (Tool File)

2.1.4.1 ข้อมูลพัสดุหลัก (Item Master File)

ข้อมูลพัสดุหลัก อาจจะเรียกอีกอย่างได้ว่า ข้อมูลชิ้นส่วนหลัก (Part Master File) หรือ ข้อมูลรายการพัสดุคงคลัง (Inventory record File) ประกอบด้วยรายการของพัสดุทุกประเภทในคลังสินค้า เช่น สินค้าสำเร็จรูป สินค้ากึ่งสำเร็จรูป ชิ้นส่วน วัตถุดิบ และวัสดุสิ้นเปลือง สามารถแบ่งออกได้ 4 ประเภทตามชนิดของข้อมูล คือ การบ่งชี้และคำอธิบาย, ข้อมูลการวางแผน, รายการคลังสินค้า, ข้อมูลต้นทุน สำหรับใช้ในทางบัญชี ข้อมูลพัสดุหลักโดยทั่วไปประกอบด้วยข้อมูลดังรูป

Item number
Item description
Drawing number
Routing number
Group technology code
User code
Make/buy code
Buyer/planner code
Master schedule code
Low-level code
Unit of measure
ABC class
Engineering change number
Engineering change reason code
Engineering change disposition
Engineering change effective date
Order policy code
Fixed order quantity
Maximum order quantity
Minimum order quantity
Order quantity multiple
Safety Stock
Setup lead time
Run lead time
Forecast model code
Smoothed average
Trend
Seasonal indices
MAD/standard deviation
Smoothing labor cost
Actual labor cost
Standard overhead cost
Actual overhead cost
Total ordering cost
Inventory carrying cost
Selling price
Stock locations
Pegging code
Purchase lead time
Shrinkage factor
On-hand inventory
Allocated inventory
Forecast demand
Gross requirements
Scheduled receipts
Projected available balance
Firm planned orders
Planned orders
Time fences

รูปที่ 2.5 รายการข้อมูลพัสดุหลัก

2.1.4.2 ข้อมูลสูตรการผลิต (Bill of Material File)

สูตรการผลิตระบุถึงวัตถุดิบ, ชิ้นส่วน, สินค้ากึ่งสำเร็จรูป ที่ต้องใช้สำหรับการผลิตสินค้า สูตรการผลิตแต่ละรายการบันทึกความสัมพันธ์ระหว่างพัสดุที่ต้องการทำ (Parent item) และชิ้นส่วนประกอบ ซึ่งพัสดุที่ต้องการทำและชิ้นส่วนประกอบนั้นสามารถเป็นได้ทั้ง สินค้าสำเร็จรูป, สินค้ากึ่งสำเร็จรูป, วัตถุดิบ และวัสดุสิ้นเปลือง ซึ่งประกอบข้อมูลหลักดังรูป

Component item number
Parent item number
Quantity per parent
Unit of measure
Point of usage
Parent operation number
Scrap allowance
Current engineering change number

รูปที่ 2.6 รายการข้อมูลสูตรการผลิต

ข้อมูลของสูตรการผลิตสามารถแสดงได้ในหลายรูปแบบ โดยส่วนมากใช้ 6 รายงาน ซึ่งแบ่งเป็น 2 กลุ่มประกอบด้วย การแผ่ขยาย (Explosion) แสดงให้เห็นถึงชิ้นส่วนประกอบที่ต้องในการการผลิต อีกประเภทคือ การแตกส่วน (Implosion) แสดงถึงที่ที่ชิ้นส่วนถูกใช้

1) Single- Level Bill

สูตรการผลิตระดับเดียวแสดงถึงรายการชิ้นส่วนที่ถูกใช้โดยตรงในการผลิต โดยจากบอกส่วนประกอบเพียงแค่ระดับเดียว ซึ่งใช้สำหรับการร้องขอคำสั่งผลิต และการวางแผนการผลิตแบบทีละระดับ

2) Indented Bill

สูตรการผลิตแบบระดับขั้นแสดงให้เห็นทุกชิน้ส่วนในต้องใช้ในการผลิตและแสดงระดับที่ปรากฏอยู่ในโครงสร้างผลิตภัณฑ์ ซึ่งสูตรการผลิตแบบนี้จะแสดงถึงลำดับการจะประกอบชิ้นส่วนเข้าด้วยกัน และใช้ในการวางแผนการประกอบ และใช้ในการเตรียมรายการชิ้นส่วนประกอบในการให้บริการ

3) Summarized Bill

สูตรการผลิตอย่างย่อ หรือรายการชิ้นส่วน แสดงรายการชิ้นส่วนทั้งหมดของสินค้า ซึ่งจะแต่ละชิ้นส่วนจะแสดงเพียงครั้งเดียวโดยจำนวนที่แสดงจะเป็นจำนวนรวมทั้งหมดที่ต้องการใช้ผลิตสินค้าหนึ่งชิ้น ซึ่งใช้ในการวางแผนการผลิตโดยรวม และต้นทุนผลิตภัณฑ์

4) Single-Level Where Used

รายงานแสดงที่ใช้ระดับเดียวแสดงรายการของชิ้นส่วนประกอบที่ใช้ชิ้นส่วนโดยตรง ซึ่งใช้สำหรับการวิเคราะห์การเปลี่ยนแปลงทางวิศวกรรมเมื่อวิศวกรต้องการพิจารณาถึงชิ้นส่วนประกอบใดบ้าง ที่มีผลกระทบจากการเปลี่ยนเปลี่ยนและใช้ในการทบทวนการวางแผนการผลิตเมื่อชิ้นส่วนมีการขาดตลาด อาจจะเกิดจากผู้นำส่งหรือสาเหตุอื่นๆ

5) Indented Where Used

รายงานแสดงที่ใช้แบบระดับขั้น แสดงถึงชิ้นส่วนประกอบที่ใช้โดยตรง แสดงเป็นลำดับขั้นจากระดับที่อยู่สูงกว่า จนกระทั้งถึงสินค้าสำเร็จรูป ซึ่งอาจจะแสดงให้เห็นซ้ำหลายครั้งถ้าสมีการใช้ชิ้นส่วนประกอบซ้ำในหลายส่วน รายงานนี้ใช้ในการวิเคราะห์ความเป็นไปได้ในการเปลี่ยนแปลงทางวิศวกรรมในการพิจารณาถึงชิ้นส่วนประกอบและสินค้าที่มีผลกระทบทั้งหมด

6) Summarized Where Used

รายงานแสดงที่ใช้อย่างย่อ เป็นรายการพัสดุที่ใช้ชิ้นส่วนประกอบทั้งทางตรงและทางอ้อม รายงานนี้ใช้สำหรับการพิจารณาถึงผลกระบทจากการเปลี่ยนแปลงต้นทุนของชิ้นส่วน และใช้ในการวิเคราะห์คุณค่าผลิตภัณฑ์ที่มีจุกดประสงค์เพื่อลดต้นทุนและปรังปรุงประสิทธิภาพของผลิตภัณฑ์

2.1.4.3 ข้อมูลเส้นทางการผลิต (Routing File)

เส้นทางการผลิตระบุถึงงานที่ต้องทำในการผลิตพัสดุ โดยปกติข้อมูลนี้จะถูกเตรียมการโดยฝ่ายวิศวกรรมการผลิต ฝ่ายนี้จะพิจารณาถึงกระบวนการที่จะใช้,ลำดับในการผลิต และเครื่องจักรรวมถึงอุปกรณ์ที่ต้องใช้ ฝ่ายศึกษาการทำงานจะตั้งค่าเวลามาตรฐาน โดยทั่วไปจะประกอบด้วยข้อมูลดังรูป

Routing number
For each operation on the routing:
- Operation number
- Operation description
- Work center
- Machine number
- Drawing number
- Tool number
- Numerical control tape number
- Standard setup time
- Standard time labor run time per piece
- Standard machine run time per piece
- Interoperation time
- Overlap pieces or hours
- Number of machines for split operation
- Shrinkage
- Alternate operation number

รูปที่ 2.7 รายการข้อมูลเส้นทางการผลิต

เส้นทางการผลิตต้องระบุถึงรหัสจากรหัสพัสดุที่ใช้ในการผลิต ถึงพัสดุหลายพัสดุสามารถใช้เส้นทางการผลิตเดียวกันได้ ในบางโรงงานเส้นทางการผลิตจะระบุถึงสถานีงานแต่การเลือกใช้เครื่องจักรขึ้นอยู่กับหัวหน้างาน

เวลามาตรฐานในการทำงานของคนและเครื่องจักรถูกเก็บเนื่องจากมีความแตกต่างกันเมื่อพนักงานคนหนึ่งๆ ทำงานหลายเครื่องจักร กระบวนการทำงานทางเลือกอาจจะระบุเมื่อสถานีงานและเครื่องจักรในลำดับแรกเสียหรือไม่ว่างทำงาน

2.1.4.4 ข้อมูลสถานีงาน (Work Center File)

ข้อมูลสถานีงานจะแสดงถึงข้อมูลทรัพยากรในการผลิตขององค์กร สถานีงานเป็นกลุ่มของเครื่องจักรหรือเครื่องจักรหนึ่งเครื่อง หรืออาจจะรวมทั้งพนักงานที่พิจารณาให้เป็นหนึ่งหน่วยในการผลิตสำหรับการวางแผนกำลังการผลิตและการจัดตารางการผลิต ข้อมูลพื้นฐานของสถานีงานประกอบด้วยข้อมูลดังรูปด้านล่าง

Work center number
Description
Location
Department number
Number of machines
Number of workers
Work center efficiency
Work center utilization
Number of hours per shift
Number of shifts
Maximum daily hours
Machine rate
Labor rate
Overhead rate
Setup rate
Queue time
Wait time

For each machine in the work center:
    Machine number
    Machine description
    Last preventive maintenance date
    Next planned preventive
maintenance date

รูปที่ 2.8 รายการข้อมูลสถานีงาน

2.1.4.5 ข้อมูลอุปกรณ์และเครื่องมือ (Tool File)

ข้อมูลอุปกรณ์และเครื่องมือประกอบด้วยข้อมูลการบ่งชี้อุปรกรณ์ และสถานะ รวมทั้งข้อมูลการวางแผนการบำรุงรักษาตามระยะเวลา เครื่องมือแต่ละเครื่องจะมีหมายเลขประจำตัว เพื่อใช้ในการบ่งชี้เครื่องมือชนิดเดียวกัน ซึ่งประกอบด้วยข้อมูลพื้นฐานดังรูป

Tool number
Description
Status (in use, available, being repaired)
Number of tools
Serial number (to identify each of multiple like tools)
Storage location
Current assignment (work center, order number)
Usage code
Accumulated use since last repair
Estimate tool life
Date of last tool repair

รูปที่ 2.9 รายการข้อมูลเครื่องมือ

2.1.4.6 การเปลี่ยนแปลงทางวิศวกรรม (Engineering Changes)

การเปลี่ยนแปลงทางวิศวกรรมเป็นกระบวนการจัดการการเปลี่ยนแปลงข้อมูล มักเกิดขึ้นตามเวลาที่เปลี่ยนไป จำนวนการเปลี่ยนแปลงขึ้นอยู่กับระยะเวลาชีวิตของผลิตภัณฑ์และความซับซ้อนของผลิตภัณฑ์ เช่น เครื่องบิน เครื่องใช้ไฟฟ้า

การเปลี่ยนแปลงทางวิศวกรรมถ้าไม่มีการวางแผนและควบคุมให้เหมาะสมจะส่งผลให้เกิดค่าใช้จ่ายจำนวนมหาศาล เนื่องจากพัสดุที่เสีย, ค่าใช้จ่ายของเครื่องมือที่เกินความจำเป็น, การผลิตไม่มีประสิทธิภาพ และความล่าช้าในการผลิต แต่อย่างไรก็ตามการพัฒนาผลิตภัณฑ์โดยเป็นสิ่งจำเป็นในการรักษาตำแหน่งทางการตลาด และในบางครั้งเป็นสิ่งที่จำเป็นอย่างคับขัน ดังนั้นการเปลี่ยนแปลงต้องถูกพิจารณาอย่างหลีกเลี่ยงไม่ได้ และเป็นสิ่งท้าทายนการจัดการให้มีประสิทธิภาพและส่งผลต่อการหยุดชะงักในการผลิตให้น้อยที่สุด

การร้องขอการเปลี่ยนแปลงสามารถเกิดขึ้นได้ทุกที่ในองค์กรหรือจากลูกค้า ซึ่งอาจจะมาแรงบันดาลใจดังนี้

- ปรับปรุงประสิทธิภาพ
- ปรับปรุงความปลอดภัย
- ลดต้นทุน
- กฎระเบียบของทางราชการ

ระบบการจัดการการเปลี่ยนแปลงที่ดีต้องมาจากความร่วมมือของบุคลากรในแต่ละหน่วยงาน ดังนั้นคณะกรรมการการเปลี่ยนแปลงจะถูกจัดตั้งขึ้นเพื่อดูแล ซึ่งประกอบด้วยตัวแทนจากฝ่ายควบคุมการผลิต, ฝ่ายวิศวกรรม, ฝ่ายการผลิต, ฝ่ายการตลาดและการเงิน ฝ่ายจัดซื้อและฝ่ายควบคุมคุณภาพอาจจะเข้าร่วมด้วย หน้าที่ของคณะกรรมการมีดังนี้

พิจารณาข้อเสนอในการเปลี่ยนแปลงทางวิศวกรรม

- อนุมัติ หรือไม่อนุมัติ
- พิจารณาเริ่มต้นการเปลี่ยนแปลง
- วางแผนการประสานงานสำหรับการเปลี่ยนแปลงถึงผลกระบท
- ดำเนินการเปลี่ยนแปลง
- ทบทวนผลลัพธ์ที่เกิดขึ้น

การตัดสินใจถึงเวลาที่จะเริ่มเปลี่ยนแปลงนั้น อาจจะเรียกว่า Effectivity Date การอนุมัติการเปลี่ยนสามารถจัดเป็นกลุ่มได้ดังนี้

- Mandatory Immediate ยกเลิกผลิตภัณฑ์เดิมที่ยังไม่มีการเปลี่ยนแปลง การผลิตปัจจุบันต้องหยุด ซึ่งการตัดสินใจนี้จะเกิดขึ้นจากการค้นพบว่าสินค้ามีความผิดพลาดร้ายแรง เช่น สินค้ามีอันตราย
- As soon as possible การผลิตปัจจุบันต้องหยุดการผลิต แต่การเปลี่ยนแปลงจะถูกดำเนินการเมื่อชิ้นส่วนใหม่พร้อมและชิ้นส่วนเดิมจะถูกละทิ้ง บางทีการพัฒนาผลิตภัณฑ์อาจจะเป็นผลมาจากการพัฒนาของคู่แข่ง
- At minimum cost โดยส่วนมากจะหมายถึงการเปลี่ยนแปลงจะเริ่มขึ้นเมื่อพัสดุที่มีอยู่ในปัจจุบันถูกใช้จนหมด ในกรณีนี้จะถูกเลือกใช้สำหรับการพัฒนาประสิทธิภาพเล็กน้อยหรือการลดต้นทุนในการผลิตพัสดุ
- On a given date กรณีนี้อาจจะใช้กับการเปลี่ยนรุ่นหรือการเริ่มสัญญาใหม่
- Variance กรณีนี้เป็นการเปลี่ยนแปลงแบบชั่วคราวซึ่งเกิดขึ้นจากการที่ลูกค้าต้องการสินค้าที่ต่างจากมาตรฐานที่มี หรือบางทีอาจจะเกิดจากวัตถุดิบบางตัวในสูตรการผลิตขาดทำให้ต้องใช้วัตถุดิบทดแทน

2.1.5 การออกแบบระบบเชิงวัตถุ (Object Orientation)

2.1.5.1 Object Orientation

"Object Orientation" เป็นการมองทุกสิ่งในโลกความจริงให้เป็นวัตถุ (Object) ทั้งสิ่งที่จับต้องได้เรียกว่า "Tangible Object" เช่น คน ไพ่ รถยนต์ เป็นต้น ส่วนสิ่งที่จับต้องไม่ได้เรียกว่า "Intangible Object" เช่น เพลง วิชาเรียน ภาควิชา คณะ เป็นต้น

โดยทั่วไปอ็อบเจกต์หนึ่งๆ อาจอยู่นิ่งหรือไม่อยู่นิ่ง ถ้าไม่อยู่นิ่งก็จะดำเนินการหรือถูกดำเนินการอย่างใดอย่างหนึ่ง ซึ่งก่อให้เกิดกิจกรรม (Activity) ความเคลื่อนไหว (Movement) การกระทำ (Action) หรือการดำเนินการ (Operation) เช่น กิจกรรม คนปั่นจักรยาน เกิดจากคนดำเนินการ (ปั่น) ต่อจักรยาน เป็นต้น ดังนั้นหากพิจารณาในรายละเอียดแล้ว จะเห็นว่ากิจกรรมต่างๆ ที่เกิดขึ้นนั้น ล้วนเกิดจากการมีความสัมพันธ์ (Relationship) และปฏิสัมพันธ์ (Interaction) กันระหว่าง 2 ตัวขึ้นไป

- Relationship คือ ความเกี่ยวข้องหรือความสัมพันธ์กันระหว่างวัตถุ 2 ตัวขึ้นไป ซึ่งโดยทั่วไปความสัมพันธ์ดังกล่าวเราจะไม่สามารถมองเห็นได้โดยตรง แต่ต้องอาศัยการตีความ เช่น ความเป็นแม่-ลูก ความเป็นเจ้าของ เป็นต้น
- Interaction คือ ปฏิสัมพันธ์หรือการกระทำใดๆ ที่เกิดขึ้นระหว่างวัตถุ 2 ตัวขึ้นไป ซึ่งโดยทั่วไปเราสามารถมองเห็นหรือสังเกตเห็น Interaction ได้ง่าน เช่น การสร้าง การเปลี่ยนแปลง การเล่น การกระตุ้น เป็นต้น ซึ่ง Interaction นี้เองที่ทำให้เกิด กิจกรรม (Activity) ต่างๆ ในโลกนี้

2.1.5.2 อ็อบเจ็กต์ (Object) และคลาส (Class)

2.1.5.2.1 อ็อบเจ็กต์ (Object)

วัตถุ (Object) คือ ทุกๆ สิ่งที่เราสนใจในเหตุการณ์ใดเหตุการณ์หนึ่ง ทั้งที่จับต้องได้และจับต้องไม่ได้ แต่ที่ผ่านมานั้นเรามองแต่วัตถุที่อยู่ในกรอบของความสนใจที่มีอยู่ในโลกความจริงเท่านั้น ถ้าเราต้องการที่จะจำลองสิ่งต่างๆ ให้อยู่ในเครื่องคอมพิวเตอร์จะทำได้อย่างไร? สิ่งที่ต้องทำก็คือ เราต้องทำให้เกิดวัตถุขึ้นในเครื่องคอมพิวเตอร์ ในขณะเดียวกันก็ต้องสร้างความสัมพันธ์และปฏิสัมพันธ์ ระหว่างวัตถุประเภทต่างๆ ให้เกิดขึ้นในเครื่องคอมพิวเตอร์ด้วย ซึ่งแน่นอนว่าวัตถุในโลกกับวัตถุที่เกิดขึ้นในเครื่องคอมพิวเตอร์นั้นย่อมไม่เหมือนกัน และด้วยเหตุที่ว่าเราไม่สามารถนำเอาวัตถุในโลกความเป็นจริง เข้ามาใส่ในเครื่องคอมพิวเตอร์ได้ ดังนั้นสิ่งที่ต้องทำ

ก็คือ การใส่แนวคิด (Concept) ให้แก่วัตถุแล้วจึงสร้างแบบจำลองของวัตถุในโลกความเป็นจริงนั้นๆ เพื่อนำไปใส่ไว้ในคอมพิวเตอร์

แนวคิด หมายถึง ความคิดรวบยอดที่เรามีให้กับวัตถุใดๆ ภายใต้กรอบที่สนใจ เช่น ถ้าเราต้องการให้แนวคิดกับรถยนต์ นั่นคือ รถทุกคันมีตัวถัง มีล้อ และเครื่องยนต์เหมือนกันทุกคัน หรือเมื่อต้องการให้แนวคิดกับคน นั่นคือ คนทุกคนมี 2 แขน 2 ขา 1ศีรษะ และมีภาษาพูด เป็นต้น

2.1.5.2.2 คลาส (Class) หรือ Abstract Object

การให้แนวคิดกับวัตถุต่างๆ นั้นจะถูกกำหนดโดยกรอบที่สนใจ เพราะเราจะให้แนวคิดกับวัตถุในบางส่วนของวัตถุที่เราสนใจเท่านั้น เช่น เมื่อกรอบที่เราสนใจเฉพาะ แขน และขาของคนเท่านั้น ดังนั้นเราจะให้แนวคิดของคนว่าเป็นวัตถุที่มี 2 แขน และ 2 ขา โดยเราไม่สนใจ หู ตา หรือจมูก ของคนซึ่งถือว่าอยู่นอกเหนือกรอบที่สนใจ

ผลจากการให้แนวคิดกับวัตถุนั้นทำให้เกิดการจัดกลุ่มของวัตถุขึ้น ซึ่งกลุ่มของวัตถุที่ได้จากกระบวนการนี้เรียกว่า "Abstract Object" หรือเรียกอีกอย่างหนึ่งว่า "คลาส (Class)"

คลาสเกิดจากการให้แนวคิดกับวัตถุ ดังนั้นจึงมีความจริงข้อหนึ่งในทาง Object Orientation ว่า "คลาสถือเป็นนามธรรม (Abstract) เราไม่สามารถทำให้คลาสดำเนินกิจกรรมใดๆ ได้เลย" นอกจากนี้ คลาสต่างๆ ที่อยู่ในกรอบที่สนใจ คือ สิ่งที่อยู่ในความคิดเราซึ่งไม่สามารถทำกิจกรรมใดๆ ให้เกิดขึ้นจริงได้ แต่ถ้าเราต้องการให้เกิดกิจกรรมขึ้นในระบบคอมพิวเตอร์ของเรา เราต้องสร้างวัตถุของคลาสต่างๆ ขึ้นในคอมพิวเตอร์ของเราเสียก่อน เพื่อให้วัตถุนั้นๆ สามารถทำงานและดำเนินบทบาทของตนเองได้ ซึ่งหากเราจะเทียบกับแนวทางการพัฒนาโปรแกรมแบบเดิม แล้วคลาสจะคล้ายคลึงกับชนิดของตัวแปร และวัตถุจะคล้ายคลึงกับตัวแปรนั่นเอง

2.1.5.3 Abstraction และ Instantiation

เราเรียกกระบวนการในการให้ Concept กับอ็อบเจ็กต์จนเกิดเป็นคลาสว่า "Abstraction" และเรียกกระยวนการของการทำให้เกิดอ็อบเจ็กต์จากคลาสที่เราสร้างขึ้นว่า "Instantiation" ซึ่งในบางครั้ง หรือหนังสือบางเล่มจะเรียกอ็อบเจ็กต์ที่เกิดขึ้นในคอมพิวเตอร์ว่า "Instance" เพราะอ็อบเจ็กต์เป็นสิ่งที่เกิดจากกระบวนการ "Instantiation"

กระบวนการ Abstraction และ Instantiation ตามแนวคิด Object Orientation ซึ่งสามารถอธิบายได้ว่า เมื่อใดก็ตามที่เราต้องการใช้ Object Orientation เพื่อการวิเคราะห์และออกแบบระบบ สิ่งแรกที่ต้องทำก็คือ พิจาณาอ็อบเจ็กต์ทั้งหมดใน Domain ที่เราสนใจ ซึ่งอยู่ในโลกของความเป็นจริง โดยใช้หลักการต่างๆของ Abstraction เป็นเครื่องมือในการพิจารณาดังกล่าว ผลลัพธ์ที่ได้จากการพิจารณาก็คือ คลาสซึ่งเป็นสิ่งที่ไม่มีตัวตนอยู่จริง แต่เป็นสิ่งที่อยู่ในความคิดของเราที่สามารงเราที่สามารถนำเสนอ (Represent) ในรูปของแผนภาพ (Diagram) ได้

อย่างไรก็ตาม สิ่งที่อยู่ในความคิดนั้นไม่สามารถทำกิจกรรมใดๆ ได้ หรือกล่าวได้อีกอย่างหนึ่งคือ การที่เรามีเพียงคลาสนั้นยังไม่เพียงพอต่อการสร้างระบบ (เพราะระบบจำสมบูรณ์ได้ก็ต่อเมื่อมีกิจกรรมเกิดขึ้นภายในส่วนประกอบต่างๆ ของระบบ) แต่ถ้าเราใช้คลาสให้เป็นเสมือนแม่พิมพ์ เพื่อสร้างตัวตนที่จำลองภาพของวัตถุในโลกของความเป็นจริง ให้มีตัวตนอยู่จริงในคอมพิวเตอร์ วัตถุเหล่านั้นซึ่งเรียกว่า "อ็อบเจ็กต์" หรือบางครั้งเราจะเรียกว่า "Instance" (อยู่ด้านปลายของลูกศร) จะสามารถทำหน้าที่ก่อให้เกิดกิจกรรมต่างๆของระบบตามที่เราต้องการได้ เรียกกระบวนการในการสร้างอ็อบเจ็กต์จากคลาสว่า "Instantiation"

ถ้าเราได้ทราบแล้วว่า Abstraction เป็นการมองสิ่งต่างๆแล้วใส่ความคิดรวบยอด (Concept) ลงไปว่า สิ่งที่มองนั้นมีคุณลักษณะอย่างไร ดังนั้นการมองอ็อบเจ็กต์หนึ่งชนิดของหลายคนจะมีมุมมองต่างกันออกไปขึ้นอยู่กับความสนใจของแต่ละคน

นอกจากนี้ Abstraction ยังเป็นส่วนหนึ่งในการช่วยวิเคราะห์ถึงปัญหาของระบบที่ต้องการพัฒนา (Problem Domain) ซึ่งในการทำ Abstraction เพื่อการวิเคราะห์ Domain Problem จะมีกระบวนการย่อยหลายกระบวนการ

2.1.5.4 องค์ประกอบของอ็อบเจ็กต์

ทุกๆอ็อบเจ็กต์ตามแนวคิดจอง Object Orientation จะมีองค์ประกอบ 3 อย่าง ได้แก่ คุณสมบัติ (Attribute/Property) การดำเนินการหรือพฤติกรรม (Operation/Behavior/Method) และการบ่งชี้อ็อบเจ็กต์ (Unique Identity)

2.1.5.4.1 คุณสมบัติ (Attribute)

เราสามารถบรรยายคุณสมบัติของอ็อบเจ็กต์ต่างๆ ตราบเท่าที่คุณสมบัติดังกล่าวเป็นคุณสมบัติที่เราสนใจหรืออยู่ใน Domain ที่สนใจ เช่น สนและจำนวนของประตูรถคันหนึ่ง สีผิว

และเพศของคนๆหนึ่งเป็นต้น ในทาง Object Orientation จะเรียกสิ่งที่ใช้ในการบรรยายคุณลักษณะต่างๆ ของอ็อบเจ็กต์ว่า "Attribute" หรือ "Property"

2.1.5.4.2 การดำเนินการ/เมธอด (Operation/Method)

ในโลกความจริงทุกๆสิ่งต้องมีความสามารถดำเนินการบางอย่างได้ เช่น คนสามารถวิ่ง เดิน หรือ กรณีเครื่องเล่นซีดีที่มีความสามารถเล่นแผ่นซีดีได้ เป็นต้น ดังนั้น หากเรามองทุกๆสิ่งเป็นอ็อบเจ็กต์ ย่อมแสดงว่าทุกอ็อบเจ็กต์ต้องมี "ความสามารถในการดำเนินการ (Operation)" บางอย่างหรือหลายอย่างได้ ซึ่งหมายถึง การกระทำที่อ็อบเจ็กต์สามารถทำให้หรือสามารถถูกขอร้องให้กระทำได้

ความสามารถในการดำเนินการบางอย่างจะถูกแสดงออกมาให้เห็นเป็นพฤติกรรมได้ ต้องเกิดจากการสื่อสารหรือปฏิสัมพันธ์กันระหว่างอ็อบเจ็กต์ หมายความว่า จะต้องมีอ็อบเจ็กต์ใดอ็อบเจ็กต์หนึ่งเป็นตัวกระตุ้น (Trigger) อีกหนึ่งอ็อบเจ็กต์เป็นผู้ถูกกระตุ้น (หรือถูกกระทำ – Passive Object) ให้แสดงพฤติกรรมด้วยการที่ Trigger ส่ง Message ไปเรียกใช้ความสามารถของ Passive Object การปฏิสัมพันธ์กันระหว่างอ็อบเจ็กต์นี้เองที่ทำให้เกิดการดำเนินการใดๆขึ้นมาได้

2.1.5.4.3 การบ่งชี้อ็อบเจ็กต์ (Unique Identity / Object Identity)

ถ้าสังเกตการณ์อ้างถึงอ็อบเจ็กต์ต่างๆ ที่เราสนใจหรือกล่าวถึง เราจะใช้ประโยคหรือวลีที่บ่งบอกถึงความจำเพาะเจาะจงของอ็อบเจ็กต์นั้นๆ เช่น รถยนต์หมายถึงเลขทะเบียน..., คอมพิวเตอร์ยี่ห้อ A ของนาย ก เป็นต้น สาเหตุที่เราต้องระบุให้เจาะจง เนื่องจากอ็อบเจ็กต์แต่ละตัวจะไม่สามารถซ้ำกับอ็อบเจ็กต์ตัวอื่นๆได้ ไม่ว่ากรณีใดๆเรียกคุณสมบัติของความโดดเด่นและไม่ซ้ำกันของอ็อบเจ็กต์แต่ละตัวว่า "Unique Identity"

2.1.5.5 ประเภทของ Abstraction

ที่ผ่านมาทราบแล้วว่า Abstraction คือ กระบวนการในการสร้างแนวคิดของคลาสจากกลุ่มของอ็อบเจ็กต์ที่เราสนใจ ในหัวข้อนี้ได้หยิบเอากระบวนการ Abstraction มาจำแนกเป็น 4 ประเภทด้วยกันได้แก่ Classification, Association, Aggregation และ Generalization/Inheritance

2.1.5.5.1 Classification Abstraction

Classification Abstraction คือกระบวนการในการให้แนวคิดกับอ็อบเจ็กต์ที่เราสนใจ เพื่อก่อให้เกิดแนวคิดของคลาส ดังนั้นหัวใจสำคัญของ Classification Abstraction ก็คือ "แนวคิดรวบยอด (Concept)" ที่จะให้กับอ็อบเจ็กต์ การให้แนวความคิดคือ การให้ขอบเขตแก่อ็อบเจ็กต์ว่าต้องมีคุณลักษณะอะไรบ้าง Concept เป็นเครื่องมือสำคัญที่สามารถทำให้เราจัดหมวดหมู่ที่ไม่ซ้ำกันให้กับอ็อบเจ็กต์ใน Problem Domain ที่เราสนใจ หลังจากที่เราได้หมวดหมู่ของอ็อบเจ็กต์แล้ว ความคิดรวบยอดหรือ Concept ที่เรามีต่อกลุ่มของอ็อบเจ็กต์แต่ละกลุ่ม ก็คือ คลาสแต่ละคลาสนั่นเอง ลักษณะของการให้แนวคิดเพื่อแยกแยะ กำหนดขอบเขต และจัดหมวดหมู่ของอ็อบเจ็กต์เพื่อให้เกิดคลาสก็คือ ลักษณะของ "Classification Abstraction"

ก่อนที่เราจะให้แนวคิดรวบยอดกับอ็อบเจ็กต์ที่เราสนใจ Problem Domain สิ่งสำคัญสิ่งแรกที่ควรกระทำก็คือ "การกำหนดขอบเขตของ Problem Domain" ขึ้นมาก่อน ซึ่งสามารถกำหนดได้โดยอาศัย "ข้อมูลความต้องการจากผู้ใช้ระบบหรือผู้ที่เกี่ยวข้อง (User Requirement)" เป็นส่วนประกอบหลักในการพิจารณา หลังจากที่สามารถกำหนดขอบเขตของ Problem Domain แล้วก็สามารถค้นหาอ็อบเจ็กต์ที่เราสนใจ และทำการให้แนวคิด แยกแยะและจัดหมวดหมู่ จนกลายเป็นคลาสขึ้นมาได้ในที่สุด

2.1.5.5.2 Association Abstraction

Association หมายถึงความสัมพันธ์ระหว่างคลาส (หรืออ็อบเจ็กต์) ที่อยู่ในระดับเดียวกัน กล่าวคือ คลาสทั้งสองมีความสำคัญเท่าเทียมกัน ไม่มีคลาสใดเป็นองค์ประกิบของคลาสใด เช่น ลูกค้าจัดทำใบสั่งซื้อ นักเรียนลงทะเบียนวิชาเรียน สินค้าอยู่ในคลังสินค้า เป็นต้น และกระบวนการในการหาความสัมพันธ์ระหว่างคลาสที่เราสนใจในลักษณะที่คลาสทั้งสองมีความเกี่ยวข้องสัมพันธ์กันในระดับเดียวกัน เรียกว่า "Association Abstraction"

คลาสที่มีความสัมพันธ์กันจะถูกเชื่อความสัมพันธ์ด้วย "ชื่อความสัมพันธ์ (Association Name)" เช่น ลูกค้าจัดทำใบสั่งซื้อ คลาส "ลูกค้า" กับ "ใบสั่งซื้อ" ถูกเชื่อมความสัมพันธ์ด้วย Association ที่ชื่อว่า "จัดทำ (Make)" เป็นต้น ดังนั้นการเขียนสัญลักษณ์แทนความสัมพันธ์ระหว่างคลาสจึงควรระบุชื่อ Association ไว้ด้วย โดยอาจใช้ลูกศรเพื่อแสดงให้เห็นทิศทางของความสัมพันธ์ด้วย

2.1.5.5.3 Aggregation Abstraction

I22916015

Aggregation เป็นความสัมพันธ์อีกชนิดหนึ่งของ Association โดยที่ Aggregation หมายถึงความสัมพันธ์ระหว่างคลาส (หรืออ็อบเจ็กต์) แบบต่างระดับกัน กล่าวคือ คลาสหนึ่งมีความสัมพันธ์แบบเป็นองค์ประกอบ (Part) ของอีกคลาสหนึ่ง (Whole) หรือเรียกว่า ความสัมพันธ์แบบ "Whole-Part" และกระบวนการในการหาความสัมพันธ์ระหว่างคลาสในลักษณะดังกล่าวจะเรียกว่า "Aggregation Abstraction"

จากย่อหน้าข้างต้น ประกอบกับการพิจารณาความเป็นจริงในโลก จะพบว่าจะมีวัตถุหลายชนิดในโลกที่เกิดจากการรวมตัวกับวัตถุอื่น เช่น คนเกิดจากการรวมตัวกันเองของ แขน ขน หัว ลำตัว หรือคอมพิวเตอร์เกิดจากการรวมตัวกันของ Main Board, Rom, Disk Drive และ Case ในทางกลับกันเราสามารถบอกหรือสรุปได้ว่า วัตถุชั้นหนึ่งสามารถแยกออกเป็นวัตถุย่อยๆ ได้ ซึ่งสิ่งที่แบ่งออกนั้นมี Concept ที่แตกต่างจากเดิม เช่น หนังสือสามารถแบ่งแยกออกเป็นหน้าปกและหน้าหนังสือ หรือ คณะรัฐมนตรีแบ่งออกเป็นฝ่ายค้านและฝ่ายรัฐบาลเป็นต้น

ดังนั้นหากเป็นเนื้อหาในทาง Object Orientation แล้วเราอาจกล่าวได้ว่าคลาสบางคลาสในโลกสามารถแยก (Decompose หรือ Split) ออกเป็นคลาสย่อยๆได้ แต่ละคลาสย่อยนั้นมี Concept ที่แตกต่างออกไปได้ หลายๆคลาสในทางตรงกันข้าม เราสามารถกล่าวได้ว่าคลาสหลายๆคลาสที่มี Concept แตกต่างกัน เมื่อนำมารวมกัน (Compose หรือ Assemble) ก็สามารถที่จะสร้างคลาสใหม่ซึ่งมี Concept ใหม่ได้เช่นเดียวกัน ลักษณะดังกล่าวเป็นลักษณะความสัมพันธ์แบบ "Whole-Part หรือ Aggregation" นั่นเอง

2.1.5.5.4 Generalization Abstraction หรือ Inheritance

Generalization หมายถึง ความสัมพันธ์แบบต่างระดับระหว่างคลาสหลัก (Superclass) กลับคลาสรอง (Subclass) โดยที่ Subclass จะสืบทอดคุณลักษณะทั้ง Attribute และ Operation ที่สำคัญของ Superclass นั้นมาด้วย ทำให้ Subclass มี Attribute บางอย่างเหมือนกับ Superclass ในขณะเดียวกัน Subclass เองก็สามารถสร้าง Attribute และ Operation เพิ่มเติมได้ด้วย

จากความสัมพันธ์แบบ Aggregation ซึ่งเป็นความสัมพันธ์แบบต่างระดับเหมือนกัน แต่ Aggregation ใช้อธิบายความจริงบนโลกที่ว่ามีวัตถุหลายชนิดที่เกิดจากการประกอบรวมเข้าด้วยกันของวัตถุอื่นๆ แสดงว่าวัตถุอื่นที่เป็นองค์ประกอบกับวัตถุหลัก ไม่มีความคล้ายคลึงกันทางด้านโครงสร้าง แต่ยังมีความจริงบนโลกอีกประการหนึ่งของความสัมพันธ์แบบต่างระดับที่ว่า "วัตถุหลักและวัตถุย่อยที่ถึงแม้ว่าจะอยู่ต่างระดับกันแต่มีความคล้ายคลึงกัน

ทางด้านโครงสร้างและความหมาย (มีลักษณะบางอย่างร่วมกัน)" เช่นการถ่ายทอดลักษณะทางพันธุกรรมจากพ่อ แม่ มาสู่ลูก ที่ลูกจะสืบทอดลักษณะบางประการมาจากพ่อแม่ ทำให้ลูกมีโครงสร้างบางอย่างคล้ายกับพ่อและแม่ ในขณะที่ลูกก็สามารถ (อาจ) มีลักษณะเฉพาะพิเศษของตนเองเพิ่มเติมได้ด้วยเป็นต้น เรียกการสืบทอดคุณลักษณะดังกล่าวนี้ว่า "Inheritance" และ Abstraction ประเภทที่สามารถอธิบายความจริงข้อนี้ได้ก็คือ "Generalization Aggregation" มีข้อสังเกตจาความสัมพันธ์แบบ Generalization ที่ทำให้แตกต่างจาก Aggregation คือ Generalization จะช่วยอธิบายการจำแนกประเภทของคลาสสามัญ (General) ออกเป็นคลาสพิเศษ (Special) ใดๆหรือในทางกลับ Generalization จะช่วยรวมเอาลักษณะร่วมกันของคลาสพิเศษใดๆ เข้าด้วยกัน เพื่อสร้างเป็นคลาสใหม่ที่มีลักษณะเป็นสามัญ (General) ได้

2.1.6 UML

UML (Unified Modeling Language) คือภาษารูปภาพหรือสัญลักษณ์ (Graphical Language) ที่ใช้เพื่อถ่ายทอดความคิดของเราที่มีต่อระบบออกมาเป็นแผนภาพ ซึ่งประกอบไปด้วยรูปภาพหรือสัญลักษณ์มากมายตามกฎในการสร้างแผนภาพนั้น กล่าวง่ายๆก็คือ "UML เป็นภาษาสำหรับสร้างแบบจำลองของระบบ" ที่ได้รับการพัฒนาขึ้นมาเพื่อใช้ในการวิเคราะห์และออกแบบระบบเชิงวัตถุโดยเฉพาะ

แน่นอนว่าการบรรยายภาพรวมของระบบที่มีขนาดใหญ่และซับซ้อนนั้น หากใช้รูปภาพหรือสัญลักษณ์ย่อมทำให้ผู้อื่นเข้าใจได้ง่ายกว่าการบรรยายด้วยข้อความเพียงอย่างเดียว ดังนั้น "ภาษารูปภาพ (Graphical Language)" จึงได้รับการคิดค้นขึ้นมาเพื่อให้เป็นภาษาสำหรับสร้างแบบจำลอง (Modeling Language) UML จึงจัดว่าเป็นภาษารูปภาพชนิดหนึ่งเพื่อการสร้างแบบจำลองนั่นเอง ตัวอย่างภาษารูปภาพชนิดอื่น เช่น XML-GL ใช้จำลองโครงสร้างการสอบถามข้อมูล (Query) ใน XML หรือ Molecular Interaction Map (MIM) ที่ใช้จำลองโครงสร้างโมเลกุลทางชีววิทยา เป็นต้น

การที่ UML ถูกเรียกเป็นภาษา เนื่องจากภาษาโดยทั่วไปนั้น จะต้องมีโครงสร้างที่สำคัญ 2 ส่วน ได้แก่ "คำศัพท์ (Vocabulary)" และ "ไวยากรณ์ (Syntax)" ซึ่ง UML ก็มีโครงสร้างทั้ง 2 อย่างครบถ้วน โดยที่ "คำศัพท์" ของ UML จะมีทั้งการแสดงให้เห็นว่าคำศัพท์คำนั้น ชื่ออะไร มีรูปร่างลักษณะอย่างไร (ซึ่งก็คือสัญลักษณ์และชื่อของสัญลักษณ์) ส่วน "ไวยากรณ์" ใช้เป็นข้อกำหนดในการให้ความหมายแก่คำศัพท์และการนำคำศัพท์ใดๆ มาประกอบรวมเข้าด้วยกัน หรือกล่าวง่ายๆก็คือ ไวยากรณ์หมายถึง ข้อกำหนดในการนำสัญลักษณ์ต่างๆมารวมกันเพื่อสร้างแบบจำลองเป็นแผนภาพชนิดต่างๆ

จุดเริ่มต้นของ UML มาจากการที่ในช่วงปี ค.ศ. 1890-1990 ซึ่งเป็นยุคแห่งการขยายตัวของหลักการเชิงวัตถุในวงการวิศวกรรมซอฟต์แวร์ และการพัฒนาระบบคอมพิวเตอร์ทั่วไป นักพัฒนาระบบหลายท่านต่างคิดค้น Methodology ที่ใช้ในการวิเคราะห์และออกแบบระบบเชิงวัตถุนี้ขึ้นมามากมาย ทั้งนี้เนื่องจากต้องการให้ใช้งานง่ายและเหมาะสมกับโครงการพัฒนาระบบที่ตนเองรับผิดชอบอยู่ ทำให้แบบจำลองที่ได้มีความแตกต่างกันออกไปไม่เป็นมาตรฐานเดียวกัน เมื่อนำไปใช้ก็ได้แบบจำลองของระบบไม่ครบถ้วนตามความต้องการ อย่างไรก็ตาม แบบจำลองของแต่ละ Methodology ก็มีข้อดีข้อเสียแตกต่างกัน ดังนั้นนักพัฒนาระบบ 3

ท่าน ได้แก่ Grady Booch, James Rumbaugh และ Ivar Jacobson จึงได้นำข้อดีในการสร้างแบบจำลองตาม Methodology ของตนมารวมกัน แล้วพัฒนาให้เป็นภาษารูปภาพ UML ด้วยหวังจะให้ UML เป็นภาษารูปภาพมาตรฐานในการสร้างแบบจำลองเชิงวัตถุ

เริ่มต้นปลายปี ค.ศ. 1994 เมื่อ James Rumbaugh ได้ร่วมกับ Grady Booch (จากบริษัท Rational Software Corporation) พัฒนา Unified Method ขึ้นมาใช้งานก่อนในปี 1995 ต่อมาได้ชักชวนให้ Ivar Jacobson เข้าร่วมด้วย ในชื่อใหม่คือ "UML" เปิดตัวต่อวงการพัฒนาระบบด้วยเวอร์ชั่น 0.9 ในปี 1996 ซึ่งได้รับการตอบรับเป็นอย่างดี ทำให้มีการพัฒนา UML เวอร์ชั่นต่อมาเรื่อยๆ พร้อมกับความร่วมมือจากบริษัทอื่นๆ มากมาย เช่น IBM, Hewlett-Packard, Microsoft, Oracle เป็นต้น นอกจากการนำข้อดีต่างๆมารวมกันแล้ว ยังได้มีการนำเทคนิคการสร้างแบบจำลอง ได้แก่ Fusion, Shlaer-Mellor และ Coad-Yourdon มาประยุกต์รวมด้วย จนกระทั่งมาถึงเวอร์ชั่น 1.0 UML ได้รับการรับรองให้เป็นภาษาสัญลักษณ์มาตรฐานสำหรับการวิเคราะห์และออกแบบระบบเชิงวัตถุจากหน่วยงานที่มีชื่อว่า "Object Management Group (OMG)" ในปลายปี 1997 และมีการพัฒนาเรื่อยมาจนถึงปัจจุบัน (กลางปี พ.ศ. 2547) คือเวอร์ชั่น 2.0

ถึงแม้ว่าบางครั้ง UML จะถูกเรียกว่าเป็นภาษารูปภาพ "มาตรฐาน" ในการสร้างแบบจำลองสำหรับการวิเคราะห์และออกแบบระบบเชิงวัตถุก็ตาม แต่ด้วยกฎของ UML ที่มีการเพิ่มส่วนขยายให้กับสัญลักษณ์เพื่อทำให้กลายเป็นสัญลักษณ์แทนสิ่งใหม่ในระบบได้ ส่งผลให้พบว่าในปัจจุบัน บางแผนภาพก็ยังมีสัญลักษณ์แทนสิ่งใดสิ่งหนึ่งต่างกัน มากบ้างน้อยบ้างในแต่ละองค์กร ทั้งนี้ ก็เพื่อต้องการสื่อสารให้ตรงกับความต้องการขององค์กรตนเองให้มากที่สุด อย่างไรก็ตาม หากทีมงานสร้างแบบจำลองด้วยแผนภาพของ UML ได้อย่างถูกต้องตามหลักการแล้ว แผนภาพดังกล่าวจะเชื่อมต่อไปยังขั้นตอนการเขียนโปรแกรมด้วยภาษาโปรแกรมเชิงวัตถุต่างๆได้อย่างง่ายดาย เช่น Java, C++, Visual Basic เป็นต้น นอกจากนี้ UML ไม่ได้ถูกกำหนดไว้ว่าจะต้องใช้กับ Object-Oriented Methodology ใด Methodology หนึ่งเพียงอย่างเดียว ดังนั้นทีมงานพัฒนาระบบจึงสามารถนำ UML ไปใช้กับ Methodology ใดก็ได้ตามความเหมาะสม

2.1.6.1 คำศัพท์ในภาษา UML

คำศัพท์ในภาษา UML จะแสดงตามกลุ่มองค์ประกอบทั้ง 3 ได้แก่ Things, Relationships และ Diagrams โดยคำศัพท์ในแต่ละกลุ่มถูกแสดงให้อยู่ในรูปของสัญลักษณ์ต่างๆที่จะนำมาประกอบกันเป็นแผนภาพ

2.1.6.1.5 Things

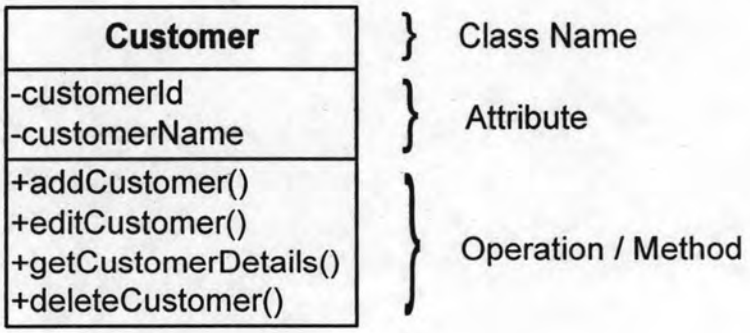
Things คือ สิ่งที่ได้จากการ Abstraction ซึ่งสามารถแบ่งออกเป็นกลุ่มตามชนิดของคำได้ ดังนี้

1) Structural Things

Structural Things คือ คำนาม (Nouns) ในภาษา UML (เมื่อเทียบกับภาษาทั่วไป) คำนามเหล่านี้จะไปปรากฏอยู่ในแบบจำลองแต่จะถูกแสดงแทนด้วยสัญลักษณ์ต่างๆดังนั้นจึงกล่าวได้ว่าสิ่งที่เกิดขึ้นในกระบวนการ Abstraction สิ่งใดที่มีชื่อเป็นคำนาม สิ่งนั้นคือ Structural Things ในภาษา UML Structural Things แบ่งออกได้ ดังนี้

- คลาส (Class)

คือกลุ่มของอ็อบเจ็กต์ที่มีคุณลักษณะ ความสัมพันธ์ และพฤติกรรม (หรือการปฏิบัติการ) ร่วมกัน สัญลักษณ์ของ"คลาส"จะเป็นรูปสี่เหลี่ยม แบ่งออกเป็น 3 ส่วน ได้แก่ ชื่อคลาส (Class Name) แอททริบิวต์ (Attribute) และการดำเนินการ (Operation หรือ Method) ดังรูป



รูปที่ 2.10 สัญลักษณ์ "Class"

- ยูสเคส (Use Case)

สิ่งที่ใช้อธิบายถึงกิจกรรมของระบบที่เกิดขึ้นตามลำดับขั้นตอน อันจะส่งผลตอบสนองต่อผู้กระทำต่อระบบ (Actor) สัญลักษณ์ของ Use Case จะใช้รูปวงรี และเขียนชื่อ Use Case ไว้ในวงรี ดังรูป

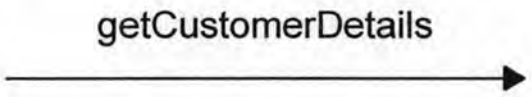
รูปที่ 2.11 สัญลักษณ์ "Use Case"

2) Behavioral Things

คือคำกริยา (Verbs) ในภาษา UML (เมื่อเทียบกับภาษาทั่วไป) Behavioral Things เป็นส่วนประกอบประเภท Dynamic Part ของแบบจำลอง กล่าวคือ เป็นสิ่งที่อาจมีการเปลี่ยนแปลงสถานะได้เมื่อเกิดเหตุการณ์ใดเหตุการณ์หนึ่ง หรือกล่าวง่ายๆ ก็คือ ส่วนที่แสดงพฤติกรรมของระบบ (ในขณะที่ Structural Things เป็นประเภท Static Part ของแบบจำลอง) Behavioral Things ใน UML มี 2 ชนิด คือ Interaction และ State Machine

- อินเตอร์แอคชั่น (Interaction)

เป็นพฤติกรรมในการส่ง Message สื่อสารกันระหว่างออบเจ็กต์ เพื่อร่วมกันทำกิจกรรมใดกิจกรรมหนึ่ง สามารถระบุ Operation หรือพฤติกรรมใดๆของอ็อบเจ็กต์โดยใช้ Interaction ได้ นอกจากนี้ยังสามารถแสดง Message ที่ส่งระหว่างอ็อบเจ็กต์ไปกับสัญลักษณ์ Interaction ได้ด้วย โดยสัญลักษณ์ของ Interaction จะใช้เส้นลูกศร พร้อมกับเขียนชื่อ Operation หรือ Message ไว้บนเส้นลูกศร ดังรูป



รูปที่ 2.12 สัญลักษณ์ "Interaction"

- สเตทแมชชีน (State Machine)

เป็นพฤติกรรมที่แสดงให้เห็นถึงลำดับการเปลี่ยนสถานะของอ็อบเจ็กต์ในช่วงระยะเวลาของการตอบสนองต่อเหตุการณ์ใดเหตุการณ์หนึ่ง สามารถระบุชื่อคลาสหรือ Collaboration ที่เกี่ยวข้องกับเหตุการณ์ที่ทำให้สถานะของอ็อบเจ็กต์เปลี่ยนไว้ในสัญลักษณ์ State Machine ได้ การเกิด State Machine มีความเกี่ยวข้องกับการเปลี่ยนสถานะ (Transition) เหตุการณ์ (Event) และการกระทำ (Activity) สัญลักษณ์ที่ใช้แทน State Machine คือรูปสี่เหลี่ยมผืนผ้ามุมมน พร้อมกับเขียนชื่อ State ไว้ด้านใน ดังรูป

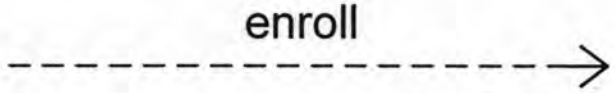
รูปที่ 2.13 สัญลักษณ์ "State Machine"

2.1.6.1.6 Relationships

องค์ประกอบส่วนที่ 2 ของ UML คือ Relationship หรือความสัมพันธ์ที่ทำหน้าที่เชื่อมกลุ่มคำต่างๆของภาษา UML เข้าด้วยกัน ซึ่งก็คือ เชื่อมโยง Things ต่างๆเข้าด้วยกัน ตามชนิดของความสัมพันธ์ของภาษา UML ซึ่งมีด้วยกัน 4 ชนิด คือ Dependency, Association, Generalization และ Realization

- Dependency

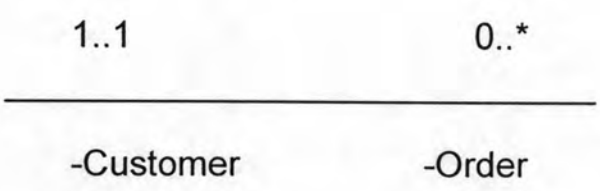
อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่งแบบส่งผลกระทบต่อกัน โดยหากมีการเปลี่ยนแปลงในสิ่งหนึ่งจะส่งผลกระทบต่อสิ่งหนึ่ง เช่น การเปลี่ยนแปลงโครงสร้างของคลาสย่อมส่งผลกระทบต่ออ็อบเจ็กต์ของคลาสนั้น สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบ Dependency คือ เส้นประหัวลูกศรแบบก้างปลา โดยที่อาจมีการเขียนลักษณะความสัมพันธ์ไว้บนเส้นก็ได้ ดังรูป



รูปที่ 2.14 สัญลักษณ์ "Dependency"

- Association

อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่งที่มีระนาบความสัมพันธ์เดียวกัน (มีความสำคัญเทียบเท่ากัน ไม่มีสิ่งใดสำคัญกว่าสิ่งใด) เช่น ความสัมพันธ์ระหว่างลูกค้ากับใบสั่งซื้อ เป็นต้น สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบ Association คือ เส้นตรง และมีข้อความแสดงบทบาทความสัมพันธ์ไว้บนเส้นตรง หรืออาจเพิ่ม Multiplicity ก็ได้ (Multiplicity หมายถึง ค่าของจำนวนสมาชิกในคลาสที่มีส่วนร่วมในความสัมพันธ์ มีรูปแบบคือ Minimum...Maximum เช่น 0..* หรือ 1..1 เป็นต้น) ดังรูป



รูปที่ 2.15 สัญลักษณ์ "Association"

นอกจากนี้ยังมี Relationship ที่เปลี่ยนรูปจาก Association เพิ่มอีก 2 แบบ คือ "Aggregation Relationship" และ "Composition Relationship" เป็นความสัมพันธ์แบบต่างระดับ คือมีลักษณะเป็น "องค์ประกอบ (Part-of)" สัญลักษณ์ของ Aggregation และ Composition มีลักษณะดังรูป

รูปที่ 2.16 สัญลักษณ์ "Composition"

รูปที่ 2.17 สัญลักษณ์ "Aggregation"

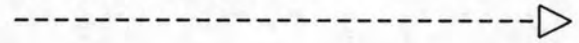
- Generalization

หรือ Specialization / Generalization อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่ง แบบจำแนกประเภท (เป็นความสัมพันธ์ต่างระดับ) หรือเป็นความสัมพันธ์ระหว่างคลาสแบบจำแนกประเภท (Type-of, Kind-of, Is-a) คลาสที่เป็นประเภทหรือชนิดใดชนิดหนึ่ง (Specialized) จะเรียกว่าเป็น "Child Class/Subclass" ที่มีคุณลักษณะและพฤติกรรมร่วมกับคลาสที่เป็นประเภททั่วไป (Generalized) ซึ่งถูกเรียกว่า "Parent Class/Superclass" สัญลักษณ์ที่ใช้แทน Generalization คือ เส้นตรงหัวลูกศรโปร่ง และหันลูกศรไปยังคลาสที่เป็น Superclass ดังรูป

รูปที่ 2.18 สัญลักษณ์ "Generalization Relationship"

- Realization

อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่ง โดยที่สิ่งหนึ่งจะทำหน้าที่ในการดำเนินการตาม Method ที่อีกสิ่งหนึ่งได้ประกาศไว้ ใน UML มี Realization 2 ประเภทคือ Realization ระหว่างคลาสกับอินเตอร์เฟส และ Realization ระหว่างคอมโพเนนท์กับอินเตอร์เฟส สัญลักษณ์ที่ใช้แทน Realization คือ เส้นประลูกศรโปร่ง ดังรูป



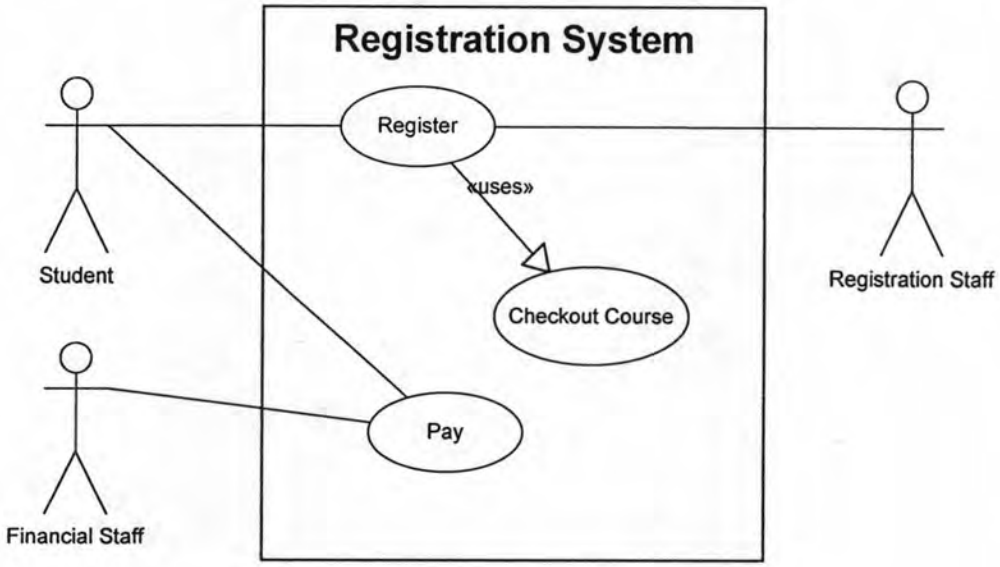
รูปที่ 2.19 สัญลักษณ์ "Realization Relationship"

2.1.6.1.7 Diagram

องค์ประกอบส่วนที่ 3 ของ UML คือ "ไดอะแกรม (Diagram)" หรือ "แผนภาพ" เป็นส่วนที่ทำหน้าที่รวบรวม Things และ Relationships เข้าไว้ในที่เดียวกัน ซึ่งหากเปรียบเทียบกับภาษาทั่วไปแล้ว Diagram ก็คือ ประโยคที่เกิดจากการรวมคำศัพท์ (Things และ Relationship) ต่างๆเข้าไว้ด้วยกัน ได้แก่

1) Use Case Diagram

เป็นแผนภาพที่ใช้แสดงถึงขั้นตอนการทำงานที่สำคัญของระบบ (Use Case) อาจกล่าวได้ว่าเป็น หน้าที่หรืองานที่ระบบจะต้องปฏิบัติ เพื่อตอบสนองต่อผู้กระทำต่อระบบ (Actor) โดย Use Case Diagram จะแสดงความสัมพันธ์ระหว่าง Use Case และ Actor จัดว่าเป็นคลาสพิเศษ แสดงตัวอย่าง Use Case Diagram ดังรูป



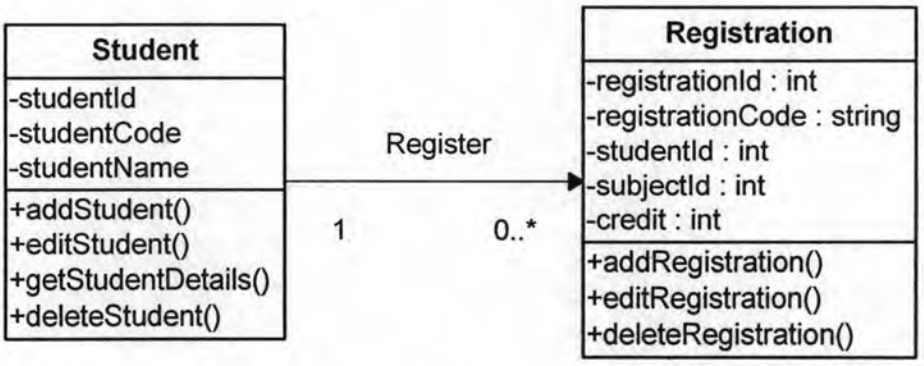
รูปที่ 2.20 ตัวอย่าง Use Case Diagram

Use Case Diagram จะประกอบไปด้วย

- Use Case คือ หน้าที่แต่ละหน้าที่ที่ระบบจะต้องปฏิบัติ ใช้สัญลักษณ์ "วงรี"
- Actor คือ ผู้กระทำต่อระบบ ใช้สัญลักษณ์ "รูปคน"
- System Boundary คือ เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ ใช้สี่เหลี่ยมเป็นสัญลักษณ์
- Relationship คือ ความสัมพันธ์ระหว่าง Use Case ใช้เส้นลูกศร และเขียน Stereotype <<...>> ที่บอกให้ทราบถึงชนิดของความสัมพันธ์ตรงกึ่งกลางเส้นลูกศรด้วย โดยความสัมพันธ์ระหว่าง Use Case มี 2 ลักษณะ ได้แก่ Include และ Use

2) Class Diagram

เป็นแผนภาพที่ใช้ในการแสดงกลุ่มของคลาส โครงสร้างของคลาส อินเตอร์เฟส (Interface) และแสดงความสัมพันธ์ (Relationship) ระหว่างคลาส ซึ่งแผนภาพนี้เป็นแผนภาพที่จะพบมากที่สุดในทาง Object Orientation แสดงตัวอย่าง Class Diagram ดังรูป

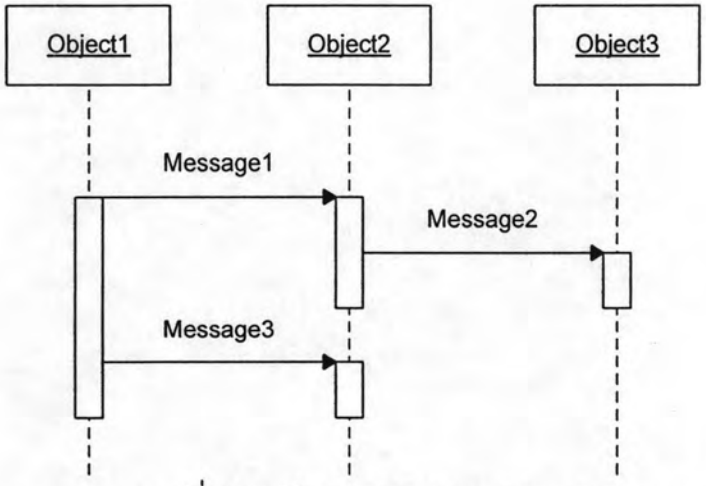


รูปที่ 2.21 ตัวอย่าง Class Diagram

ตามที่เคยกล่าวไว้แล้วว่าสัญลักษณ์แทน Class นั้นจะใช้รูปสี่เหลี่ยมแบ่งออกเป็น 3 ส่วน คือ ส่วนบน ให้แสดงชื่อคลาส (เป็นตัวหนาและขึ้นต้นด้วยตัวอักษรตัวใหญ่เสมอในทุกๆคำ) ส่วนกลางแสดง Attribute (คำแรกขึ้นต้นด้วยอักษรตัวพิมพ์เล็ก ส่วนคำต่อไปขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่) และส่วนล่างแสดง Operation / Method (เช่นเดียวกับการเขียน Attribute คือ คำแรกขึ้นต้นด้วยอักษรตัวพิมพ์เล็ก ส่วนคำต่อไปขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่) จากรูปข้างต้น แสดงให้เห็นถึงการนำ Relationship มารวมเอาไว้ในแผนภาพ นั่นคือ Association จะเห็นว่ามีการเขียนถึงบทบาทความสัมพันธ์ (Make) โดยใช้ลูกศรชี้ไปในทิศทางของคลาสที่มีความสัมพันธ์ด้วยและยังแสดงให้เห็นถึง Multiplicity อีกด้วย (1..1 และ 0..*)

3) Sequence Diagram

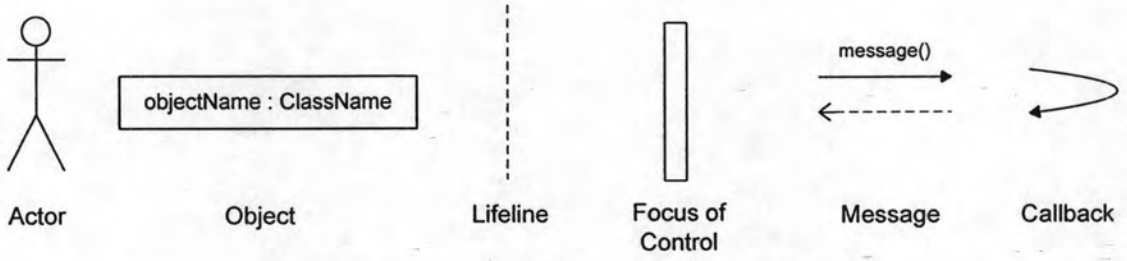
เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจ็กต์โดยเฉพาะการส่ง Message ระหว่างอ็อบเจ็กต์ตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้นจากน้อยไปมาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน แสดงลักษณะของ Sequence Diagram ดังรูป



รูปที่ 2.22 ลักษณะของ Sequence Diagram

จากรูป Sequence Diagram ประกอบไปด้วยสัญลักษณ์ต่างๆ ดังนี้

- Actor คือ ผู้กระทำต่อระบบ
- Object คือ อ็อบเจ็กต์ที่ต้องทำหน้าที่
- Lifeline คือ เส้นแสดงชีวิตของอ็อบเจ็กต์หรือคลาส
- Focus on Control / Activation คือ จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่
- A Message คือ คำสั่งหรือฟังก์ชันที่คลาสหนึ่งส่งให้อีกคลาสหนึ่ง ซึ่งสามารถส่งกลับได้ด้วย

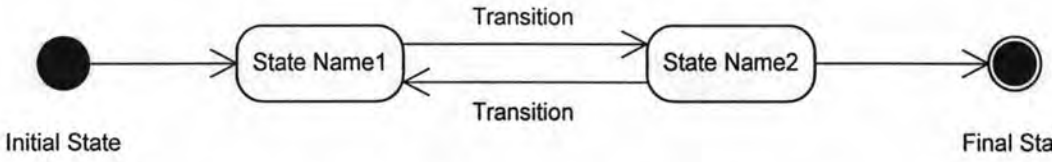


รูปที่ 2.23 สัญลักษณ์ภายใน Sequence Diagram

4) Statechart Diagram

เป็นแผนภาพที่แสดงให้เห็นพฤติกรรมของอ็อบเจ็กต์เช่นเดียวกับแผนภาพในกลุ่ม Behavioral Diagram อื่นๆ แต่ Statechart Diagram จะเน้นที่การแสดงให้เห็นถึงสถานะ (State)

การเปลี่ยนสถานะ (Transition) ที่มีต่อเหตุการณ์ (Event) ทีเกิดขึ้นในช่วงชีวิตของอ็อบเจ็กต์ 1 ช่วง (1 Sequence) แสดงลักษณะของ Statechart Diagram ดังรูป



รูปที่ 2.24 ลักษณะของ Statechart Diagram

สัญลักษณ์ที่ปรากฏอยู่ใน Statechart Diagram มีดังนี้

- Initial State คือ จุดเริ่มต้นการเปลี่ยนสถานะ
- Final State คือ จุดสิ้นสุดของการเปลี่ยนสถานะ
- Transition คือ เส้นกระตุ้นให้เปลี่ยนสถานะ
- State คือ สถานะของอ็อบเจ็กต์

2.1.6.2 ข้อดีและข้อเสียของภาษา UML

<u>ข้อดี</u>

1) UML สามารถสะท้อนภาพของระบบได้ใกล้เคียงกับโลกของความเป็นจริงมากที่สุด จึงทำให้เป็นเรื่องง่ายที่จะทำความเข้าใจ
2) UML เป็นภาษาที่มีแบบแผนแน่นอนและเป็นหนึ่งเดียว ไม่ว่าใครก็ตามที่มีส่วนเกี่ยวข้องกับระบบก็สามารถอ่านและทำความเข้าใจแบบจำลองระบบที่สร้างด้วยภาษา UML ตัวเดียวกันได้ในทิศทางเดียวกัน และเข้าใจตรงกัน
3) UML สามารถใช้เป็นเครื่องมือในการพัฒนาระบบได้ทั้งกระบวนการนับตั้งแต่การสรุปความต้องการการวิเคราะห์ความต้องการ การออกแบบระบบ และยังใช้เป็นเครื่องมือเพื่อชี้แนะแนวทางในการเขียนโปรแกรมได้อีกด้วย
4) สามารถหาซอฟต์แวร์ที่สนับสนุนการสร้างแบบจำลองภาษา UML ตามท้องตลาดได้ง่าย

ข้อเสีย

1) กรณีที่เป็นระบบงานขนาดใหญ่และจำเป็นต้องมีการระบุหมายเหตุ (Note) ไว้ด้วย จะทำให้แผนภาพดูรกและยุ่งเหยิงจนเกินไป
2) Business Rule หรือเงื่อนไขทางธุรกิจไม่ได้ถูกจัดให้อยู่รวมกันเป็นกลุ่มในที่เดียวกัน แต่กลับกระจัดกระจายกันอยู่ตามแผนภาพชนิดต่างๆ ทำให้การตรวจสอบเงื่อนไขทางธุรกิจที่เกี่ยวข้องกันเป็นไปด้วยความยากลำบาก
3) ไม่สามารถตรวจสอบความสอดคล้องกันของแผนภาพแต่ละชนิดได้ (Consistency Checking)

2.2 งานวิจัยที่เกี่ยวข้อง

ความเป็นมาของการจัดการข้อมูลผลิตภัณฑ์ เริ่มขึ้นในช่วงปี 1990 ที่มีแนวคิดในการแข่งขันโดยใช้วิศวกรรมคู่ขนาน (Concurrent Engineering) ซึ่ง Mills et al. (1991) Davis และ Trapp (1991) Cleetus (1992) Nagy et al. (1992) และ Miller (1993) ได้กล่าวว่าวิศวกรรมคู่ขนาน ถูกยอมรับว่าเป็นแนวทางในการจัดการเพื่อรองรับการแข่งขันในระดับโลกสำหรับอุตสาหกรรม ซึ่งวิศวกรรมคู่ขนานเป็นกระบวนการที่รวบรวมและร่วมกันทำของผู้ใช้ที่แตกต่างกันเพื่อร่วมกันออกแบบผลิตภัณฑ์และพัฒนากระบวนการที่เกี่ยวข้องผ่านการประสานงาน การสื่อสารและการควบคุม ในช่วงปี 1991 – 1992 Trapp(1991) และ Karinthi et al. (1992) ได้สรุปปัจจัยที่ทำให้วิศวกรรมคู่ขนานประสบความสำเร็จขึ้นอยู่กับความเข้าใจและการใช้ข้อมูลที่เกี่ยวข้องกับผลิตภัณฑ์และกระบวนการร่วมกัน ตลอดช่วงการพัฒนา

Bradley และ Agogino (1991) กระบวนการของวิศวกรรมคู่ขนานมีแนวคิดในการใช้เครื่องมือมาช่วยให้สามารถดำเนินงานได้ง่ายขึ้น โดยเครื่องมือหรือโปรแกรมนั้นจะช่วยในการจัดการกิจกรรมการส่งข้อมูลของผลิตภัณฑ์และติดตามผลิตภัณฑ์ตั้งแต่เริ่มต้นจบกระทั่งละทิ้ง และในเวลาเดียวกันสนับสนุนให้การสื่อสารเป็นไปอย่างมีประสิทธิภาพ และการใช้ข้อมูลร่วมกันของกลุ่มงานเดียวกันและในแต่ละทีมพัฒนา Miller (1993) Stover (1993) McIntosh (1995) Chen และ Hsiao (1997) ได้กล่าวถึงเครื่องมือหรือโปรแกรมที่สนับสนุนกระบวนการดำเนินงานของวิศวกรรมคู่ขนาน ซึ่งเครื่องมือนี้เป็นที่รู้จักกันในชื่อ การจัดการข้อมูลผลิตภัณฑ์ (Product Data Management: PDM) และ การจัดการข้อมูลทางวิศวกรรม (Engineering Data Management: EDM)

ระบบจัดการข้อมูลผลิตภัณฑ์ที่มีอยู่ขายอยู่ส่วนมากมีองค์ประกอบสำคัญโดยทั่วไปคือ การจัดการการไหลของงาน (Workflow) และกระบวนการ, การจัดการโครงงาน, การจัดการการเปลี่ยนแปลงและ การจัดการโครงสร้างผลิตภัณฑ์ ซึ่งการจัดการข้อมูลผลิตภัณฑ์จะรู้จักในหลากหลายชื่อ แต่ทั้งหมดก็ใช้แนวคิดเดียวกันในการจัดหาแนวทางโครงสร้างที่จะจัดเก็บ, รวบรวม, จัดการ และควบคุมข้อมูลและกระบวนการทางวิศวกรรมจากการออกแบบ, การผลิตจนถึงการกระจายสินค้าให้เป็นระเบียบและมีประสิทธิภาพ

Peltonen et al. (1996) และ1996Lebovitz (1997) ได้กล่าวถึงระบบจัดการข้อมูลผลิตภัณฑ์ว่าระบบจัดการข้อมูลผลิตภัณฑ์ใช้ในการควบคุมข้อมูล, ไฟล์ (Files), เอกสารและกระบวนงาน เพื่อรองรับการออกแบบ, การสร้าง, การสนับสนุน, การกระจาย และการบำรุงรักษาผลิตภัณฑ์ โดยพื้นฐานข้อมูลที่เกี่ยวกับผลิตภัณฑ์ประกอบด้วย ขนาด, แบบงานทาง

วิศวกรรม, แผนงาน, ไฟล์ชิ้นส่วน, แผนภาพการประกอบ, ข้อจำกัดของผลิตภัณฑ์, ชุดคำสั่งควบคุมเครื่องจักร, ผลการวิเคราะห์, สูตรการผลิต, การเปลี่ยนแปลงทางวิศวกรรม และอื่นๆ ซึ่งการจัดการข้อมูลผลิตภัณฑ์เป็นเครื่องมือที่ใช้สำหรับบูรณาการเชื่อมโยงหลายๆ ส่วนเข้าด้วยกัน เพื่อให้มั่นใจว่าสามารถให้ข้อมูลที่ถูกต้องกับผู้ใช้ที่ถูกต้องได้ทันเวลาในรูปแบบที่สามารถนำไปใช้ได้ตลอดทั้งองค์กร

Tony และ William (2001) กล่าวว่าในช่วงปี 2000 ได้มีแนวคิดเกี่ยวกับการปรับปรุงช่วงชีวิตของผลิตภัณฑ์ ที่สนใจถึงเวลานำผลิตภัณฑ์ออกสู่ตลาดให้สั้นลงเพื่อใช้เป็นกลยุทธ์ในการแข่งขันทางการตลาด จากแนวคิดนี้ส่งผลให้มีการพัฒนาระบบจัดการข้อมูลผลิตภัณฑ์ให้สนับสนุนข้อมูลตลอดช่วงชีวิตผลิตภัณฑ์จากเริ่มต้นแนวคิดจนถึงผลิตภัณฑ์สำเร็จรูป ซึ่งช่วยให้สามารถระบุถึงกระบวนการเปลี่ยนแปลงทางวิศวกรรมได้ดีขึ้น ด้วยการจัดการที่ต่างกันในแต่ละระดับของช่วงการออกแบบของผลิตภัณฑ์

การศึกษาถึงกระบวนการในการออกแบบระบบจัดการข้อมูลผลิตภัณฑ์ได้มีหลายงานวิจัยที่แสดงให้เห็นว่าการออกแบบเชิงวัตถุเป็นวิธีการที่มีประสิทธิภาพในการจำลองในมุมมองต่างๆ ของกระบวนการภายในองค์กร Medina-Mora et al. (1992) ได้มีศึกษาความแตกต่างสามมุมมองของกระบวนการภายในองค์กร ซึ่งประกอบด้วย วัตถุดิบ, ข้อมูล หรือกระบวนงานทางธุรกิจ การศึกษาที่ใช้แนวคิดเชิงวัตถุในการจำลองกระบวนงานในมุมมองของข้อมูลเป็นแนวคิดหลัก ซึ่งศึกษาโดย Rambaugh et al. (1991), Martin (1993) และ Larman (1998) งานวิจัยที่แสดงถึงการใช้แนวคิดเชิงวัตถุในการจำลองกระบวนงานในมุมมองของวัตถุดิบได้อย่างมีประสิทธิภาพศึกษาโดย Park et al. (1997), Choi et al. (1996) และ Narayanan et al. (1998) และในส่วนของการจำลองในมุมมองของกระบวนงานทางธุรกิจโดยใช้แนวคิดเชิงวัตถุประกอบด้วย Eriksson, Penker (2000), Marshall (2000) และ Shan, Earle (1998)

จากการศึกษาจะเห็นว่าแนวคิดเชิงวัตถุเป็นเครื่องมือที่เหมาะสมในการจำลองกระบวนการขององค์กรในหลายๆ มุมมอง และ ในปี 1997 Object Management Group (OMG) ได้ยอมรับให้ Unified Modeling Language (UML) เป็นเครื่องมือมาตรฐานในการจำลองแนวคิดเชิงวัตถุ (Object-Oriented Modeling) จึงได้มีแนวคิดที่จะนำเอาการออกแบบแนวคิดเชิงวัตถุโดยใช้ UML เป็นเครื่องมือในการวิเคราะห์และออกแบบระบบจัดการข้อมูลผลิตภัณฑ์ในงานวิจัยนี้ ซึ่งจะกล่าวในบทต่อไป