บทที่ 2
ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

2.1 การจัดตารางเวลา (Scheduling)

การจัดตารางเวลาได้มีการพัฒนามาอย่างมากในช่วงเวลา 40 ปีที่ผ่านมา Baker (1974) ได้ให้คำนิยามของการจัดตารางเวลางานว่า "เป็นการจัดสรรทรัพยากรที่มีอยู่จำกัดในช่วงเวลาใดเวลาหนึ่ง ให้กับชุดของงานอย่างมีประสิทธิภาพ" ซึ่งผลงานที่ผ่านมาได้มีการประยุกต์ทฤษฎีทางด้านการจัดตารางเวลากับงานประเภทต่างๆอย่างมาก

ทฤษฎีทางด้านการจัดตารางเวลาได้พัฒนาจากแบบจำลองทางคณิตศาสตร์โดยการนำเทคนิคในการหาคำตอบต่างๆมาใช้ในการจัดตารางเวลาของงานที่มีข้อจำกัดอยู่ 2 แบบ คือ ทรัพยากรที่มีอยู่อย่างจำกัด และ.ลักษณะของงานที่ต้องดำเนินการ ทำให้แบบจำลองทางด้านการจัดตารางเวลาเป็นแบบจำลองช่วยในการตัดสินใจเชิงปริมาณ (Quantitative approach)

ปัญหาของงานด้านการจัดตารางเวลามักกำหนดเป้าหมายของการตัดสินใจด้านการจัดตารางเวลา

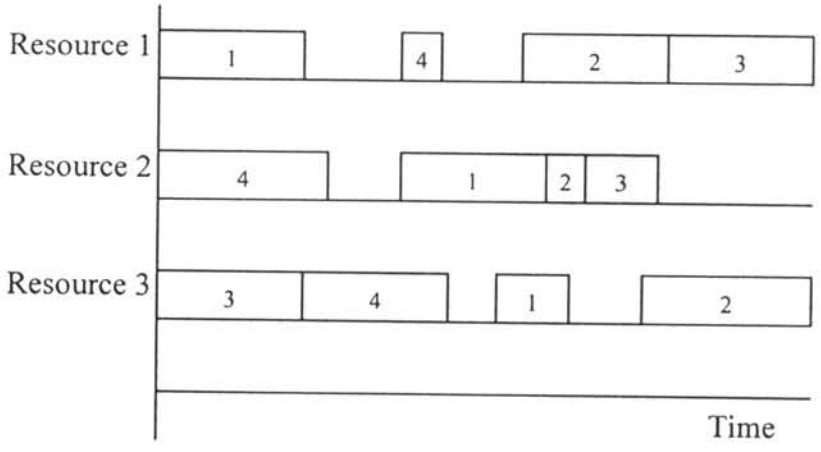
1. ประสิทธิภาพในการใช้ทรัพยากรที่มีอยู่ (Resource Utilization)
2. ความสามารถในการตอบสนองได้ทันเวลา
3. การดำเนินงานให้เสร็จทันตามเวลาที่กำหนด

ทำให้ปัญหาของการจัดตารางเวลาแบ่งออกได้เป็น 2 ลักษณะ คือ

1. การตัดสินใจในการจัดสรรทรัพยากร (Allocation decision)
2. การตัดสินใจในการจัดลำดับของงาน (Sequencing decision)

การกำหนดจุดประสงค์ของการจัดตางเวลาในทางทฤษฎีควรคำนึงถึงค่าใช้จ่ายที่เกิดขึ้นในระบบหรือปัจจัยอื่นๆ แต่ในทางเป็นจริงแล้วการระบุหรือว่า การกำหนดการทำงานตามปัจจัยนั้นๆค่อนข้างยาก จึงมักกำหนดจุดประสงค์ใหม่เป็นประสิทธิภาพของระบบแทน

การจำลองปัญหางานทางการจัดตารางเวลาส่วนใหญ่ กำหนดทรัพยากร (Resource) แทนด้วยเครื่องจักรหรือบุคลากร และลักษณะของงานที่ต้องดำเนินการทำงานแทนด้วยงาน (Job) ซึ่งส่วนใหญ่เล้วนิยมแสดงการจัดตารางงานด้วน Gantt Chart ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่าง Gantt Chart การจัดตารางเวลาในการทำงาน

Pinedo (1995) ได้แบ่งชนิดของการจัดตารางเวลาออกเป็น 2 แบบ คือ

1. Deterministic Models เป็นการจัดตารางเวลาที่สามารถรู้ลักษณะและปริมาณของทรัพยากรที่แน่นอนและลักษณะของงานจะไม่เปลี่ยนแปลงตามเวลา
2. Stochastic Models เป็นการจัดตารางเวลาที่ระบบแบบเกิดความไม่แน่นอนต่างๆ ที่ไม่สามารถทราบล่วงหน้าได้ เช่น เครื่องจักรเสีย หรือ เกิดงานที่มีความสำคัญสูงถูกป้อนเข้ามาในระบบ เป็นต้น

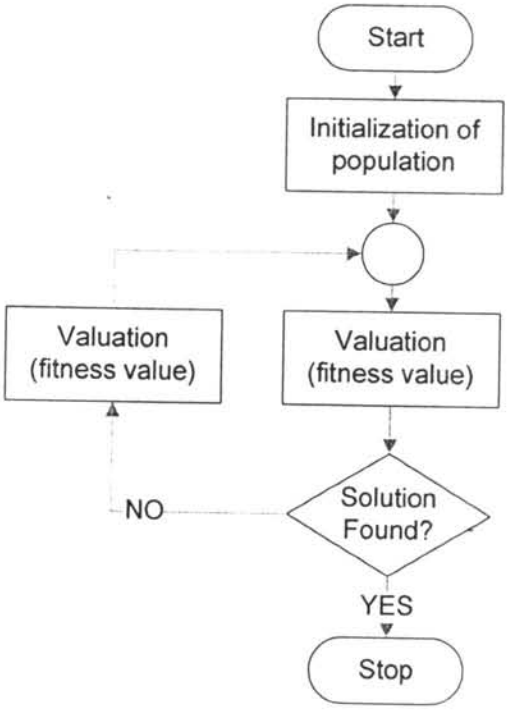
2.1.2 เทคนิคในการหาคำตอบ

เทคนิคในการหาคำตอบของด้วยวิธีการปรับปรุงด้วยการวนซ้ำ (Iterative improvement) ได้เริ่มถูกนำมาใช้ในการแก้ปัญหาด้านการจัดตารางเวลาอย่างมาก โดยเฉพาะงานประเภท Flow shop เนื่องจากเป็นงานที่มีความซับซ้อนและประกอบด้วยลักษณะของงานจำนวนมาก แนวความคิดในการหาคำตอบด้วยวิธีนี้ได้พัฒนามาจากวิธีการปีนเขาเพื่อหาจุดยอด โดยการค้นหาเส้นทางที่นำไปสู่ระดับความสูงที่เพิ่มขึ้นเรื่อยๆ แต่ยังเป็นไปได้ที่ยอดที่พบจะไม่ใช่จุดยอดที่สูงสุด จากปัญหาเหล่านี้จึงได้มีการพัฒนาวิธีการค้นหาแบบสุ่ม (Random search) ขึ้นและต่อมาได้พัฒนากลายเป็นการค้นหาคำตอบด้วยวิธีฮิวริสติกส์ (Heuristics)ปัจจุบันวิธีการที่นิยมใช้ประกอบด้วย (Girsch และ Skele , 1994)

- Genetic Algorithms

แนวคิดมาจากการเลียนแบบวิวัฒนาการต่อมาได้มีการประยุกต์วิธีนี้ใช้กับปัญหาด้านการจัดตารางเวลา โดยโครโมโซมจะแสดงเป็นตัวเลขจำนวนเต็ม แทนด้วนกลุ่มของงานที่มี

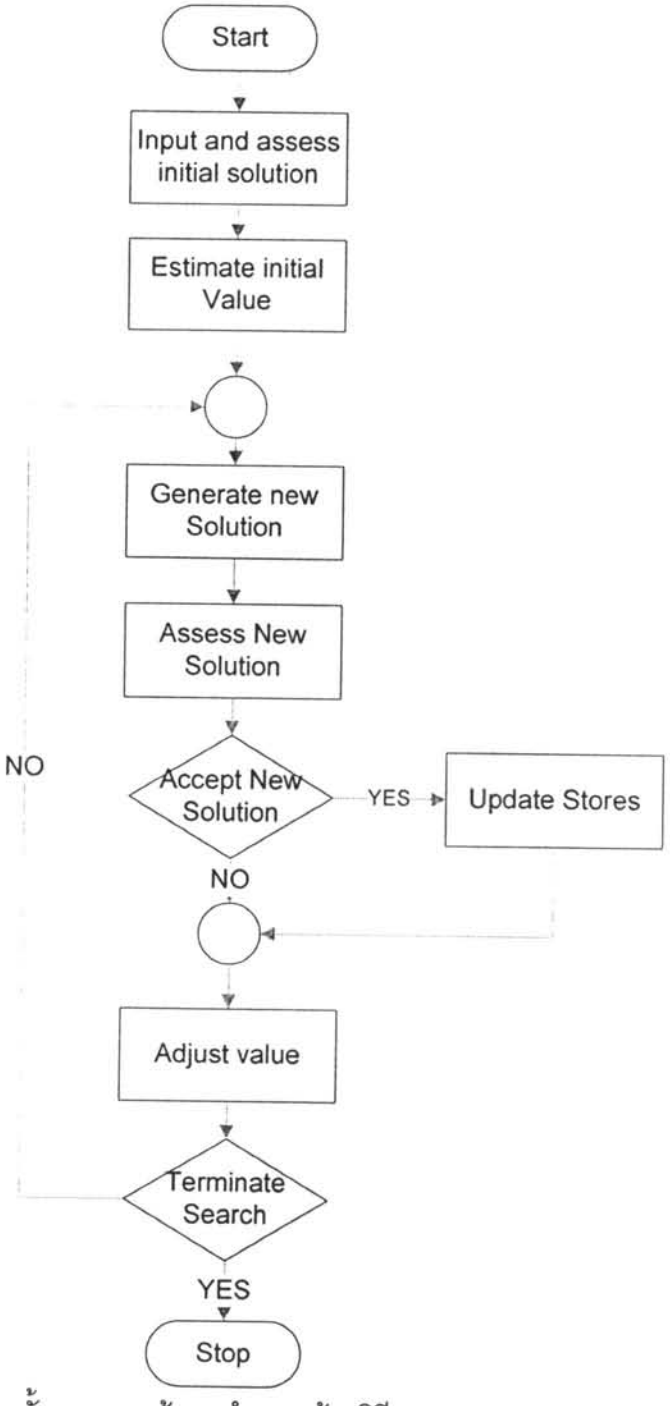
การให้ลำดับของงานและขยายพันธุ์เพื่อหาลำดับของงานที่แตกต่างกันออกไปจนได้คำตอบที่ดีที่สุด



รูปที่ 2.2 แผนผังแสดงขั้นตอนการค้นหาคำตอบด้วยวิธี Genetic Algorithms

O Simulated Annealing (SA)

วิธีการนี้มีข้อดีคือสามารถที่จะหลีกเลี่ยงคำตอบที่วนติดอยู่ในค่าที่ดีที่สุดสัมพัทธ์ (Local optimal) วิธีการจะค้นหาแบบสุ่ม และจะยอมรับการเปลี่ยนแปลงคำตอบให้ดีขึ้น เพิ่มขึ้นด้วยฟังค์ชั่นจุดประสงค์ (f) โดยการยอมรับการเปลี่ยนแปลงคำตอบที่เพิ่มขึ้นด้วยความน่าจะเป็น



รูปที่ 2.3 แผนผังแสดงขั้นตอนการค้นหาคำตอบด้วยวิธี Simulated annealing Tabu Search

วิธีนี้ได้ถูกนำมาประยุกต์ใช้เพื่อการหลีกเลี่ยงการหาค่าดีที่สุดสัมพัทธ์ (Local Optimal) เช่นเดียวกับวิธีของ Simulated Annealing โดยการเคลื่อนย้ายคำตอบในโครงสร้างของ List ที่แสดงด้วยเซตของงานทั้งหมดที่จะมาจัดตารางเวลา และเลือกรูปแบบการย้ายที่ให้ค่าคำตอบที่ดีที่สุดเพื่อมาเป็น List ใหม่ในการเตรียมที่จะย้ายคำตอบซ้ำต่อไป

2.2 การจัดการทางด้านเวชระเบียน

เวชระเบียน คือ การบันทึกเหตุการณ์กระบวนการดูและผู้ป่วยอย่างเป็นระบบ ตั้งแต่เริ่มต้นจนถึงสิ้นสุดการติดต่อระหว่างผู้ป่วยและโรงพยาบาล โดยมีส่วนเกี่ยวข้องกันไปตามกระบวนการดูและรักษาที่เกิดขึ้นจริง ข้อมูลในบันทึกเวชระเบียนประกอบด้วยข้อมูลปฐมภูมิอันแสดงลักษณะของผู้ป่วยประกอบกับความคิดเห็นและการปฏิบัติของผู้ดูแลรักษา โดยประโยชน์ของเวชระเบียนมีในส่วนต่างๆดังต่อไปนี้

1. บันทึกเพื่อการสื่อสารในทีม เวชระเบียนเป็นเครื่องมือที่ใช้ในการสื่อสารข้อมูลการดูและรักษาผู้ป่วยทั้งในวิชาชีพเดียวกัน และระหว่างวัชาชีพต่างๆที่เกี่ยวข้อง โดยเฉพาะอย่างนิ่งในผู้ป่วยที่มีความซับซ้อน ต้องการดูและรักษาจากสหสาขาวิชาชีพ
2. บันทึกเพื่อการดูแลรักษาต่อเนื่อง เวชระเบียนเป็นเครื่องมือเพื่อส่งต่อข้อมูลเพื่อการดูและรักษาผู้ป่วยอย่างต่อเนื่องทั้งในระหว่างการดูแลรักษาในโรงพยาบาลและการติดตามผลภายหลังออกจากโรงพยาบาล
3. บันทึกเพื่อการทบทวน การทบทวนเพื่อการประกันและการพัฒนาคุณภาพนั้นจำเป็นต้องอาศัยคุณภาพของบันทึกเวชระเบียนทั้งในเชิงปริมาณที่ต้องมีเนื่อหาที่จำเป็นครบถ้วน และเชิงคุณภาพซึ่งหมายความถึงความสอดคล้องต่อเนื่องของเนื้อหา ความสะดวกต่อการค้นหาข้อมูลในเวชระเบียน
4. บันทึกเพื่อเป็นหลักฐานทางกฎหมาย เวชระเบียนที่บันทึกเหตุการณ์ที่อาจนำไปสู่การร้องเรียนในภายหลังเป็นหลักฐานสำคัญที่ทางกฎหมายประกอบการพิจารณาของผู้รับผิดชอบ แต่การใช้เวชระเบียนเพื่อเป็นหลักฐานในการฟ้องร้องนั้นเป็นการแก้ไขปัญหาที่ปลายเหตุ ดังนั้นจึงควรให้ความสำคัญกับการแก้ไขปัญหาโดยการให้ความสำคัญของเวชระเบียนในส่วนประกอบที่กล่าวไว้ข้างต้นมากกว่า

2.2.1 มาตรฐานในการบันทึกเวชระเบียน (พญ.จามรี เชื้อเพชรระโสภณ)

1. การเข้าถึงการรักษาและการดูแลอย่างต่อเนื่อง เป้าหมายเพื่อให้การให้บริการต่างๆ ที่มีอยู่แล้วตอบสนองความต้องการของผู้ป่วยในด้านการดูแลได้ และช่วยให้การบริการต่างๆที่ถูกจัดให้กับผู้ป่วยในองค์กรเป็นไปในทิศทางเดียวกัน ตามมาด้วยเรื่องจองการวางแผนการจำหน่ายผู้ป่วยและ

การติดตามผล ผลที่ได้คือจะช่วยในการปรับปรุงผลการดูและรักษาผู้ป่วย แบละมีการใช้ทรัพยากรอย่างคุ้มค่ามากขึ้น

2. การประเมินอาการผู้ป่วย การประเมินอาการผู้ป่วยอย่างมีประสิทธิภาพนั้นจะส่งผลในด้านการตัดสินใจเกี่ยวกับภาวะการฉุกเฉินของผู้ป่วยหรือความจำเป็นที่จะต้องมีการรักษาอย่างเร่งด่วน รวมไปถึงความจำเป็นที่จะต้องมีการรักษาอย่างต่อเนื่องแม้ว่าในสภาพการณ์ ของผู้ป่วยจะเปลี่ยนไปก็ตาม
3. การดูแลผู้ป่วย เป้าหมายหลักขององค์กรที่ทำหน้าที่ดูแลสุขภาพนั้นคือการดูและผู้ป่วยการเตรียมการดูแลผู้ป่วยให้เหมาะกับสภาพการณ์ ช่วบเอื้อและตอบสนอความต้องการเฉพาะตัวของผู้ป่วยนั้นจำเป็นจะต้องมีการวางแผนและความร่วมมือกันอย่างมาก
4. การให้การศึกษาแก่ผู้ป่วยและครอบครัวการให้การศึกษาแก่ผู้ป่วยและครอบครัวช่วยให้ผู้ป่วยมีส่วนร่วมต่อการรักษาได้ดีขึ้น และสามารถดูแลสุขภาพของตนตามที่ได้รับคำแนะนำ เจ้าหน้าที่ขององค์กรแต่ละคนควรจะให้ความรู้ทั้งต่อผู้ป่วยและครอบครัวของผู้ป่วย โดยแพทย์หรือพยาบาลอาจจะให้ความรู้ทั้งต่อผู้ป่วยและครอบครัวของผู้ป่วย จึงเป็นเรื่องที่สำคัญที่เจ้าหน้าที่ทุกคนจะต้องร่วมมือกัน และมุ่งเน้นไปที่สิ่งที่ผู้ป่วยจำเป็นที่จะต้องรู้

2.2.2 แนวทางในการบันทึกเวชระเบียน (นายแสวง บุญเฉลิมวิภาส)

1. แนวทางของแพทย์สภา
2. ผู้ป่วยนอก ข้อมูลที่พึงปรากฏในเวชระเบียนได้แก่
   - อาการสำคัญและประวัติการเจ็บป่วยที่สำคัญ
   - ประวัติการแพ้ยา สารเคมี หรือสารอื่นๆ
   - บันทึกสัญญาณชีพ (Vital sign)
   - ผลการตรวจร่างกายผู้ป่วยที่ผิดปกติ เหรือที่มีความสำคัญต่อการวินิจฉัยโรคหรือการให้การรักษาแก่ผู้ป่วย
   - ปัญหาของผู้ป่วย หรือการวินิจฉัยโรคหรือการแยกโรค
   - การสั่งการรักษาพยาบาล รวมตลอดถึงชนิดของยาและจำนวน
   - ในกรณีมีการทำหัตการควรมี
     - บันทึกเหตุผล ความจำเป็นของการทำหัตถการ

- ใบยินยอมของผู้ป่วยหรือผู้แทน ภายหลังที่ได้รับทราบและเข้าใจถึงขั้นตอน ผลดี และอาการแทรกซ้อนที่อาจเกิดจากการทำหัตถการ

- คำแนะนำอื่นๆ ที่ให้กับผู้ป่วย

2.1 ผู้ป่วยแรกรับไว้ในสถานพยาบาล ข้อมูลที่พึงปรากฏในเวชระเบียนขณะ แรกรับผู้ป่วยได้แก่

- อาการสำคัญและประวัติการเจ็บป่วยที่สำคัญ
- ประวัติการแพ้ยา สารเคมี หรือสารอื่นๆ
- ประวัติการเจ็บป่วยในอดีตที่สำคัญ ซึ่งอาจสัมพันธ์เกี่ยวข้องกับการเจ็บป่วยในครั้งนี้
- บันทึกสัญญาณชีพ (Vital sign)
- ผลการตรวจร่างกายทุกระบบที่สำคัญ
- ปัญหาของผู้ป่วย หรือการวินิจฉัยโรค หรือการแยกโรค
- เหตุผลความจำเป็นในการรับไว้รักษาในสถานพยาบาลและแผนการดูแลรักษาผู้ป่วยต่อไป

2.2 ผู้ป่วยระหว่างนอนพักรักษาในสถานพยาบาล ข้อมูลที่พึงปรากฏในเวชระเบียนได้แก่

- บันทึกเมื่อมีการเปลี่ยนแปลงทางคลินิกของผู้ป่วยที่สำคัญระหว่างพักรักษาตัวในสถานพยาบาล
- บันทึกอาการทางคลินิก และเหตุผลเมื่อมีการสั่งการรักษาพยาบาลหรือเพิ่มเติม หรือเปลี่ยนแปลงการรักษาพยาบาล
- ผลการตรวจทางห้องปฏิบัติการ การตรวจพิเศษต่างๆ
- ในกรณีมีการทำหัตการควรมี
  - บันทึกเหตุผล ความจำเป็นของการทำหัตถการ
  - ใบยินยอมของผู้ป่วยหรือผู้แทน ภายหลังที่ได้รับทรายและเข้าใจถึงขั้นตอน ผลดี และอาการแทรกซ้อนที่อาจเกิดจากการทำหัตถการ

3. ข้อกำหนดในมาตรฐานโรงพยาบาลฉบับปีกาญจนาภิเษก

มาตรฐานเกี่ยวกับการบันทึกเวชระเบียนในเชิงผลลัพธ์ไว้ดังนี้

3.1 ให้มีการจัดทำเวชระเบียนสำหรับผู้ป่วยทุกรายที่เข้ารับบริการของโรงพยาบาล โดยมีข้อมูลและรายละเอียดเพียงพอสำหรับวัตถุประสงค์ต่อไปนี้

- ทราบว่าผู้ป่วยเป็นใคร
- ทราบเหตุผลของการรับไว้นอนในโรงพยาบาล
- มีข้อมูลสนับสนุนการวินิจฉัยโรค
- ประเมินความเหมาะสมของการดูแลรักษาผู้ป่วย
- ทราบผลลัพธ์และการเปลี่ยนแปลงที่เกิดขึ้นกับผู้ป่วย
- เอื้ออำนวยต่อการดูแลอย่างต่อเนื่องของผู้ให้บริการ
- ให้รหัสได้อย่างถูกต้อง

โดยมีนโยบายและวิธีเชิงปฏิบัติเป็นลายลักษณ์อักษรสำหรับการบันทึกและการเก็บรายงานผลในเวชระเบียน

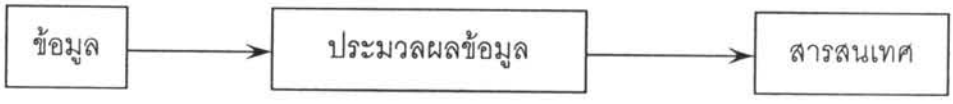
2.3 ระบบสารสนเทศ

2.3.1 นิยาม

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

ข้อมูล (data) หมายถึง ข้อเท็จจริงต่าง ๆ ที่มีอยู่ในธรรมชาติ เป็นกลุ่มสัญลักษณ์แทนปริมาณหรือการกระทำต่าง ๆ ที่ยังไม่ผ่านการประมวลผล ข้อมูลอาจจะอยู่ในรูปของตัวเลข ตัวหนังสือ และท้ายที่สุดข้อมูลก็คือ วัตถุดิบของสารสนเทศ

สารสนเทศ (information) ได้แก่ ข้อมูลต่าง ๆ ที่ได้รับการประมวลผลแล้วด้วยวิธีการต่าง ๆ เป็นความรู้ที่ต้องการสำหรับใช้ทำประโยชน์ เป็นส่วนผลลัพธ์หรือเอาต์พุตของระบบการประมวลผลข้อมูล เป็นสิ่งซึ่งสื่อความหมายให้ผู้รับเข้าใจและสามารถนำไปกระทำกิจกรรมใดกิจกรรมหนึ่งโดยเฉพาะได้ หรือเพื่อเป็นการย้ำความเข้าใจที่มีอยู่แล้วให้มีมากยิ่งขึ้น และเป็นผลลัพธ์ของระบบสารสนเทศ หรือ อาจแสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 ความสัมพันธ์ระหว่างข้อมูลและสารสนเทศ

ระบบสารสนเทศ (Information System) หมายถึง ระบบที่ประกอบด้วยคน เครื่องคอมพิวเตอร์ และอุปกรณ์ต่าง ๆ ที่ทำงานประสานกัน เพื่อจัดทำสารสนเทศสำหรับสนับสนุนการปฏิบัติงาน การจัดการ และการตัดสินใจในหน่วยงาน หรือ องค์กร

2.3.2 ประเภทของระบบสารสนเทศ

ระบบสารสนเทศสามารถจำแนกได้เป็น 2 ประเภทใหญ่ ๆ คือ ระบบสารสนเทศแบบกว้างๆ ที่ไม่ได้นำไปใช้กับงานด้านหนึ่งด้านใดโดยเฉพาะ และ ระบบสารสนเทศที่จัดทำขึ้นสำหรับใช้งานประยุกต์โดยตรง

1. ระบบสารสนเทศแบบกว้างๆ

เป็นระบบสารสนเทศที่ขยายขึ้นมาจากระบบการประมวลผลธรรมดา โดยมุ่งที่จะจัดทำรายงานสารสนเทศเพื่อให้ผู้ปฏิบัติงานและผู้บริหารใช้งาน อาจสรุปหน้าที่และประโยชน์ได้ย่อ ๆ ดังต่อไปนี้

1.1 ระบบสารสนเทศทั่วไป เป็นระบบที่สร้างขึ้นให้มีความสามารถในการประมวลผล และจัดทำรายงานที่ผู้ใช้และผู้บริหารต้องการได้

1.2 ระบบสารสนเทศเพื่อการจัดการ (Management Information System) เป็นระบบสารสนเทศที่เน้นด้านการผลิตเอกสารรายงานสำหรับผู้บริการ และมีความสามารถในการค้นหาและจัดทำรายงานพิเศษบางอย่างในแบบออนไลน์

1.3 ระบบสารสนเทศสำนักงาน (Office Information System) เป็นระบบสารสนเทศสำหรับเก็บบันทึกข้อมูลเอกสารภายในสำนักงาน และอำนวยความสะดวกในการส่งเอกสารผ่านระหว่างผู้ปฏิบัติงาน

1.4 ระบบสนับสนุนการตัดสินใจ (Decision Support System) เป็นระบบสารสนเทศสำหรับผู้บริหารในการทดสอบแนวทางเลือกในการตัดสินใจ ทำให้ทราบว่าการเลือกแนวทางเช่นนั้น ๆ จะเกิดอะไรขึ้น

1.5 ระบบสารสนเทศเพื่อผู้บริหาร (Executive Information System) เป็นระบบสารสนเทศที่ช่วยให้ผู้บริหารค้นหาข้อมูล และสารสนเทศที่สำคัญต่อการบริหารมาใช้งานได้เมื่อจำเป็น และอำนวยความสะดวกในการติดตามหารายละเอียดของข้อมูลบางรายการที่มีปัญหาได้

2. ระบบสารสนเทศที่จัดทำขึ้นสำหรับใช้งานประยุกต์โดยตรง

เป็นระบบสารสนเทศที่ใช้เฉพาะในงานประยุกต์บางด้าน ระบบสารสนเทศประเภทนี้มีมาก ขึ้นกับการคิดจัดทำและตั้งชื่อ โดยมากจะนำเอาชื่องานประยุกต์มาใช้ควบกับชื่อระบบสารสนเทศ ตัวอย่างเช่น

2.1 ระบบสารสนเทศงานบัญชี เป็นระบบสารสนเทศทั้งหมดที่เกี่ยวกับการเก็บบันทึกข้อมูลบัญชีและจัดทำรายงานบัญชี

2.2 ระบบสารสนเทศการตลาด เป็นระบบสารสนเทศสำหรับใช้เก็บรายละเอียดของผลิตภัณฑ์ลูกค้า การผลิต และอื่นๆ สำหรับช่วยในการวางแผนและส่งเสริมการตลาด

2.3 ระบบสารสนเทศในโรงพยาบาล เป็นระบบสารสนเทศสำหรับใช้ในการเก็บข้อมูลเกี่ยวกับผู้ป่วย ยา แพทย์และการรักษาพยาบาล เพื่อช่วยในการคิดเงินค่ารักษาพยาบาลและให้บริการแก่ผู้ป่วย

2.4 ระบบสารสนเทศห้องสมุด เป็นระบบสารสนเทศสำหรับใช้เก็บข้อมูลเกี่ยวกับหนังสือ และวัสดุที่เก็บรวบรวมในห้องสมุด ข้อมูลเกี่ยวกับสมาชิกผู้ยืม ข้อมูลเกี่ยวกับบริษัทผู้ขายทั้งหมด เพื่อให้งานให้บริการของห้องสมุดดำเนินไปได้อย่างมีประสิทธิภาพ

2.5 ระบบสารสนเทศทรัพยากรบุคคล เป็นระบบสารสนเศที่ใช้เก็บข้อมูลเกี่ยวกับบุคลากรของหน่วยงานและสามารถให้สารสนเทศที่เกี่ยวข้อง เช่น ด้านผลงาน ด้านการฝึกอบรมและพัฒนา ด้านสวัสดิการ ด้านสุขภาพอนามัย ด้านการดำรงตำแหน่ง

2.3.3 เป้าหมายของระบบสารสนเทศ

ระบบสารสนเทศสำหรับองค์กรต่าง ๆ โดยส่วนใหญ่แล้วมักมีเป้าหมายที่สำคัญ (ประสงค์ ปราณีตพลกรังและคณะ,2541) ดังนี้

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

การเพิ่มประสิทธิภาพในการทำงาน (Operational Efficiency) เป็นการ ช่วยให้งานที่ทำอยู่นั้นสามารถทำได้เร็วขึ้น มีความถูกต้องมากขึ้น ทำให้พนักงาน มีเวลาในการเรียนรู้งานใหม่ ๆ เพื่อเพิ่มประสิทธิภาพในการทำงาน ลักษณะที่เห็น ได้ คือ เป็นการทำสิ่งที่มีอยู่ให้ดีขึ้น (Do things better)

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

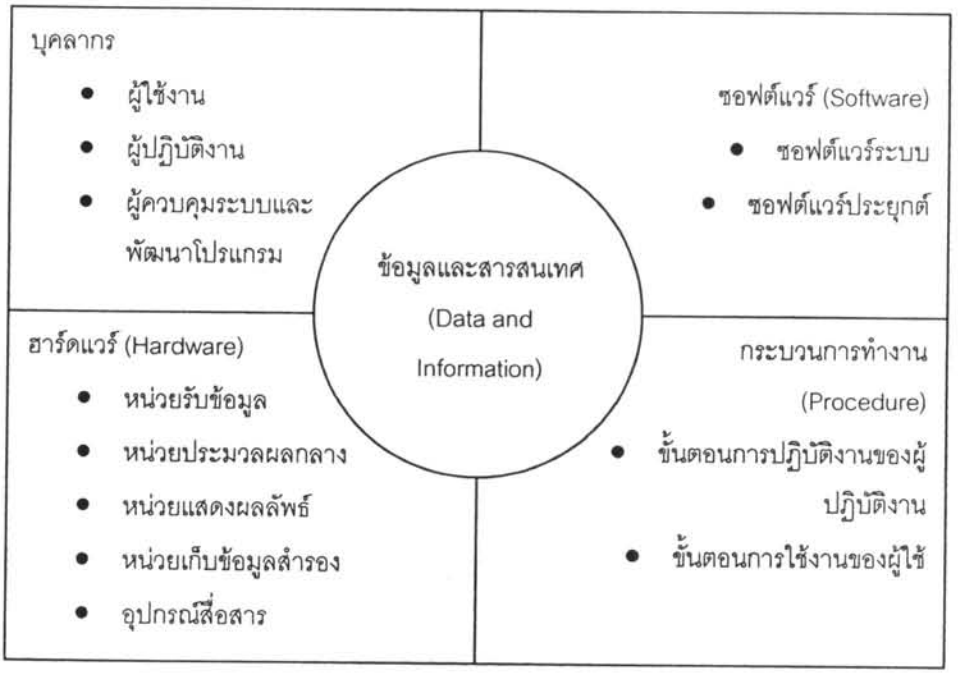
การเพิ่มคุณประโยชน์ในเชิงการแข่งขัน (Competitive Advantage) เป็นการสร้างความได้เปรียบในการแข่งขันเมื่อเทียบกับคู่แข่ง ไม่ว่าจะเป็นในเรื่องของการตอบสนองความต้องการของลูกค้า การผลิตสินค้าใหม่ ๆ เข้าสู่ตลาด การสร้างโอกาสทางธุรกิจ เป็นต้น ประโยชน์ในข้อนี้ ถือได้ว่าเป็นปัจจัยที่สำคัญอย่างยิ่งสำหรับองค์กรต่างๆ ในปัจจุบันลักษณะที่เห็นได้ คือ เป็นการทำในสิ่งที่ดีและสิ่งที่ใหม่ (Do better things and do the new things)

2.3.4 องค์ประกอบของระบบสารสนเทศ

ระบบสารสนเทศประกอบด้วยองค์ประกอบที่สำคัญ 5 ส่วน คือ

1. บุคลากร (Personnel)
2. ฮาร์ดแวร์ (Hardware)
3. ซอฟต์แวร์ (Software)
4. กระบวนการทำงาน หรือขั้นตอนการปฏิบัติงาน (Procedure)
5. ข้อมูลและสารสนเทศ (Data and Information)

โดยสามารถแสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 องค์ประกอบของระบบสารสนเทศ

2.3.5 โครงสร้างระบบสารสนเทศ

การอธิบายถึงโครงสร้างระบบสารสนเทศสามารถพิจารณาได้จาก 2 แนวทาง คือ โครงสร้างระบบสารสนเทศแบ่งตามระดับการบริหารและโครงสร้างระบบสารสนเทศแบ่งตามแหล่งที่มาของข้อมูล

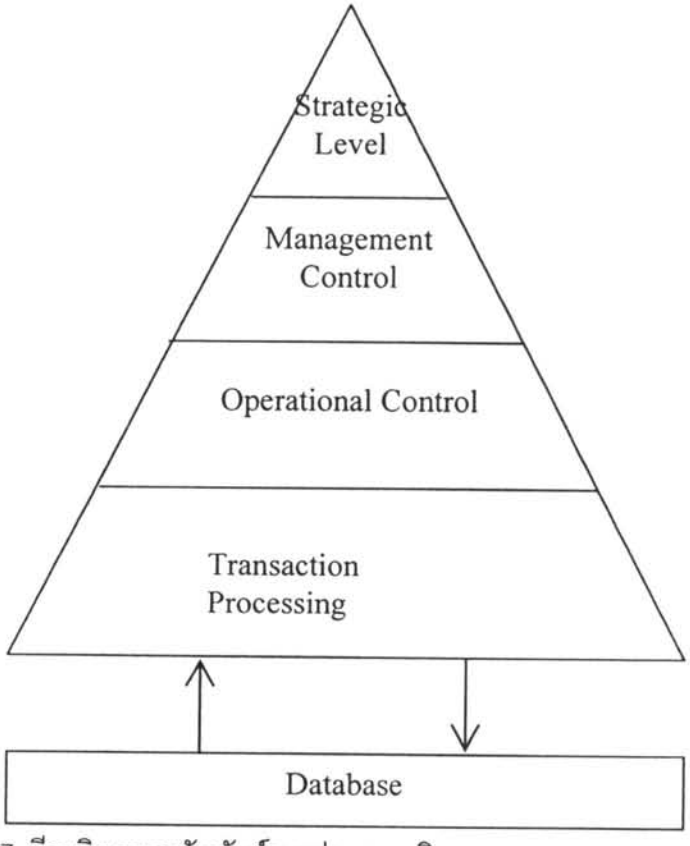
โครงสร้างระบบสารสนเทศแบ่งตามระดับการบริหาร โดยปกติการบริหารจัดการในหน่วยงานต่าง ๆ มักจะแบ่งเป็น 3 ระดับ คือ

- การบริหารระดับสูง ซึ่งเรียกกันว่า ระดับกลยุทธ์ (Strategic Level) เป็นระดับที่การจัดการเน้นไปด้านการวางแผนระยะยาว การกำหนดวัตถุประสงค์และเป้าหมายไกลออกไปข้างหน้าขนาด 3-5 ปี หรือมากกว่านั้น
- การบริหารระดับกลาง ซึ่งเรียกกันว่า ระดับกลวิธี (Tactical Level)เป็นระดับที่เน้นการจัดการให้สอดคล้องกับวัตถุประสงค์และเป้าหมายระยะยาวโดยจัดทำแผนดำเนินการในช่วงสั้น ๆ ระยะเวลาประมาณ 1 ปี
- การบริหารระดับล่าง ซึ่งเรียกกันว่า ระดับปฏิบัติการ (Operational Level) เป็นระดับที่เน้นการดำเนินงาน หรือ ปฏิบัติงานให้เป็นไปตามแผนงานระยะสั้นที่ได้กำหนดไว้

ทั้งนี้โครงสร้างการบริหารทั้งสามระดับมักจะเขียนเป็นรูปพีระมิด ดังรูปที่ 2.6

รูปที่ 2.6 พีระมิดของโครงสร้างการบริหาร 3 ระดับ

โครงสร้างการบริหารทั้งสามระดับดังกล่าวเมื่อนำมาสัมพันธ์กับระบบสารสนเทศ จะเกิดเป็นโครงสร้างระบบสารสนเทศ ดังรูปที่ 2.7



รูปที่ 2.7 พีระมิดความสัมพันธ์ระหว่างการบริหารและระบบสารสนเทศ

โครงสร้างระบบสารสนเทศซึ่งแบ่งตามระดับการบริหาร จะมีลักษณะเป็นรูปพีระมิด โดยฐานที่กว้างและสอบขึ้นไปบรรจบกันเป็นมุมแหลมตอนบน นั่นหมายถึง ขอบเขตกว้างขวางของข้อมูลที่มีมากในระดับล่าง และลดหลั่นน้อยลงไปเมื่อถึงยอดพีระมิดนี้ แบ่งออกได้เป็น 4 ระดับ คือ

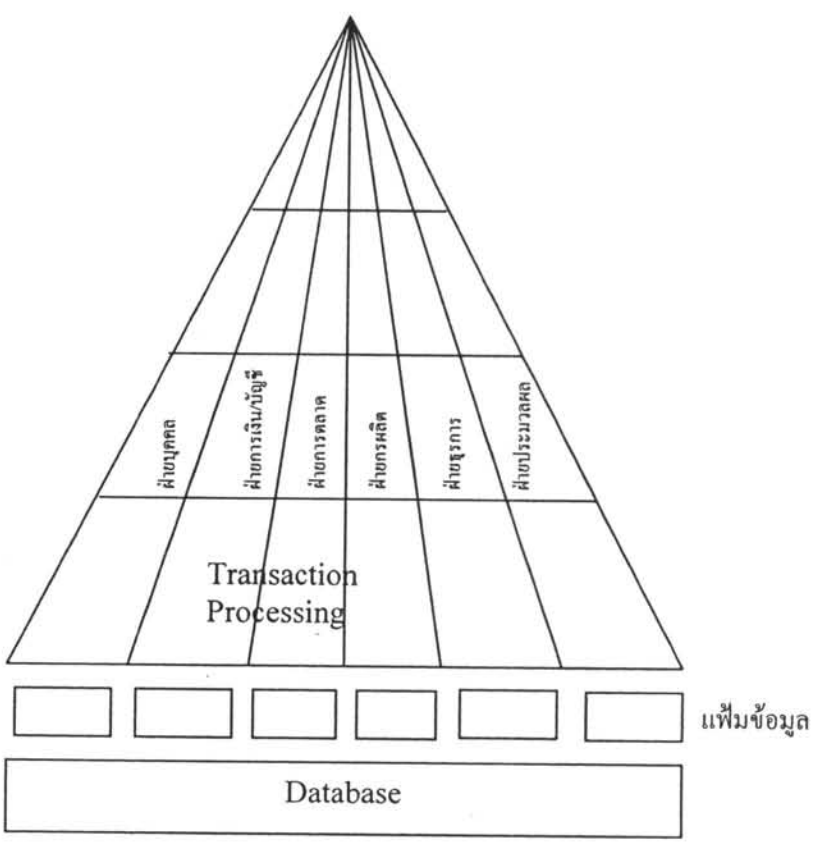
- ระดับล่างสุด หมายถึง การใช้คอมพิวเตอร์ทำงานประมวลผลข้อมูล ในแบบที่เรียกว่า Transaction Processing
- ระดับที่ 2 หมายถึง การใช้คอมพิวเตอร์จัดทำสารสนเทศ เพื่อใช้ในการวางแผน การควบคุม และการตัดสินใจที่เกี่ยวเนื่องกับงานประจำวัน ซึ่งเรียกว่าเป็นงาน Operational Control
- ระดับที่ 3 หมายถึง การใช้คอมพิวเตอร์จัดทำสารสนเทศสำหรับผู้บริหารจัดการระดับกลางใช้ในงานจัดการและวางแผนระยะสั้น ซึ่งเรียกว่าเป็นงาน Management

Control ซึ่งสารสนเทศระดับนี้ยังใช้สำหรับควบคุมและตัดสินใจเกี่ยวกับงานต่าง ๆ ว่าจะสามารถดำเนินการไปตามแผนระยะสั้นนั้นได้ด้วย

- ระดับที่ 4 หรือระดับยอด หมายถึง การใช้คอมพิวเตอร์จัดทำสารสนเทศสำหรับผู้บริหารจัดการระดับสูง สำหรับใช้ในงานวางแผนระยะยาวที่เรียกว่า Strategic Planning

จากรูป 2.7 ข้อที่ควรสังเกต คือ มีการใช้เทคโนโลยีฐานข้อมูลเป็นรากฐานในการบันทึกข้อมูลเอาไว้เป็นแหล่งกลางสำหรับให้งานประยุกต์ของทุกหน่วยงานใช้ร่วมกัน

นอกจากนี้โดยปกติแล้วองค์กรหนึ่งๆมักจะแบ่งการปฏิบัติงานออกเป็นฟังก์ชันหรือ ฝ่ายต่าง ๆ หลายฝ่าย เช่น แบ่งเป็นฝ่ายบัญชี ฝ่ายบริหาร ฝ่าย โรงงาน ฝ่ายบุคคล ฝ่ายการขาย เป็นต้น ในแต่ละฝ่ายนี้ก็มีการบริการทั้งสามระดับเหมือนกัน ดังนั้นจึงสามารถขยายรูปที่ 2.7 อีกให้เห็นรายละเอียดมากยิ่งขึ้นดังรูปที่ 2.8

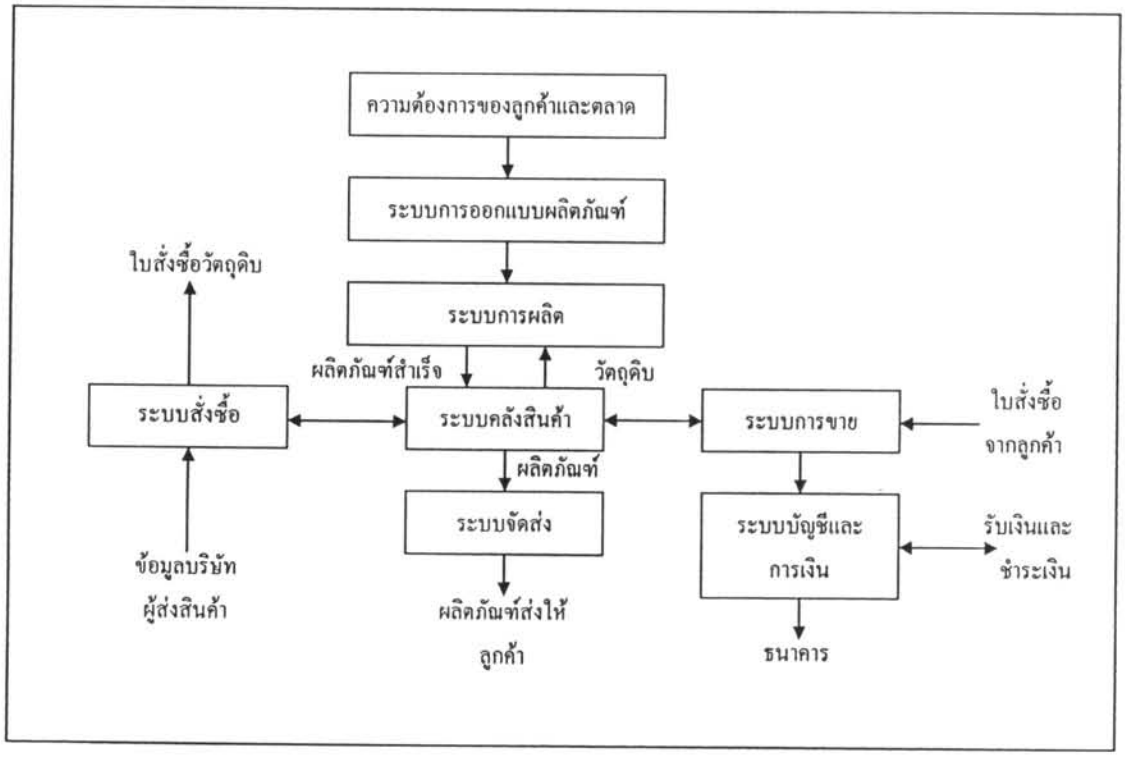


รูปที่ 2.8 ความสัมพันธ์ระหว่างการบริหารและแฟ้มข้อมูลเฉพาะ

จากรูปที่ 2.8 โครงสร้างใหม่นี้ได้แสดงแฟ้มข้อมูลเฉพาะของแต่ละฝ่ายเพิ่มเติมจากฐานข้อมูลที่มีอยู่เดิม ซึ่งหมายความว่า โดยปกติแม้มีการกำหนดโครงสร้างระบบสารสนเทศให้ใช้ฐานข้อมูลร่วมกัน เพื่อแบ่งกันใช้ข้อมูลโดยไม่ต้องจัดเก็บซ้ำซ้อน แต่ในทางปฏิบัติแต่ละฝ่ายอาจมีข้อมูลพิเศษที่ใช้เฉพาะของตัวเอง โดยไม่ต้องแบ่งกับฝ่ายอื่นๆก็ได้ ดังนั้นจึงควรจัดทำขึ้นเป็นแฟ้มข้อมูลสำหรับใช้เฉพาะในฝ่ายนั้น ๆ เท่านั้น

โครงสร้างระบบสารสนเทศแบ่งตามแหล่งที่มาของข้อมูลข้อมูลที่นำมาประมวลเป็นสารสนเทศในระบบสารสนเทศนั้นมีอยู่ 3 แบบ คือ

- ข้อมูลธุรกิจที่เกิดจากการดำเนินงานธุรกิจตามปกติ (Transaction) เป็นข้อมูลการสั่งซื้อสินค้า การรับใบสั่งสินค้า เป็นต้น
- ข้อมูลการดำเนินงาน เช่น ข้อมูลที่บอกว่า การดำเนินการได้ผล อย่างไร อาทิ ผลิตสินค้าได้วันละกี่ชิ้น การตรวจสอบคุณภาพและพบสินค้าที่ไม่ได้ มาตรฐานจำนวนเท่าใด การจัดทำเอกสารรายงานต่าง ๆ ล่าช้าหรือรวดเร็วประการใด
- ข้อมูลภายนอก ได้แก่ ข้อมูลภาวะตลาด เศรษฐกิจ สังคม และสิ่งแวดล้อม ที่จะมีผลต่อการดำเนินการของหน่วยงาน โครงสร้างแบบนี้จะมีลักษณะดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างระบบสารสนเทศเมื่อแบ่งตามแหล่งที่มาของข้อมูล

2.3.6 การพัฒนาระบบสารสนเทศ

องค์กรใดๆก็ตามโดยทั่วไปจะมีระบบสารสนเทศที่ใช้งานอยู่และได้รับการนำไปใช้งานโดยผู้บริการแต่เมื่อดำเนินเการไประยะหนึ่งอาจจำเป็นต้องมีการปรับปรุงและพัฒนาระบบสารสนเทศ

เหตุที่มาของการพัฒนาระบบสารสนเทศ มักจะเกิดขึ้นจากสาเหตุดังนี้

1. เพื่อแก้ไขปัญหาที่เกิดขึ้น ไม่ว่าจะเนื่องด้วย การวางระบบเดิมไม่เหมาะสม หรือสภาพการณ์เปลี่ยนแปลงไปเช่นองค์กรขยายใหญ่ขึ้น 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

- การพัฒนาระบบงานตามวงจรการพัฒนาระบบ (System Development Life Cycle)

- การพัฒนาระบบงานโดยการสร้างระบบต้นแบบ (Prototyping)
- การพัฒนาระบบงานโดยการนำชุดซอฟต์แวร์สำเร็จรูปมาใช้ (Application Software Package)
- การพัฒนาระบบงานโดยผู้ใช้งานปลายทาง (End-User Development)
- การพัฒนาระบบงานโดยการจ้างหน่วยงานภายนอก (Outsourcing)

2.3.7 ระบบสารสนเทศเพื่อการบริหาร

ระบบสารสนเทศเพื่อการบริหาร (Management Information System) หรือ MIS คือ ระบบที่มีการจัดอย่างเป็นระเบียบ และรวมเข้าเป็นกลุ่มโครงสร้างที่ ประกอบขึ้นมาจากบุคคลจำนวนมาก เครื่องมือ และระเบียบวิธีการต่าง ๆ ที่ช่วยให้มีข้อมูลที่ถูกต้องทั้งจากแหล่งภายในและภายนอก กล่าวคือ ระบบสารสนเทศเพื่อการบริหารเป็นระบบที่รวม (Integrate) ผู้ใช้และเครื่อง (User-Machine) เข้า ไว้ด้วยกัน โดยทำหน้าที่ในการจัดหาสารสนเทศ หรือข่าวสารเพื่อช่วยในการ ตัดสินใจของผู้บริหาร ในเรื่องของกระบวนการจัดการองค์กร เช่น การวางแผนการจัดองค์กร และการควบคุม เพื่อให้องค์กรสามารถดำเนินการไปตามวัตถุประสงค์ที่ตั้งไว้

ระบบสารสนเทศเพื่อการบริหารจะต้องมีการประสานร่วมกับหน่วยงานหรือระบบย่อยอื่น ๆ ในองค์กร โดยมีลักษณะการจัดตั้งที่เป็นระบบ และง่ายแก่การประสานงานกับระบบย่อยอื่น ๆ ในองค์กรด้วย

ระบบสารสนเทศเพื่อการบริหารสามารถดำเนินการได้โดยไม่ต้องอาศัยคอมพิวเตอร์เข้ามาช่วย แต่เนื่องจากความสามารถของคอมพิวเตอร์ ในอันที่จะประมวลผลข้อมูลได้จำนวนมากในเวลาอันรวดเร็ว ดังนั้นในปัจจุบันระบบสารสนเทศเพื่อการบริหารจึงมักจะผ่านกระบวนการประมวลผลด้วยคอมพิวเตอร์

หน้าที่หลักของสารสนเทศเพื่อการบริหาร ประกอบด้วย

- ให้สารสนเทศเพื่อช่วยในการตัดสินใจของผู้บริหารได้
- ให้สารสนเทศแก่ผู้บริหารทุกระดับได้
- ให้สารสนเทศเพื่อช่วยในการแก้ไขปัญหาทุกรูปแบบของปัญหา
- ให้สารสนเทศที่รวดเร็วและเหมาะสมกับการใช้งาน

ประโยชน์ที่ผู้บริหารจะได้รับจากระบบสารสนเทศเพื่อการบริหาร สามารถสรุปได้ดังนี้

- ช่วยให้ผู้บริหารมองเห็นปัญหาและโอกาสได้รวดเร็วขึ้น
- ช่วยให้ผู้บริหารมีเวลาสำหรับการวางแผนได้มากขึ้น
- ช่วยให้ผู้บริหารใช้เวลาในการพิจารณาปัญหาที่มีความซับซ้อนได้มากขึ้น
- ช่วยให้ผู้บริหารควบคุมการดำเนินการได้ดีขึ้น
- คุณลักษณะที่สำคัญของระบบสารสนเทศเพื่อการบริหารประกอบด้วย
- เป็นระบบที่เกี่ยวข้องกับการจัดการ สิ่งนี้ถือได้ว่าเป็นคุณลักษณะที่สำคัญของระบบสารสนเทศเพื่อการบริหาร กล่าวคือ ต้องสามารถตอบสนองความต้องการของผู้บริหาร และต้องเป็นสารสนเทศที่ใช้เพื่อการบริหาร คือ สามารถใช้ประกอบในการวางแผน การควบคุมงานได้
- ผู้บริการต้องเป็นแกนนำในการพัฒนาระบบ เนื่องจากระบบสารสนเทศเพื่อการบริหารนี้เกี่ยวข้องและถูกใช้งานโดยตรงจากผู้บริหาร ดังนั้น ผู้บริหารต้องเป็นผู้ที่มีส่วนร่วมในการออกแบบและกำหนดสารสนเทศที่ต้องการ
- มองปัญหาในลักษณะเป็นอันหนึ่งอันเดียวกัน ระบบสารสนเทศเพื่อการบริหารจะต้องประสานระบบย่อย ๆ ในองค์กรให้เป็นหนึ่งเดียวกัน ไม่ว่าจะเป็นฝ่ายการตลาด ฝ่ายผลิต ฝ่ายวิศวกรรมและอื่น ๆ
- การใช้ฐานข้อมูลร่วมกัน ถือได้ว่าเป็นหัวใจสำคัญของการจัดการอย่างมีประสิทธิภาพ ทำให้ลดความซ้ำซ้อนของข้อมูล ทำให้ระบบทำงานเร็วขึ้น และประหยัดค่าใช้จ่าย
- ต้องการการวางแผนที่ดี เนื่องจากการที่ไม่สามารถสร้างขึ้นได้ด้วยระยะเวลาอันสั้น ดังนั้น จึงต้องมีการวางแผนอย่างดี และคำนึงถึงปัญหาต่าง ๆอย่างรอบคอบในการพัฒนาแลใช้งานระบบ
- อาศัยแนวความคิดเชิงระบบในการพัฒนาระบบ
- เป็นระบบที่โดยทั่วไปอาศัยคอมพิวเตอร์

ระบบสารสนเทศเพื่อการบริหารนั้น แม้จะสร้างขึ้นให้กับผู้บริหารใช้ก็จริงอยู่ แต่ผลลัพธ์ของระบบ หรือรายงานที่จะจัดทำให้ผู้บริหารแต่ละระดับนั้นมีความแตกต่างกัน เพราะขึ้นอยู่กับหน้าที่ของผู้บริหารแต่ละคนซึ่งจะบังคับให้ต้องการสารสนเทศที่ต่างกัน ดังได้เคยกล่าวไปแล้วว่าผู้บริหารระดับบนสุดต้องการสารสนเทศสำหรับการวางแผนกลยุทธ์ ซึ่งเป็นแผนสำหรับการทำให้บริษัทแข่งขันกับบริษัทอื่น ๆ ได้ ดังนั้น สารสนเทศที่ต้องใช้จึงมักจะเป็นสารสนเทศที่เกี่ยวกับสภาพของตลาดและสถานการณ์ภายนอกบริษัทมากกว่าจะเป็นสารสนเทศจากภายในบริษัท ในทางตรงกันข้าม ผู้บริหารระดับล่างซึ่งต้องควบคุมการปฏิบัติงานภายในให้ดำเนินไปตามเป้าหมายและวัตถุประสงค์ที่วางไว้ก็ต้องการสารสนเทศจากภายในมากกว่าภายนอก ดังแสดงได้

ตารางที่ 2.1 ตารางแสดงความสัมพันธ์ของระดับการบริหาร และคุณลักษณะสารสนเทศที่ต้องการ

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

การออกแบบระบบสารสนเทศเพื่อการบริหาร เป็นการจัดวางระบบ สารสนเทศเพื่อการบริหารใหม่ทั้งหมด หรือเป็นการปรับปรุงระบบสารสนเทศเพื่อการบริหารเดิมเพียงบางส่วน โดยการออกแบบนี้จะขึ้นกับผลที่ได้จากการศึกษาและวิเคราะห์ระบบสารสนเทศเพื่อการบริหารเดิม และผลการตัดสินใจของผู้บริหารว่าต้องการระบบสารสนเทศเพื่อการบริหารใหม่เป็นอย่างไร ทั้งนี้กระบวนการดังกล่าว จะประกอบด้วยขั้นตอนที่สำคัญดังนี้

- การออกแบบรายงาน
- การออกแบบข้อมูลเพื่อนำเข้าระบบประมวลผล
- การออกแบบระบบประมวลผล

การออกแบบรายงาน รายงานเป็นส่วนที่สำคัญสำหรับผู้บริหารที่จะไปใช้ประโยชน์ ดังนั้น ถ้ารายงานเป็นไปตามความต้องการของผู้บริหารแล้ว ก็ถือได้ว่าระบบที่ออกแบบ

บรรลุเป้าหมายไปได้ส่วนหนึ่ง สำหรับขั้นตอนโดยละเอียดของการออกแบบรายงานจะประกอบด้วย

- การกำหนดรายงานที่ต้องการ เป็นการกำหนดถึงรายงานที่ต้องการจากระบบ โดยนำผลจากขั้นตอนการศึกษาและวิเคราะห์ระบบมาทบทวนและพิจารณาร่วมกับความต้องการของผู้บริหารและผู้ปฏิบัติงาน หลักที่ใช้ในการพิจารณารายงานที่ต้องการจากระบบได้แก่ 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 ซึ่งงานในขั้นตอนนี้จะ

นำเอาผลการวิเคราะห์รายงานที่ได้ออกแบบไว้ มาพิจารณาถึงชนิดและขนาดของข้อมูลที่ใช้เป็นข้อมูลนำเข้า

- แหล่งข้อมูลนำเข้า ในการวิเคราะห์จำเป็นต้องหาแหล่งข้อมูลที่จำเป็นต้องใช้เพื่อกำหนดแหล่งข้อมูลนำเข้าของระบบ ทั้งนี้แหล่งข้อมูลที่ใช้ดังกล่าวเพื่อจัดทำรายงานอาจแบ่งออกได้เป็น
  - แหล่งข้อมูลจากเอกสารชิ้นเดียวกัน การใช้แหล่งข้อมูลนี้จะไม่ยุ่งยากมาก เนื่องจากข้อมูลทั้งหมดมาจากเอกสารในชิ้นเดียวกัน
  - แหล่งข้อมูลที่เกิดจากการคำนวณ บางรายงานอาจมีข้อมูลที่มาจากแหล่งเดียว และข้อมูลบางส่วนได้มาจากการนำข้อมูลไปคำนวณ
  - แหล่งข้อมูลหลายแหล่ง ลักษณะแหล่งข้อมูลแบบนี้จะทำให้เกิดความยุ่งยากในการออกแบบระบบสารสนเทศ เนื่องจากข้อมูลที่นำเข้าจะมีหลายแบบ
  - แหล่งข้อมูลจากตารางที่ได้กำหนดขึ้น เป็นการกำหนดค่าไว้เป็นตารางอ้างอิง และนำมาประมวลผล ซึ่งเป็นวิธีการที่นิยมใช้กันโดยทั่วไป เนื่องจากเป็นการสรุปข้อมูลในรูปแบบที่สามารถนำเสนอได้ง่าย และการเตรียมข้อมูลนำเข้าก็สะดวก
- การกำหนดระยะเวลาของข้อมูลนำเข้า เป็นการกำหนดระยะเวลาและความถี่ของข้อมูลนำเข้า ทั้งนี้เพื่อให้ทันต่อความต้องการใช้ในการประมวลผลให้ได้เป็นรายงานตามที่ต้องการ

การออกแบบระบบประมวลผลจะครอบคลุมตั้งแต่ การเก็บรวบรวมข้อมูล การจดบันทึก การเก็บรักษา การคำนวณ การประมวลผล การวิเคราะห์และการเรียกกลับมาใช้ในภายหลัง ทั้งนี้เพื่อที่จะประมวลผลข้อมูลให้ได้เป็นสารสนเทศและรายงานตามที่ต้องการ

2.4 การออกแบบระบบเชิงวัตถุ (Object Orientation) (กิตติ ภักดีวัฒนะกุล, พนิดา พานิชกุล : 2548)

2.4.1 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 ได้ง่าย เช่น การสร้าง การเปลี่ยนแปลง การเล่น การกระตุ้น เป็นต้น ซึ่ง Interaction นี้เองที่ทำให้เกิด กิจกรรม (Activity) ต่างๆ ในโลกนี้

2.4.2 อ็อบเจ็กต์ (Object) และคลาส (Class)

2.4.2.1 อ็อบเจ็กต์ (Object)

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

เราไม่สามารถนำเอาวัตถุในโลกความเป็นจริง เข้ามาใส่ในเครื่องคอมพิวเตอร์ได้ ดังนั้นสิ่งที่ต้องทำก็คือ การใส่แนวคิด (Concept) ให้แก่วัตถุแล้วจึงสร้างแบบจำลองของวัตถุในโลกความเป็นจริงนั้นๆ เพื่อนำไปใส่ไว้ในคอมพิวเตอร์

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

2.4.2.2 <u>คลาส (Class) หรือ Abstract Object</u>

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

2.4.3 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 (เพราะระบบจำลมบูรณ์ได้ก็ต่อเมื่อมีกิจกรรมเกิดขึ้นภายในส่วนประกอบต่างๆ 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

2.4.4 องค์ประกอบของอ็อบเจ็กต์

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

2.4.4.1 <u>คุณสมบัติ (Attribute)</u>

เราสามารถบรรยายคุณสมบัติของอ็อบเจ็กต์ต่างๆ ตราบเท่าที่คุณสมบัติดังกล่าวเป็นคุณสมบัติที่เราสนใจหรืออยู่ใน Domain ที่สนใจ เช่น สนและจำนวนของประตูรถคันหนึ่ง สีผิว

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

2.4.4.2 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

2.4.4.3 การบ่งชี้อ็อบเจ็กต์ (Unique Identity / Object Identity)

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

2.4.5 ประเภทของ Abstraction

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

2.4.5.1 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 และให้แนวคิด แยกแยะและจัดหมวดหมู่ จนกลายเป็นคลาสขึ้นมาได้ในที่สุด

2.4.5.2 Association Abstraction

Association หมายถึงความสัมพันธ์ระหว่างคลาส (หรืออ็อบเจ็กต์) ที่อยู่ในระดับเดียวกัน กล่าวคือ คลาสทั้งสองมีความสำคัญเท่าเทียมกัน ไม่มีคลาสใดเป็นองค์ประกอบของคลาสใด 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

2.4.5.3 Aggregation Abstraction

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

2.4.5.4 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 ไม่มีความ

คล้ายคลึงกันทางด้านโครงสร้าง แต่ยังมีความจริงบนโลกอีกประการหนึ่งของความสัมพันธ์แบบต่างระดับที่ว่า "วัตถุหลักและวัตถุย่อยที่ถึงแม้ว่าจะอยู่ต่างระดับกันแต่มีความคล้ายคลึงกันทางด้านโครงสร้างและความหมาย (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

2.5 UML (กิตติ ภักดีวัฒนะกุล, พนิดา พานิชกุล : 2548)

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 จึงได้นำข้อดีในการสร้าง

แบบจำลองตาม 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

2.5.1 คำศัพท์ในภาษา UML

คำศัพท์ในภาษา UML จะแสดงตามกลุ่มองค์ประกอบทั้ง 3 ได้แก่ Things, Relationships และ Diagrams โดยคำศัพท์ในแต่ละกลุ่มถูกแสดงให้อยู่ในรูปของสัญลักษณ์ต่างๆ ที่จะนำมาประกอบกันเป็นแผนภาพ

I22917593X

2.5.1.1 Things

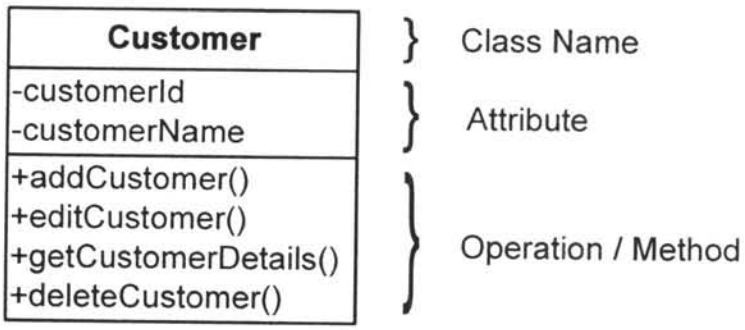
Things คือ สิ่งที่ได้จากการ Abstraction ซึ่งสามารถแบ่งออกเป็นกลุ่มตามชนิดของคำได้ ดังนี้

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) ดังรูปที่ 2.10



รูปที่ 2.10 สัญลักษณ์ "Class"

○ ยูสเคส (Use Case)

สิ่งที่ใช้อธิบายถึงกิจกรรมของระบบที่เกิดขึ้นตามลำดับขั้นตอน อันจะส่งผลตอบสนองต่อผู้กระทำต่อระบบ (Actor) สัญลักษณ์ของ Use Case จะใช้รูปวงรี และเขียนชื่อ Use Case ไว้ในวงรี ดังรูปที่ 2.11

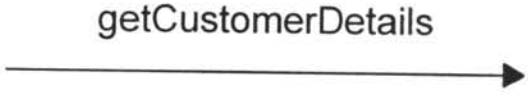
รูปที่ 2.11 สัญลักษณ์ "Use Case"

2.5.1.2 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 ดังรูปที่ 2.12



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 ดังรูปที่ 2.13

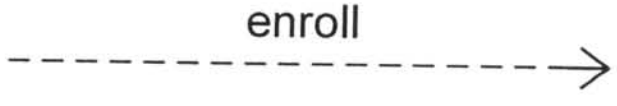
รูปที่ 2.13 สัญลักษณ์ "State Machine"

2.5.2 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

2.5.2.1 Dependency

อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่งแบบส่งผลกระทบต่อกัน โดยหากมีการเปลี่ยนแปลงในสิ่งหนึ่งจะส่งผลกระทบต่อสิ่งหนึ่ง เช่น การเปลี่ยนแปลงโครงสร้างของคลาสย่อมส่งผลกระทบต่ออ็อบเจ็กต์ของคลาสนั้น สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบ Dependency คือ เส้นประหัวลูกศรแบบก้างปลา โดยที่อาจมีการเขียนลักษณะความสัมพันธ์ไว้บนเส้นก็ได้ ดังรูปที่ 2.14

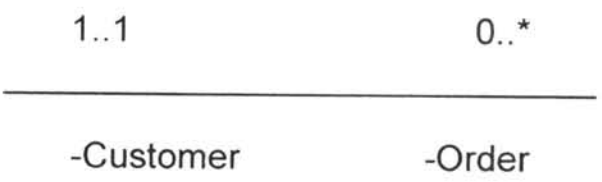


รูปที่ 2.14 สัญลักษณ์ "Dependency"

2.5.2.2 Association

อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่งที่มีระนาบความสัมพันธ์เดียวกัน (มีความสำคัญเทียบเท่ากัน ไม่มีสิ่งใดสำคัญกว่าสิ่งใด) เช่น ความสัมพันธ์ระหว่างลูกค้ากับใบสั่งซื้อ เป็นต้น สัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบ Association คือ เส้นตรง และมีข้อความแสดงบทบาทความสัมพันธ์ไว้บนเส้นตรง หรืออาจเพิ่ม Multiplicity ก็ได้ (Multiplicity หมายถึง ค่าของ

จำนวนสมาชิกในคลาสที่มีส่วนร่วมในความสัมพันธ์ มีรูปแบบคือ Minimum...Maximum เช่น 0..* หรือ 1..1 เป็นต้น) ดังรูปที่ 2.15



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 มีลักษณะดังรูปที่ 2.16 และ รูปที่.2.17

รูปที่ 2.16 สัญลักษณ์ "Composition"

รูปที่ 2.17 สัญลักษณ์ "Aggregation"

2.5.2.3 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 ดังรูปที่ 2.18

รูปที่ 2.18 สัญลักษณ์ "Generalization Relationship"

2.5.2.4 Realization

อธิบายถึงความสัมพันธ์ระหว่างสิ่ง 2 สิ่ง โดยที่สิ่งหนึ่งจะทำหน้าที่ในการดำเนินการตาม Method ที่อีกสิ่งหนึ่งได้ประกาศไว้ ใน UML มี Realization 2 ประเภทคือ Realization ระหว่างคลาสกับอินเตอร์เฟส และ Realization ระหว่างคอมโพเนนท์กับอินเตอร์เฟส สัญลักษณ์ที่ใช้แทน Realization คือ เส้นประลูกศรโปร่ง ดังรูปที่ 2.19

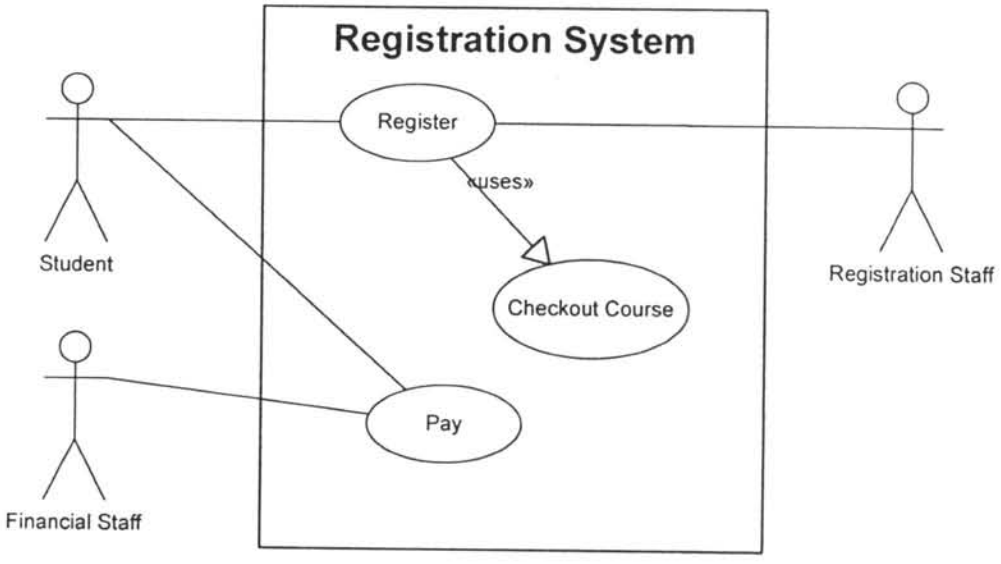
รูปที่ 2.19 สัญลักษณ์ "Realization Relationship".

2.5.3 Diagram

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

2.5.3.1 Use Case Diagram

เป็นแผนภาพที่ใช้แสดงถึงขั้นตอนการทำงานที่สำคัญของระบบ (Use Case) อาจกล่าวได้ว่าเป็น หน้าที่หรืองานที่ระบบจะต้องปฏิบัติ เพื่อตอบสนองต่อผู้กระทำต่อระบบ (Actor) โดย Use Case Diagram จะแสดงความสัมพันธ์ระหว่าง Use Case และ Actor จัดว่าเป็นคลาสพิเศษ แสดงตัวอย่าง Use Case Diagram ดังรูปที่ 2.20



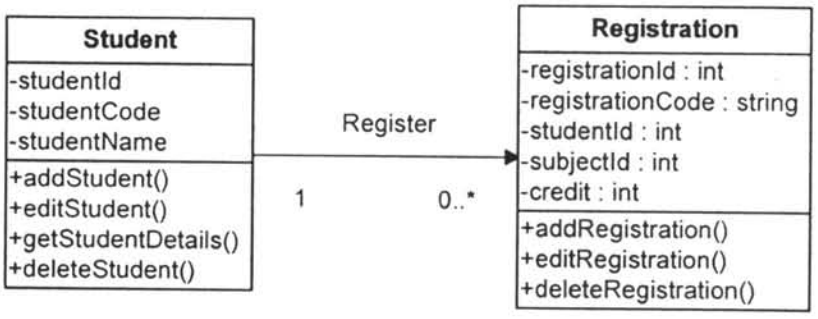
รูปที่ 2.20 ตัวอย่าง Use Case Diagram

Use Case Diagram จะประกอบไปด้วย

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

2.5.3.2 Class Diagram

เป็นแผนภาพที่ใช้ในการแสดงกลุ่มของคลาส โครงสร้างของคลาส อินเตอร์เฟส (Interface) และแสดงความสัมพันธ์ (Relationship) ระหว่างคลาส ซึ่งแผนภาพนี้เป็นแผนภาพที่จะพบมากที่สุดในทาง Object Orientation แสดงตัวอย่าง Class Diagram ดังรูปที่ 2.21

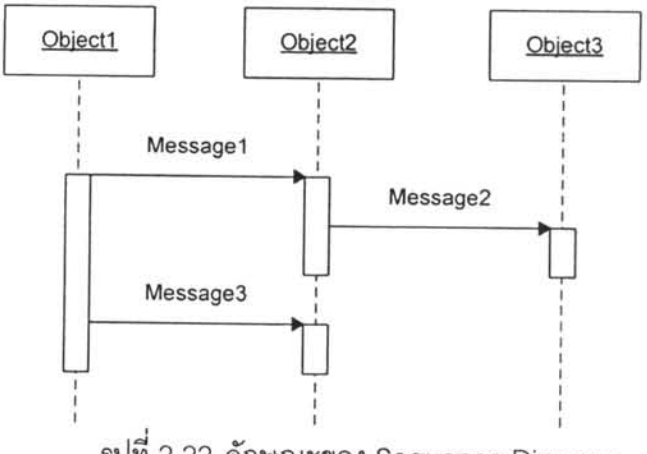


รูปที่ 2.21 ตัวอย่าง Class Diagram

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..*)

2.5.3.3 Sequence Diagram

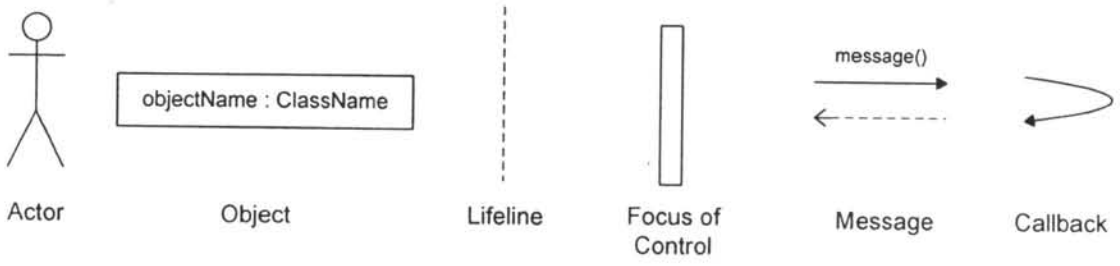
เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจ็กต์โดยเฉพาะการส่ง Message ระหว่างอ็อบเจ็กต์ตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้นจากน้อยไปมาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน แสดงลักษณะของ Sequence Diagram ดังรูปที่ 2.22



รูปที่ 2.22 ลักษณะของ Sequence Diagram

จากรูป Sequence Diagram ประกอบไปด้วยสัญลักษณ์ต่างๆ ดังนี้

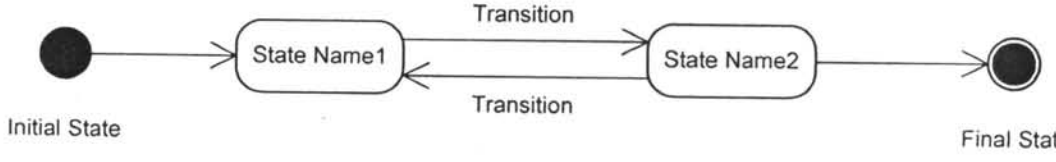
1. Actor คือ ผู้กระทำต่อระบบ
2. Object คือ อ็อบเจ็กต์ที่ต้องทำหน้าที่
3. Lifeline คือ เส้นแสดงชีวิตของอ็อบเจ็กต์หรือคลาส
4. Focus on Control / Activation คือ จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่
5. Message / Callback คือ คำสั่งหรือฟังก์ชันที่คลาสหนึ่งส่งให้อีกคลาสหนึ่ง ซึ่งสามารถส่งกลับได้ด้วย



รูปที่ 2.23 สัญลักษณ์ภายใน Sequence Diagram

2.5.3.4 State Chart Diagram

เป็นแผนภาพที่แสดงให้เห็นพฤติกรรมของอ็อบเจ็กต์เช่นเดียวกับแผนภาพในกลุ่ม Behavioral Diagram อื่นๆ แต่ Statechart Diagram จะเน้นที่การแสดงให้เห็นถึงสถานะ (State) การเปลี่ยนสถานะ (Transition) ที่มีต่อเหตุการณ์ (Event) ที่เกิดขึ้นในช่วงชีวิตของอ็อบเจ็กต์ 1 ช่วง (1 Sequence) แสดงลักษณะของ Statechart Diagram ดังรูปที่ 2.24



รูปที่ 2.24 ลักษณะของ Statechart Diagram

สัญลักษณ์ที่ปรากฏอยู่ใน Statechart Diagram มีดังนี้

1. Initial State คือ จุดเริ่มต้นการเปลี่ยนสถานะ
2. Final State คือ จุดสิ้นสุดของการเปลี่ยนสถานะ
3. Transition คือ เส้นกระตุ้นให้เปลี่ยนสถานะ

4. State คือ สถานะของอ็อบเจ็กต์

2.5.4 ข้อดีและข้อเสียของภาษา UML

ข้อดี

- UML สามารถสะท้อนภาพของระบบได้ใกล้เคียงกับโลกของความเป็นจริงมากที่สุด จึงทำให้เป็นเรื่องง่ายที่จะทำความเข้าใจ
- UML เป็นภาษาที่มีแบบแผนแน่นอนและเป็นหนึ่งเดียว ไม่ว่าใครก็ตามที่มีส่วนเกี่ยวข้องกับระบบก็สามารถอ่านและทำความเข้าใจแบบจำลองระบบที่สร้างด้วยภาษา UML ตัวเดียวกันได้ในทิศทางเดียวกัน และเข้าใจตรงกัน
- UML สามารถใช้เป็นเครื่องมือในการพัฒนาระบบได้ทั้งกระบวนการนับตั้งแต่การสรุปความต้องการการวิเคราะห์ความต้องการ การออกแบบระบบ และยังใช้เป็นเครื่องมือเพื่อชี้แนะแนวทางในการเขียนโปรแกรมได้อีกด้วย
- สามารถหาซอฟต์แวร์ที่สนับสนุนการสร้างแบบจำลองภาษา UML ตามท้องตลาดได้ง่าย

ข้อเสีย

- กรณีที่เป็นระบบงานขนาดใหญ่และจำเป็นต้องมีการระบุหมายเหตุ (Note) ไว้ด้วย จะทำให้แผนภาพดูรกและยุ่งเหยิงจนเกินไป
- Business Rule หรือเงื่อนไขทางธุรกิจไม่ได้ถูกจัดให้อยู่รวมกันเป็นกลุ่มในที่เดียวกัน แต่กลับกระจัดกระจายกันอยู่ตามแผนภาพชนิดต่างๆ ทำให้การตรวจสอบเงื่อนไขทางธุรกิจที่เกี่ยวข้องกันเป็นไปด้วยความยากลำบาก
- ไม่สามารถตรวจสอบความสอดคล้องกันของแผนภาพแต่ละชนิดได้ (Consistency Checking)

2.6 รูปแบบการทำบัตรผู้ป่วย

2.6.1 บัตรแบบรหัส Barcode(ดร.สุวิทชัย คุ้มปิติ : 2531)

รหัสแถบ (Bar code) คือ แถบเส้นดำยาวพิมพ์เรียงเป็นแถบบนตัวภาชนะสำหรับบรรจุสินค้าที่วางขายกันตามร้านค้าหรือซูเปอร์มาร์เก็ททั่วไป สิ่งซึ่งแถบดำเหล่านี้เหมายถึงนั้นมักจะเป็น "ข้อความ" ที่ใช้บ่งบอกตัวสินค้านั้น ๆ เช่นว่า ยาสีฟัน เป็นต้น

การใช้รหัสแถบบวกกับเครื่องอ่านรหัสแถบนี้ทำให้เกิดความสะดวกรวดเร็ว และความแม่นยำในการทำงานได้มาก ตัวอย่างในซูเปอร์มาร์เก็ท รหัสแถบที่ติดอยู่บนตัวสินค้า จะทำให้การคิดเงินทำได้อย่างรวดเร็วและแม่นยำ คือเมื่อพนักงานเพียงแต่ใช้ตัวอ่านรหัสแถบรูดผ่านรหัสแถบ ก็จะทราบว่าสินค้าชนิดนั้นเป็นสินค้าอะไร เมื่อบวกกับการโปรแกรมราคาสินค้าเข้ากับเครื่องคิดเงินบางประเภท ความผิดพลาดในการกดราคาสินค้าก็จะไม่เกิดขึ้น และในกรณีนี้ไม่มีความจำเป็นที่จะต้องติดราคาสินค้าลงบนสินค้าทุกตัว ทำให้สะดวกต่อการเปลี่ยนแปลงราคาสินค้าในอีกทางหนึ่ง

อีกตัวอย่างของการใช้รหัสแถบได้แก่ ศูนย์แยกจดหมายหรือสิ่งของพัสดุภัณฑ์ ในปัจจุบันการแยกแยะจดหมายอัตโนมัติโดยการให้เครื่องอ่านที่อยู่บนซองจดหมายหรือพัสดุภัณฑ์นั้นยังทำไม่ได้ถึง 100 เปอร์เซ็นต์ และอีกประการหนึ่งความรวดเร็วก็ยังไม่ได้ระดับที่น่าพอใจ ระบบแยกจดหมายจึงใช้รหัสแถบเป็นสื่อกลาง โดยก่อนที่จะมีการส่งเข้าระบบแยก เราจะตีรหัสแทนที่อยู่ปลายทางลงบนตัวจดหมายก่อน จากนั้นส่วนต่าง ๆ ในระบบแยกจดหมายก็จะอาศัยการอ่านรหัส ซึ่งสามารถทำได้อย่างง่ายดาย และแม่นยำในการแยกแยะจดหมายต่อไป

ในปัจจุบัน รหัสแถบนี้มีบทบาทอย่างมากในการประยุกต์ใช้ เพื่อการบ่งบอกวัตถุอย่างอัตโนมัติ (Automatic Identification) สืบเนื่องจากเทคนิคและอุปกรณ์สำหรับการ recognize รหัสแถบนี้อยู่ในขั้นปฏิบัติการได้อย่างแน่นอนแล้ว ซึ่งผิดกับการ Identification ด้วยภาพหรือเสียงที่ยังต้องค้นคว้าปรับปรุงกันอีกมาก

2.6.1.1 <u>หลักการอ่านรหัสแถบ</u>

สำหรับการอ่านรหัสแถบ เขาใช้หลักการที่ว่า พื้นสว่างจะสะท้อนได้มากกว่าพื้นมืด ดังนั้นเมื่อตัวอ่านถูกกวาดไปบนรหัสแถบ ลำแสงที่ถูกปล่อยออกมาจากหัวอ่านจะสะท้อนกลับมาหรือน้อยก็ขึ้นอยู่กับว่า มันได้ตกกระทบแถบขาวหรือแถบดำ แสงสะท้อนกลับเหล่านี้จะถูก

ดัดแปลงเป็นสัญญาณไฟฟ้า โดย Photodiode ที่ติดอยู่ที่หัวอ่าน องค์ประกอบสำคัญของตัวอ่านรหัสแถบก็คือ ขนาดของลำแสงที่ส่งออกมานั้น จะต้องสัมพันธ์กับความละเอียด (resolution) ของแถบ กล่าวคือ ขนาดของมันจะต้องไม่ใหญ่กว่าความกว้างของแถบดำหรือแถบขาวที่แคบที่สุด ในทางปฏิบัติเขาใช้จุดลำแสงที่มีขนาดเส้นผ่าศูนย์กลางประมาณ 0.2 มม.

ส่วนสำคัญอีกส่วนหนึ่งก็คือความยาวคลื่นของแสงที่ใช้ ซึ่งขึ้นกับว่าจะใช้อ่านรหัสแถบสีอะไร โดยทั่วไปเขาใช้แสงอินฟราเรด (Infrared) ที่มีความยาวคลื่นประมาณ 0.95 ไมครอน (micron) สำหรับอ่านแถบขาวดำ และใช้แสงสีแดงที่มีความยาวคลื่น 0.65 ถึง 0.7 ไมครอน สำหรับอ่านรหัสแถบสีเขียวหรือสีน้ำเงินที่พิมพ์บนพื้นสีเหลืองหรือส้ม

2.6.1.2 <u>ลักษณะของรหัส</u>

ในการอธิบายลักษณะของรหัสนั้น เขาจะใช้พารามิเตอร์อยู่สองสามตัว กล่าวคือ สิ่งแรก ดูว่ารหัสแถบนั้นเป็นชนิด NRZ (Not Return to Zero) หรือว่าชนิดโมดูเลชัน (Modulation) ด้วยความกว้าง ในกรณีที่เป็น NRZ การรักษาระดับลอจิค (logic) ไม่จำเป็นต้องเปลี่ยนระดับสัญญาณ กล่าวคือ ถ้าแถบขาวแทนเลข 0 เราสามารถจะแทนเลข 0 หลายตัวที่อยู่ติดกันได้ด้วยแถบขาวยาว โดยไม่ต้องมีแถบดำสลับกันไป แต่ในกรณีที่รหัสเป็นแบบโมดูเลชันด้วยความกว้างนั้น เราจะกำหนดเอาว่า 1 คือ แถบขาวหรือแถบดำที่กว้าง และ 0 คือ แถบขาวหรือแถบดำที่แคบ ดังนั้นการแทนตัวเลขสองตัวที่เหมือนกันและอยู่ติดกัน จึงต้องมีการ "สับเปลี่ยน" ตัวอย่างเช่น เลข 0 สองตัวติดกันจะต้องแทนด้วยแถบขาวและแถบดำ ไม่ใช่แถบดำหรือแถบขาวสองแถบติดกัน เพราะจะทำให้กลายเป็นการแทนเลข 1 หนึ่งตัว ซึ่งไม่ใช่เลข 0 สองตัวตามที่ต้องการไป เรายังมักเรียกรหัสแถบชนิดโมดูเลชันตามความกว้างว่าเป็นรหัสสองระดับ (แคบ/กว้าง)

สิ่งที่สองที่เราพูดกันก็คือ รหัสนั้นเป็นชนิดต่อเนื่องหรือไม่ต่อเนื่อง (Discrete) กล่าวคือ ในชนิดไม่ต่อเนื่องจะมีการแทรกช่องว่าง (เปรียบได้กับการเว้นวรรค) ระหว่างตัวอักษร ดังนั้นรหัสแถบชนิดนี้จะกินเนื้อที่มาก เพื่อเปรียบเทียบการกินเนื้อที่มากน้อย เขาจึงได้นิยามความหนาแน่นของรหัสขึ้น โดยให้มันเท่ากับ จำนวนอักษรต่อความยาวหนึ่งหน่วย (นิ้วหรือ ซม.) ความหนาแน่นนี้จะขึ้นด้วยตรงกับความกว้างของแถบขาวและแถบดำ ทั้งชนิดกว้างและชนิดแคบ พื้นที่ที่เป็นอักษรควบคุม (control character) และช่องไฟระหว่างอักษร

โดยทั่วไปแล้ว สำหรับรหัสที่มีความหนาแน่นสูง ความกว้างของแถบขาวหรือดำจะต่ำกว่า 0.009 นิ้ว (0.23 มม.) ซึ่งจะให้ความหนาแน่นของตัวอักษรสูงกว่า 8 ตัวอักษรต่อนิ้ว

โดยทั่วไป และสำหรับความหนาแน่นขนาดกลาง ความกว้างของแถบดำหรือแบบขาวจะอยู่ระหว่าง 0.009 นิ้ว ถึง 0.020 นิ้ว (0.23 มม. ถึง 0.50 มม.) ให้ความหนาแน่นอยู่ระหว่าง 4 ถึง 8 ตัวอักษรต่อนิ้ว และสุดท้ายสำหรับกรณีความหนาแน่นต่ำกว่า 4 ตัวอักษรต่อนิ้ว

2.6.1.3 ความแม่นยำในการอ่านรหัส

สำหรับพารามิเตอร์ต่อไปนั้นเกี่ยวข้องกับความแม่นยำแน่นอนในการอ่านรหัส ซึ่งได้แก่ ความละเอียด, ความแตกต่างของความเข้ม (Contrast) และความไม่สมบูรณ์ของแถบรหัส ความละเอียดนั้นจะหมายถึง ขีดความสามารถของตัวอ่านในการอ่านแถบดำหรือแถบขาวที่แคบที่สุด ดังได้กล่าวไปแล้วว่า ขึ้นอยู่กับขนาดของจุดลำแสงที่ตัวอ่านใช้สำหรับความแตกต่างของความเข้มนั้น เราวัดจาก C เท่ากับ พลังงานที่สะท้อนจากแถบสว่าง ลบ พลังงานที่สะท้อนจากแถบมืด หารด้วย พลังงานที่สะท้อนจากแถบสว่าง ซึ่ง C นี้ไม่ควรต่ำกว่า 0.7 สุดท้ายความไม่สมบูรณ์ของแถบรหัส มักจะเกิดจากความบกพร่องของการพิมพ์ ซึ่งอาจทำให้เกิดการบิดเบี้ยวของแถบ, ความกว้างของแถบไม่แน่นอน หรือความคมชัดไม่ดีพอ เป็นต้น จึงจำเป็นที่เราจะต้องเลือกเครื่องพิมพ์ให้เหมาะสมกับงานและรหัสที่ใช้

2.6.1.4 ความหลากหลายของรหัส

นอกจากนี้ รหัสยังมีลักษณะอื่นที่แตกต่างกันอีกเช่น เป็นรหัสแทนตัวเลข หรือรหัสแทนทั้งตัวเลขและตัวอักษร ความยาวของแถบรหัสคงที่หรือแปรเปลี่ยนได้ เป็นต้น การเลือกใช้นั้นก็ขึ้นอยู่กับลักษณะงาน โดยเราจะพิจารณาเลือกรหัสจากชุดตัวอักษรที่รหัสสามารถแทนได้ ความยากง่ายในการใส่รหัส ความแม่นยำของรหัส ความยืดหยุ่นต่อความเร็วที่ใช้ในการอ่าน และความต้านทานต่อความไม่สมบูรณ์ในการพิมพ์ เป็นต้น อย่างไรก็ตามรหัสที่ใช้กันแพร่หลายในปัจจุบันเห็นจะได้แก่ UPC (Universal Product Code), EAN (European Artich number), Codebar, "2 ใน 5" และรหัส 39

1. รหัส EAN/UPC เป็นรหัสแทนตัวเลขเท่านั้น แถบรหัสหนึ่งประกอบด้วยเลข 8 ตัว หรือ 13 ตัว แต่ขนาด 13 ตัวเป็นแบบที่ใช้กันแพร่หลายมากที่สุด แถบรหัสจะขึ้นต้นและลงท้ายด้วยรหัส 101 เสมอ ตัวเลข 13 หลักนี้จะถูกแบ่งเป็นสามส่วน ส่วนแรกประกอบด้วยเลข 2 ตัว ซึ่งบ่งบอกประเทศ ส่วนที่สองประกอบด้วยเลข 4 ตัว บ่งบอกผู้ผลิตและส่วนสุดท้าย ซึ่งแยกจากส่วนที่สองโดยมีรหัส 01010 เป็นตัวคั่นนั้น จะบ่งบอกรหัสตัวสินค้า รหัสแต่ละตัวจะใช้แถบ 7 แถบ แต่ละแถบมีความกว้างตายตัวเท่ากัน โดยแถบดำ

คือ 1 และแถบขาวคือ 0 รหัส EAN/UPC นี้เป็นรหัสที่ใช้กับสินค้าอุปโภคบริโภค และเป็นที่ใช้กันแพร่หลายทั่วโลก

2. รหัสตระกูล "2 ใน 5" สำหรับรหัส "2 ใน 5" ซึ่งตามความเป็นมาแล้ว เป็นรหัสชนิดแรกที่ถูกใช้อย่างเป็นกิจจะลักษณะ หนึ่งตัวรหัสจะประกอบด้วยแถบห้าแถบ ซึ่งสองในจำนวนนี้จะมีลักษณะผิดแผกจากที่เหลือ ซึ่งเราจะได้เห็นกันต่อไป รหัสในตระกูลนี้ได้แก่ "2 ใน 5 อุตสาหกรรม", "2 ใน 5 แมทริกซ์" และ "2 ใน 5 สอดแทรก" ทั้งหมดเป็นรหัสแทนตัวเลข

   - รหัส "2 ใน 5 อุตสาหกรรม" นั้น แถบรหัสหนึ่งจะมีความยาวระหว่าง 1 ถึง 32 ตัว ในรหัสชนิดนี้แถบดำเท่านั้นที่ถือเป็นองค์ประกอบของแถบรหัส โดยแถบดำแคบถือเป็น 0 และแถบดำกว้างถือเป็น 1 รหัส "2 ใน 5 อุตสาหกรรม" นี้ เป็นรหัสที่ง่ายต่อการพิมพ์ แต่ว่าขาดความแน่นอนในการอ่าน ดังนั้นจึงมีการเติมเอาอักษรควบคุมที่ท้ายแถบรหัส รหัสชนิดนี้ใช้กันแพร่หลายในโรงงานอุตสาหกรรมต่าง ๆ บนตั๋วเครื่องบิน และเครื่องแยกจดหมาย
   - สำหรับรหัส "2 ใน 5 แมทริกซ์" นั้น แถบดำและแถบขาวล้วนถือเป็นองค์ประกอบของรหัส หนึ่งตัวรหัสประกอบด้วยสามแถบดำและสองแถบขาว ระหว่างรหัสแต่ละตัวจะมีช่องไฟคั่น แถบรหัสจะขึ้นต้นและลงท้ายด้วยรหัส 10000 เสมอ การถือเอาแถบขาว ซึ่งก็คือ พื้นที่ที่ใช้ในการพิมพ์รหัสเข้าเป็นส่วนหนึ่งของรหัส ทำให้รหัสชนิดนี้กินเนื้อที่น้อยกว่ารหัสชนิดแรก จาก 28 ถึง 33 เปอร์เซ็นต์ ข้อเสียคือความต้านทานต่อความผิดพลาดจะลดต่ำลง
   - รหัส "2 ใน 5 สอดแทรก" นั้น อาจถือได้ว่าเป็นรหัสที่น่าสนใจที่สุดในรหัสตระกูลนี้ ในรหัสชนิดนี้แถบดำและขาวล้วนถือเป็นองค์ประกอบของรหัสเช่นเดียวกับ "2 ใน 5 แมทริกซ์" แต่จะไม่มีช่องไฟระหว่างรหัส และการใส่รหัสนั้นจะทำในลักษณะ "สอดแทรก" คือ อักษรตัวแรกจะถูกใส่รหัสด้วยรหัส "2 ใน 5 อุตสาหกรรม" โดยใช้แถบดำเป็นตัวประกอบ แต่ตัวอักษรตัวต่อมาจะถูกใส่รหัสด้วย "2 ใน 5 อุตสาหกรรม" ที่ใช้คราวนี้แถบขาวเป็นตัวประกอบ แถบขาวที่ได้มีห้าแถบด้วยกัน คือแบ่งเป็นสองแถบกว้างและสามแถบแคบ ซึ่งจะถูกแทรกเข้าสลับกับแถบดำห้าแถบที่ได้จากการใส่รหัสตัวอักษรแรก

แถบรหัสของ "2 ใน 5 สอดแทรก" นี้จะขึ้นต้นด้วยรหัส 0000 และลงท้ายด้วยรหัส 100 เมื่อเทียบกับรหัส "2 ใน 5 อุตสาหกรรม" รหัสชนิดนี้ให้ความหนาแน่นมากกว่าจาก 36 ถึง 42 เปอร์เซ็นต์ และจาก 10 ถึง 12 เปอร์เซ็นต์ เมื่อเทียบกับรหัส "2 ใน 5 แมทริกซ์" มันจึงเป็นที่ใช้กันอย่างแพร่หลายในวงการอุตสาหกรรม

3. รหัส 39 เป็นรหัสชนิดแรกที่ใช้แทนตัวอักษรด้วย ปัจจุบันได้มีรหัสซึ่งขยายจากรหัส 39 แล้ว คือ รหัส 128 รหัส 39 นั้น ประกอบด้วยสัญลักษณ์ 43 ตัว (เดิม 39 ตัว) ซึ่งแบ่งเป็นพยัญชนะ 26 ตัว ตัวเลข 10 ตัว และอักษรพิเศษที่เหลือรหัส 39 นี้สามารถถือเป็นรหัส "3 ใน 9) เพราะหนึ่งตัวรหัสประกอบด้วย 9 ตัวประกอบ โดยสามตัวในนั้นจะเป็นแถบกว้าง และอีกสองตัวจะเป็นแถบแคบ หนึ่งแถบรหัสจะมีหนึ่งถึงสามตัวอักษรเท่านั้นซึ่งตามด้วย Check digit ดังนั้นรหัส 39 จึงมีความแน่นอนในการอ่านสูง แต่เปลืองเนื้อที่ รหัสชนิดนี้มีใช้กันมากในอุตสาหกรรมอิเล็คทรอนิค โดยใช้ในการแยกชนิดแผงวงจร

4. Codabar เป็นรหัสสำหรับตัวเลขและมีความยาวของแถบรหัสจาก 1 ถึง 32 ตัว เป็นรหัสที่ใช้ในธนาคารเลือดของสหรัฐอเมริกา และในอุตสาหกรรมยาและทางการแพทย์ หนึ่งตัวรหัสประกอบด้วย 7 บิท ซึ่งแบ่งเป็น 4 แถบดำ และ 3 แถบขาว แถบดำหรือขาวที่แคบแทน 0 และแถบดำหรือขาวกว้างแทน 1

5. รหัสในตระกูลอื่น นอกเหนือจากรหัสที่กล่าวแล้ว ยังมีรหัสอื่น ๆ ที่เราสามารถพบเห็นได้ เพียงแต่ว่าไม่เป็นที่แพร่หลายเท่าพวกแรกเท่านั้น รหัสเหล่านี้ได้แก่ รหัส 128, รหัส "2 ใน 7" และรหัส 11

   - รหัส 128 เป็นรหัสที่ใหม่มาก มันประกอบด้วยชุดตัวอักษร 128 ตัวของแอสกี (ASCII) รหัสชนิดนี้เป็นรหัสต่อเนื่องและให้ความแน่นอนในการอ่านสูงมาก ส่วนรหัส "2 ใน 7" เป็นรหัสชนิดโมดูเลชัน ตามความกว้างสำหรับแทนตัวเลขและอักษรพิเศษ 6 ตัว คือ $-: /. และ + ความกว้างของแถบในรหัสชนิดนี้ไม่ได้ถูกกำหนดไว้เพียงขนาดเดียว แต่มีถึง 18 ขนาดให้เลือกใช้ สามารถให้ความหนาแน่นได้ถึง 11 ตัวอักษรต่อนิ้ว แต่ว่ามีกฎเกณฑ์ที่ซับซ้อนจึงไม่เป็นที่นิยมใช้กันมากนัก และสุดท้ายรหัส 11 เป็นรหัสตัวเลขเช่นกัน มีลักษณะใกล้เคียง

กับรหัส "2 ใน 5 แมทริกซ์" หนึ่งรหัสประกอบด้วย 3 แถบดำ และ 2 แถบขาว

- รหัส 11 นี้ให้ความหนาแน่นสูงมาก เนื่องจากว่ามีการออกแบบให้สัดส่วนของแถบกว้างต่อแถบแคบดีที่สุดในแต่ละรหัส แต่ผลก็คือความซับซ้อนซึ่งทำให้สู้แบบ "2 ใน 5" ไม่ได้

2.6.2 สมาร์ทการ์ด (อ.ภูมิ เหลืองจามีกร : 2548)

บัตรที่ใช้เป็นเครื่องยืนยันตัวบุคคลและเก็บข้อมูลต่างๆเช่นจำนวนเงิน สมาชิกภาพ ฯลฯ ปัจจุบันมีมากมายหลายชนิดเช่นบัตรแม่เหล็กที่ใช้เป็นบัตร ATM ฯลฯ แต่ทว่า จากปัญหาทางด้านความปลอดภัย ก็ได้มีการพัฒนาเทคโนโลยีอื่นๆขึ้นเพื่อนำมาแทนที่ สมาร์ทการ์ดก็เป็นเทคโนโลยีใหม่อย่างหนึ่งซึ่งกำลังมีบทบาทมากขึ้นในปัจจุบัน

2.6.2.1 <u>สมาร์ทการ์ดคืออะไร</u>

สมาร์ทการ์ด หรือบางแห่งจะเรียกว่า IC การ์ด คืออุปกรณ์อีเล็คโทรนิคขนาดเล็กที่ใช้บรรจุข้อมูล โดยมากจะมาในรูปของบัตรโดยมีตัวสมาร์ทการ์ดฝังอยู่ข้างใน ตัวอย่างสมาร์ทการ์ดที่เห็นกันทั่วไปก็ได้แก่ ซิมการ์ดของโทรศัพท์มือถือเป็นต้น

2.6.2.2 <u>สมาร์ทการ์ดมีประเภทไหนบ้าง</u>

ถ้าจะแบ่งสมาร์ทการ์ดออกตามคุณสมบัติแล้ว จะแบ่งออกได้เป็นสองประเภทคือ

1. ประเภทใช้เก็บข้อมูลอย่างเดียว สมาร์ทการ์ดประเภทนี้แทบจะไม่แตกต่างอะไรกับ บัตรแม่เหล็กโดยทั่วไป แค่มีความสามารถที่จะบรรจุข้อมูลลงในตัวบัตรมากกว่าเท่านั้น
2. ประเภทที่มีไมโครโปรเซสเซอร์ในตัว สมาร์ทการ์ดประเภทนี้จึงจะคุณสมบัติแตกต่างจากบัตรแม่เหล็กดั้งเดิม กล่าวคือไมโครโปรเซสเซอร์ในตัวสมาร์ทการ์ด จะสามารถเขียนโปรแกรมและประมวลผลในตัวของสมาร์ทการ์ดเอง ได้โดยไม่ต้องอาศัยอุปกรณ์ภายนอกช่วย ซึ่งจุดนี้จะเป็นส่วนช่วยเพิ่มความปลอดภัยให้แก่ตัวสมาร์ทการ์ด เช่น สามารถใส่โปรแกรมตรวจสอบรหัสเข้าถึงข้อมูลไว้ภายใน ซึ่งอาจจะกำหนดไว้ว่า ทุกครั้งที่จะมีการอ่านข้อมูลต้องใส่รหัสผ่าน ดังนั้น แม้จะนำสมาร์ทการ์ดไปใช้

กับเครื่องอ่านที่ได้รับการดัดแปลง ก็ไม่สามารถจะเข้าถึงข้อมูลได้เพราะถูกโปรแกรมซึ่งบันทึกอยู่ในตัวบัตรป้องกันไว้เป็นต้น

รูปที่ 2.25 ตัวอย่างสมาร์ทการ์ดแบบสัมผัส

รูปที่ 2.26 ตัวอย่างสมาร์ทการ์ดแบบไม่สัมผัส

ถ้าจะแบ่งสมาร์ทการ์ดออกตามลักษณะการใช้งานแล้วก็แบ่งออกได้เป็นสองประเภทคือ

1. สมาร์ทการ์ดแบบสัมผัส (Contacted Smart Card) การเขียนอ่านจะทำโดยจะต้องมีขั้วสัมผัสกับตัวสมาร์ทการ์ด โดยปกติจะเป็นในลักษณะ เสียบตัวบัตรลงในช่องแล้วจึงอ่านข้อมูล ดังแสดงในรูปที่ 2.25
2. สมาร์ทการ์ดแบบไม่สัมผัส (Contactless Smart Card) เป็นสมาร์ทการ์ดที่สามารถเขียนอ่านได้โดยไม่ต้องมีการสัมผัสกับตัวสมาร์ทการ์ด จะอาศัยคลื่นวิทยุในการติดต่อ วิธีใช้ก็เพียงแต่นำตัวสมาร์ทการ์ดเข้าไปใกล้ตัวอ่านเท่านั้น ดังแสดงในรูปที่ 2.26

นอกจากนี้ยังมีสมาร์ทการ์ดรุ่นใหม่ที่สามารถอ่านเขียนได้ทั้งสองแบบในตัวเดียวกันออกมาแล้วด้วย

2.6.2.3 สมาร์ทการ์ดใช้ที่ไหนบ้าง

ในประเทศไทย ตัวอย่างการใช้สมาร์ทการ์ดที่เห็นมากที่สุดก็คงเป็นซิมการ์ดของโทรศัพท์มือถือ แต่ในต่างประเทศ วงการที่ใช้ สมาร์ทการ์ดหลักๆ ก็คงจะเป็นธนาคารและ การคมนาคม ในวงการธนาคาร นอกจากนำมาใช้เป็นเป็นบัตรATM หรือ บัตรเครดิต ทดแทนบัตรแม่เหล็กแล้ว ยังมีการนำมาเป็นบัตรเติมเงินเพื่อใช้แทนเงินสดอีกด้วย ซึ่งก็เป็นลักษณะเดียวกับที่ใช้วงการคมนาคม ซึ่งใช้เป็นบัตรเติมเงิน(พรีเพด) เช่นบัตรเติมเงินค่าตั๋วรถไฟฟ้าเป็นต้น

ความแตกต่างของบัตรATMกับบัตรเติมเงินก็คือ บัตรATMจะใช้เป็นเพียงรหัสบ่งชี้บุคคลที่เป็นเจ้าของบัญชีเท่านั้น การตัดยอดจะต้องทำผ่านเครือข่ายเข้าไปสู่บัญชีของเจ้าของบัตรโดยตรง ซึ่งต่างจากบัตรเติมเงินก็คือบัตรเติมเงินจะเก็บยอดเงินไว้ในตัวบัตร การตัดยอดใช้จ่ายจะทำโดยตรงที่ตัวบัตรเลย ดังนั้นจึงไม่จำเป็นต้องมีเครือข่ายรองรับ

ลักษณะการใช้ก็มักจะแยกตามธุรกิจเช่นกัน โดยปกติแล้ว ตัวอ่านสมาร์ทการ์ดแบบไม่สัมผัสจะมีราคาที่สูงกว่า แต่ก็จะมีข้อดีที่สามารถอ่านได้อย่างรวดเร็ว ไม่ต้องอาศัยการเสียบสมาร์ทการ์ดเข้าออก ดังนั้นในวงการคมนาคม เพื่อความสะดวกและรวดเร็วในการที่นำเอาสมาร์ทการ์ดมาใช้แทนตั๋ว ก็มีแนวโนวที่จะใช้สมาร์ทการ์ดแบบไม่สัมผัสมากกว่า ส่วนในวงการธนาคารที่ไม่จำเป็นต้องใช้ความเร็วมากนัก ก็มักจะใช้สมาร์ทการ์ดแบบสัมผัส

2.6.2.4 สมาร์ทการ์ดกับผู้ประกอบการ

เท่าที่กล่าวมาข้างต้นอาจเห็นว่าเป็นเรื่องไกลตัว หรือไม่ก็อาจจะเข้าใจเพียงว่าเราจะเป็นเพียงผู้ใช้ของระบบขนาดใหญ่เท่านั้น แต่ความจริงแล้วการพัฒนาระบบสมาร์ทการ์ดเข้ามาใช้ในธุรกิจของตนนั้น มิใช่เรื่องที่ยากเกินความสามารถ หรือแม้กระทั้งการจ้างบริษัทที่เชี่ยวชาญเข้ามาพัฒนาระบบให้ก็อยู่ในขั้นที่กระทำได้ เพียงแต่ว่าตัวผู้ประกอบการเองจะต้องมองคุณค่าของตัวสมาร์ทการ์ดให้ออก แล้วนำไปประยุกต์ใช้ให้เหมาะแก่กิจการของตน

2.7 เอกสารและงานวิจัยที่เกี่ยวข้อง

ต้นกำเนิดของระบบสารสนเทศในโรงพยาบาลได้เริ่มต้นมา 2 ทศวรรษแล้ว ซึ่งจากการวิจัยถึงข้อมูลลักษณะการดำเนินการเรื่องสารสนเทศนั้น(Peter L. Reichertz) ได้กล่าวอ้างเกี่ยวกับระบบสารสนเทศในโรงพยาบาลไว้โดยเสนอในมุมมองของะบบสารสนเทศทั่วไปไว้ว่า

"ระบบสารสนเทศเป็นการเตรียมข้อมูลเพื่อให้พร้อมใช้สำหรับผู้ที่ต้องการใช้ข้อมูลนั้น โดยรูปแบบในการเสนอข้อมูลต้องสอดคล้องกับความต้องการของผู้ใช้ด้วย ซึ่งลักษณะของระบบสารสนเทศภายในโรงพยาบาลก็มีจุดประสงค์และความต้องการไม่ต่างไปจากระบบสารสนเทศภายในโรงงานเลย"

ในปี 1970 D.A.B.Lindberg ได้กล่าวไว้เสนอประวัติไว้ว่า ระบบสารสนเทศในโรงพยาบาลยังเป็นเรื่องของทางด้านการเมืองการปกครอง แต่ก็ได้มีการพัฒนามาเพื่อมุ่งเน้นด้านการรักษาผู้ป่วยและการบริหารเวชภัณฑ์ในที่สุด โดยประเด็นหลักในการพัฒนายุคแรกเริ่มนั้นเสนอในมุมมองของการบูรณาการข้อมูลเพื่อใช้อ้างอิงตรวจสอบและรักษาผู้ป่วยนั้นๆ แต่ก็ไม่สำเร็จเพราะเนื่องจากขาดอุปกรณ์อิเล็กทรอนิกและโปรแกรมที่สามารถรองรับการทำงานได้

ในปี 1978 Peter L. Reichertz ได้เสนอในเรื่องมุมมองของข้อมูลที่ควรเตรียมไว้เพื่อให้พร้อมสำหรับการรักษาผู้ป่วยซึ่งลักษณะของข้อมูลมี 3 เกณฑ์หลักคือ มุมมองของข้อมูลทางด้านจุดประสงค์ (Objectives) เพื่อกำหนดจุดประสงค์ในด้านการดำเนินการวางขอบเขตและผลัพธ์ ไว้ โดยต้องมีการเตรียมข้อมูลเพื่อสนับสนุนการดำเนินงาน (Drives) และผลกระทบที่ได้จากการดำเนินงานในภายหลังด้วย (Effects)

ในปี 1984 B.G. Winfree ได้เสนอกระบวนการหลักของระบบสารสนเทศภายในโรงพยาบาลไว้แบ่งออกเป็น 5 ส่วนหลักดังนี้

- การบริหารด้านผู้ป่วย ได้แก่งานทางด้านการลงทะเบียนผู้ป่วย การเก็บข้อมูลการรับบริการ และการคิดค่ารักษาพยาบาล
- การบริหารทางด้านการเงิน ได้แก่ การวิเคราะห์ทางด้านการเงินในส่วนต่างๆ การทำงบดุล การจัดการบัญชีรายรับรายจ่าย ฯลฯ
- การบริหารด้านเวชภัณฑ์ ได้แก่ การควบคุมคงคลัง การจัดระบบจุดจำหน่ายยา การจัดการด้านธนาคารเลือด และงานด้านการจัดซื้อ

- การบริหารด้านข้อมูลบุคลากร ได้แก่ การลงทะเบียนบุคลากรและการกำหนดอัตราค่าแรง
- การบริหารด้านองค์กร ได้แก่ การจัดตารางเวลาบุคลากรและการจัดการควบคุมการใช้งานเตียง

โดยจากแนวคิดในการแบ่งงานภายกระบวนงานภายในโรงพยาบาล สามารถแบ่งลักษณะการดำเนินงานภายในโรงพยาบาลออกเป็น 5 ส่วนหลักได้ ดังนี้

- การลงทะเบียนผู้ป่วย
- การจองการทำหัตถการและการจำหน่ายผู้ป่วย
- การบันทึกประวัติการรักษาผู้ป่วย
- การบริหารเวชภัณฑ์
- การคิดค่ารักษาพยาบาล

ซึ่งงานวิจัยฉบับนี้ได้มุ่งเน้นการจัดการในด้านการลงทะเบียนผู้ป่วย การจองการทำหัตถการ การจำหน่ายผู้ป่วยและการคิดค่ารักษาพยาบาล โดยมีวรรณกรรม ที่มีการระบุถึงแนวคิดในส่วนงานต่อมาโดย Jan M.H. Vissers:1994 ได้เสนอแนวคิดในการบริหารจัดการกำหนดทรัพยากรที่ใช้ในโรงพยาบาล โดย กล่าวถึงวิธีการในการจัดสรรทรัพยากรที่ใช้ในโรงพยาบาลเพื่อเป็นการลดปัญหาเรื่องการขาดแคลนของปริมาณในการให้บริการของโรงพยาบาล ซึ่งมุมมองในการวิเคราะห์ในครั้งนี้ไม่ได้รวมถึง ความสูญเสียบางประการที่เกิดขึ้นในกรณีเช่นเดียวกับการผลิตในวงการอุตสาหกรรม โดยในการวิเคราะห์ครั้งนี้เน้นไปด้านทรัพยากรหลักที่นำไปใช้ (Leading Resources) ตัวอย่างเช่น ห้องผ่าตัด คลินิกผู้ป่วยนอก หรือ บุคลากรที่ให้บริการ ซึ่งทรัพยากรหลักเหล่านี้เปรียบเสมือน จุดนับในการให้บริการต่อ 1 ครั้ง โดยคำนึงถึงลักษณะการนำทรัพยากรรอง (Following Resources) เพื่อมาตอบรับกับการให้บริการของทรัพยากรหลัก ตัวอย่างเช่น เตียง พยาบาล

การวิเคราะห์โครงสร้างของปริมาณการให้บริการของโรงพยาบาล ใช้การวิเคราะห์โดยมีมุมมองของทรัพยากรที่ใช้แบ่งออกเป็น ทรัพยากรหลัก และทรัพยากรรอง ตามที่ได้กล่าวไว้ข้างต้นโดยทรัพยากรหลักที่มีส่วนสำคัญหลักในการวิจัยครั้งนี้ได้แก่ ห้องผ่าตัด ห้องพักคนไข้ จุดบริการแผนกผู้ป่วยนอก จุดคัดกรองผู้ป่วย แพทย์และบุคลากรหลัก ซึ่งทรัพยากรเหล่านี้จำเป็นต้องใช้ในการจัดสรรจากรูปแบบการให้บริการต่างๆ โดยวิธีการจัดการเกี่ยวกับการบริการซึ่งมีรูปแบบในการบริการที่เข้ามาในระบบนั้น ได้ใช้วิธีการแก้ไขโดยให้บริการแบบการประมวลผล

แบบกลุ่ม (Batch Processing) ซึ่งเป็นการเก็บข้อมูลร้องขอความต้องการในการรับบริการของผู้ป่วยไว้สักระยะหนึ่งก่อนจุดสรรเพื่อให้บริการจริง ซึ่งส่วนใหญ่นั้นใช้เวลาในการรวบรวมข้อมูลก่อนการประมวลผลประมาณ 1 อาทิตย์ โดยวิธีการทำแบบนี้สามารถใช้ได้สำหรับความต้องการรับบริการในช่วงระยะเวลาสั้นๆ เท่านั้น โดยในการประมวลผลนั้น จำเป็นต้องมีการกำหนดควบคุมตารางเวลาของแต่ละทรัพยากรก่อนการประมวลผล เพื่อเป็นการกำหนด ขอบเขตของเวลาในการให้บริการ ซึ่งจำเป็นต้องกำหนด ตารางเวลาของทั้งจุดบริการ และ ของทั้งพนักงาน

โดยการจัดการทรัพยากรในโรงพยาบาลนั้นได้มีการนำเสนอกระบวนการในการแบ่งสรรทรัพยากรไปสู่การทำหัตถการต่างๆทางการแพทย์ โดยในการแบ่งสรรทรัพยากรนั้น ได้มีแนวคิดที่จะทำ สมองเทียม AI (Artificial Intelligence) เข้ามามีส่วนร่วมในการทำงาน โดย Angelo Oddi :2000 แต่ว่ามีเหตุผลหลัก 2 ประการที่ทำให้การนำ สมองเทียม เข้ามาใช้งานนั้นลำบากได้แก่

1. ธรรมชาติของข้อกำหนดที่ใส่ให้กับสมองเทียมไม่มีการคำนึงถึงปัจจัยทางด้านความปลอดภัยทางด้านความพึงพอใจของพนักงาน และส่วนอื่นๆที่เกี่ยวข้องกับธรรมชาติในการทำงานดังนั้นจังเป็นปัจจัยที่ยากต่อการใส่ข้อกำหนด ที่ใช้ควบคุม
2. ปัจจัยที่ไม่คาดคิดที่อาจส่งผลต่อการตัดสินใจซึ่งโดยปกติแล้วเราไม่สามารถรู้ได้หรือว่าใส่ข้อกำหนดที่เกี่ยวข้องทั้งหมดในการตัดสินใจได้จึงทำให้การคำนวณของสมองเทียม อาจจะไม่สามารถให้ผลการตัดสินใจที่ดีได้

แต่ถึงแม้ปัญหาเหล่านี้จะส่งผลต่อการตัดสินใจในการจัดตารางการจัดสรรการใช้ทรัพยากร แต่การจัดสรรยังควรจะให้มีการจัดทำขึ้นอยู่ดี เพื่อช่วยให้การทำงานสะดวกขึ้น โดยผลจากการทำงานส่วนใหญ่ เกิดจากบุคลากร ใช้ความรู้และประสบการณ์ทั้งหมด ในการจัดขึ้น ดังนั้น ควรเริ่มจัดโดยการเก็บรวมรวมข้อมูล และนำข้อมูลเหล่านั้นมาแปลงเป็นข้อกำหนดทางคณิตศาสตร์เพื่อให้กับสมองเทียมในการคำนวณ

โดยโปรแกรมที่ช่วยในการจัดตารางงานมีปัจจัยที่ต้องศึกษา 2 ปัจจัยคือ

1. ทางด้านวิธีการดำเนินงาน และรายละเอียดในการทำงานนั้น ๆ
2. ทางด้านทรัพยากรซึ่ง ในแต่ละทรัพยากรก็จะมีคุณสมบัติที่เกี่ยวข้องกับการกำหนด ข้อกำหนดในสมการต่างๆ เช่น เวลาที่สามารถได้ , จำนวน

ความสามารถในการทำงานที่สามารถทำได้ หรือ ข้อกำหนดในการใช้งานร่วมกับทรัพยากรอื่นๆ เป็นต้น

ซึ่งในการจัดการแก้ไขปัญหาโดยนำปัจจัย เหล่านี้มาจับกับข้อกำหนดต่างๆที่กำหนด ไว้ (เช่นการใช้ทรัพยากรให้มีการทำให้เกิดประโยชน์สูงสุด (Utilization) ต่างๆที่ได้กำหนดไว้ ทั้งนี้ควรแบ่ง ประเภทของข้อกำหนด ออกเป็น 2 ประเภทคือ ข้อกำหนดแบบแข็ง (ข้อกำหนดที่ไม่สามารถเปลี่ยนแปลงหรือ ยืดหยุ่นได้) ข้อกำหนดแบบอ่อน (สามารถแก้ไขข้อกำหนดได้ ยืดหยุ่นได้) ซึ่งในการกำหนด รูปแบบของข้อกำหนด จะส่งผลต่อผลลัพท์ในการจัดตารางงานที่แตกต่างกันได้

ผลลัพท์ที่ได้จากการจัดตารางการดำเนินงานในขั้นเริ่มต้นอาจจะกำหนดแนวทางในการหาคำตอบได้ดังนี้

1. หาคำตอบเบื้องต้นเพียงแค่เพื่อการหาระยะเวลาที่สามารถจองทรัพยากรได้ออกมาก่อน โดยสนใจปัจจัยด้านเวลาที่ว่างสามารถจองได้เท่านั้น
2. หาคำตอบจากวิธีการเบื้องต้นโดยเพิ่ม ข้อกำหนดด้านต่างๆเข้าไปเรื่อยๆ เพื่อตัดช่วงของผลลัพธ์ให้มีช่วงที่เหมาะสมต่อการทำงานมากที่สุดจากการใส่ข้อกำหนดเพิ่มขึ้นเข้าไป

ซึ่งงานวิจัยฉบับนี้ได้มีการเสนอชุดของคำสั่งการสร้างชุดของคำสั่งที่ใช้ในการหาผลลัพท์ในการจัดตารางการทำงานมีหลายรูปแบบต่างๆ ได้แก่ 1.Greedy Constructive Algorithm , 2. Local Search Algorithm รวมไปถึงการแสดงผลในการทำงานและการจัดตารางให้อยู่ในรูปแบบของตารางการทำงานที่ดูง่ายในรูปแบบของแผนภูมิและ ตารางเวลา ที่สามารถเข้าใจและใช้งานได้ง่ายขึ้น โดยผลจากงานวิจัยข้างต้นที่ได้นำเสนอไปนั้น จึงได้นำไปเป็นแนวคิดในการออกแบบการทำงานซึ่งเป็นไปในลักษณะของการนำเสนอเพื่อช่วยเหลือการทำงานหลักภายในระบบเท่านั้น ไม่ได้เป็นการบังคับการทำงานของระบบ ซึ่งจะทำให้การทำงานนั้นมีความยืดหยุ่นและรองรับการทำงานที่มีปัจจัยและซับซ้อนได้

และในส่วนของการจำหน่ายผู้ป่วยเพื่อเข้ารักษาในจุดบริการต่างๆ Peter L. Reichertz:2006 ได้เสนอหลักการของการรับผู้ป่วยเข้ารับบริการ ซึ่งมีประเด็นหลักใน 3 ส่วน ตามระบบ A/D/T System ซึ่งระบบที่ได้กล่าวถึงนี้เป็นกระบวนงานเกี่ยวข้องกับทางด้านการรับผู้ป่วยเพื่อเข้ารักษา (Admission) การจำหน่ายผู้ป่วยออก (Discharge) และการส่งตัวผู้ป่วย (Transfer) โดยหน้าที่การทำงานหลักของระบบมีดังนี้

1. เพื่อตรวจสอบ หมายถึงการตรวจสอบว่าผู้ป่วยได้เข้ามารับบริการในแต่ละครั้งมีกิจกรรมใดบ้างที่ผู้ป่วยได้รับ โดยให้กำหนดหมายเลขอ้างอิงเพื่อใช้ในระบบในการเข้ารับบริการแต่ละครั้ง
2. เพื่อเป็นจุดเริ่มของการเก็บข้อมูลการทำงานในส่วนต่างๆที่เกิดขึ้นระหว่างผู้ป่วยกับระบบสารสนเทศในโรงพยาบาลได้
3. เพื่อสามารถสร้างเครื่องมือหรือเอกสารเพื่อใช้ระบุถึงกระบวนการรับเข้าหรือจำหน่ายออกของผู้ป่วยที่เข้ามารักษาได้