บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

1) ระบบการจัดทำงบประมาณ

ระบบโปรแกรมที่พัฒนาขึ้น ต้องสนับสนุนวัตถุประสงค์ของการจัดทำงบประมาณแบบแผนงานตามระบบที่สำนักงบประมาณใช้อยู่ ซึ่งกำหนดสัดส่วนของวงเงินงบประมาณรายจ่ายแต่ละด้าน แต่ละกระทรวง ในลักษณะการวางแผนมหภาค ผสมผสานกับการวิเคราะห์รายละเอียดคำขอตั้งงบประมาณ ของกรม จากระดับรายละเอียดสรุปขึ้นไปเป็นยอดระดับแผนงานและด้านตามลำดับ (Bottom-Up Process) โดยต้องสอดคล้องกับแนวนโยบายของรัฐบาล และแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ (Top-Down Process) (สำนักงบประมาณ, 2530 และ 2542)

การจัดทำงบประมาณแบบแผนงาน จะมองภาพรวมของงบประมาณทั้งประเทศ เรียงลำดับจากหลักไปย่อยตามโครงสร้าง ดังนี้

1. ด้าน (Sector)
2. สาขา (Sub-Sector)
3. แผนงาน (Program)
4. แผนงานรอง (Sub-Program)
5. งาน/โครงการ (Work plan/Project)

ภายใต้งาน/โครงการ ยังจำแนกเป็นหมวดรายจ่ายอีก 11 หมวดรายจ่าย คือ หมวดเงินเดือน หมวดค่าจ้างประจำ หมวดค่าจ้างชั่วคราว หมวดค่าตอบแทน หมวดค่าใช้สอย หมวดค่าวัสดุ หมวดค่าสาธารณูปโภค หมวดค่าครุภัณฑ์ หมวดค่าที่ดินและสิ่งก่อสร้าง หมวดเงินอุดหนุน และหมวดรายจ่ายอื่น โดยแต่ละหมวดรายจ่าย ยังสามารถแจกแจงเป็นรายการงบประมาณภายใต้หมวดนั้น ๆ ได้อีก ซึ่งเป็นรายละเอียดระดับรายการที่ใช้สำหรับการบริหารงบประมาณตามที่ปรากฏอยู่ในเอกสารงบประมาณฉบับปกสีฟ้า

2) ทฤษฎีเทคโนโลยีเชิงวัตถุ

เนื่องจากปัญหาในการพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่และมีความซับซ้อนมาก หมายถึงค่าใช้จ่ายสูงและการใช้เวลามาก ตลอดทั้งการบำรุงรักษาและการแก้ไขไม่สามารถทำได้โดยง่าย ซึ่งจากปัญหาดังที่กล่าวมาแล้วจึงมีผู้คิดค้นวิธีการ หรือเทคโนโลยีสำหรับการพัฒนาระบบขึ้นมาใหม่ เรียกการพัฒนาซอฟต์แวร์แบบเทคโนโลยีเชิงวัตถุ ซึ่งมีนักพัฒนาระบบเป็นจำนวน

มากนำมาใช้เป็นเครื่องมือในการพัฒนา การพัฒนาและออกแบบระบบสามารถทำได้โดยง่าย ซอฟต์แวร์ที่พัฒนาขึ้นนำมาบำรุงรักษา แก้ไขและการนำกลับมาใช้ใหม่ทำได้ง่าย

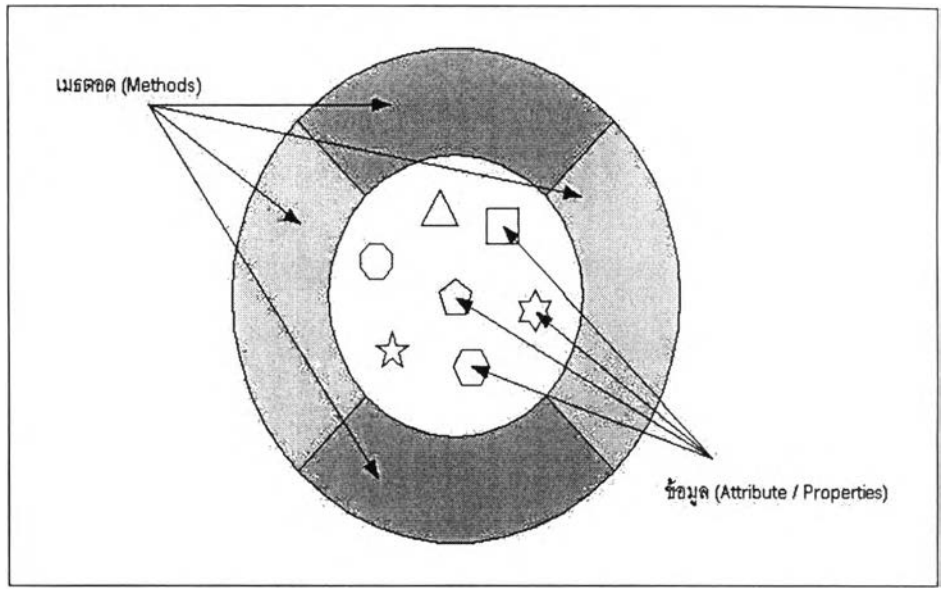
เทคโนโลยีเชิงวัตถุเป็นแนวคิดของการพัฒนาโปรแกรมที่ผนึกเอาข้อมูลและโปรแกรมเข้าด้วยกันเป็นวัตถุ และกำหนดตัวประสานระหว่างวัตถุต่าง ๆ โดยถ้าวัตถุหนึ่งต้องการจะใช้ข้อมูลอีกวัตถุหนึ่งจะต้องติดต่อผ่านตัวประสานนี้เท่านั้น จากหลักการนี้ทำให้พบว่าการเปลี่ยนแปลงแก้ไขโปรแกรมใด ๆ หรือวัตถุใด ๆ จะกระทบกระเทือนต่อวัตถุอื่นหรือโปรแกรมอื่นน้อยมาก ซึ่งทำให้การแก้ไขโปรแกรมจะกระทำได้โดยสะดวก ง่ายสำหรับการบำรุงรักษา นอกจากนี้ยังมีคุณสมบัติในการถ่ายทอด (Inheritance) คุณสมบัติจากวัตถุหนึ่งไปยังอีกวัตถุหนึ่งทำให้ตรงกับแนวคิดในการนำซอฟต์แวร์กลับมาใช้ใหม่

2.1 การพัฒนาระบบงานเชิงวัตถุ

ในการพัฒนาระบบงานเชิงวัตถุ เป็นการนำเอาแนวคิดเชิงวัตถุมาใช้ในการพัฒนาระบบงาน โดยมีองค์ประกอบในการพัฒนาระบบงานเชิงวัตถุดังต่อไปนี้

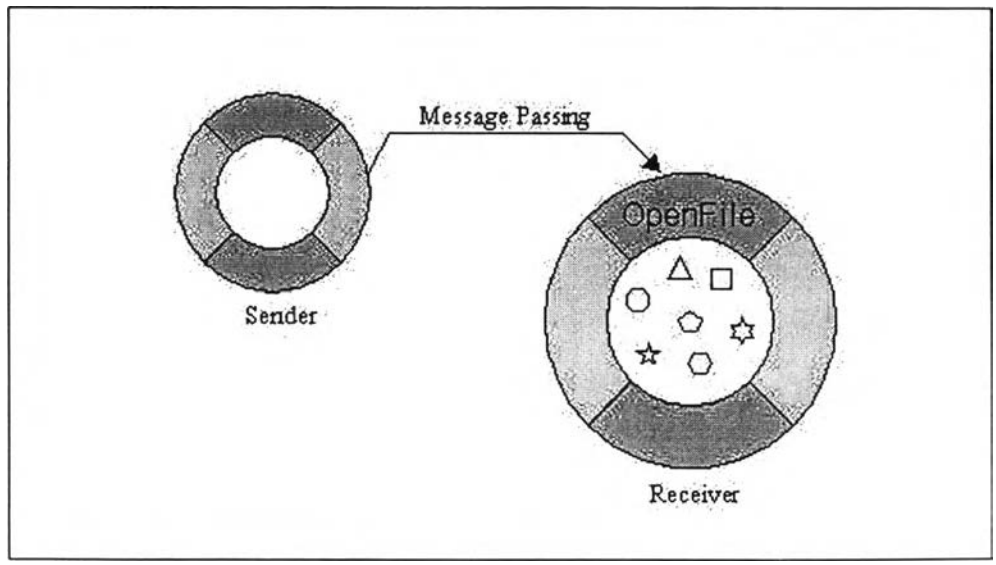
2.1.1 ออบเจ็กต์ (Object)

คำว่าวัตถุหรือออบเจ็กต โดยทั่วไปจะหมายถึง ทุกสิ่งทุกอย่างที่อยู่ล้อมรอบตัวเรา ที่เรามองเห็น สัมผัสหรือรู้สึกถึงมันได้ ไม่ว่าจะเป็นรถยนต์ โต๊ะทำงาน หรือแม้แต่เศษกระดาษชิ้นเล็ก ๆ ที่เราใช้ขีดๆ เขียนๆ สารพัดในระหว่างการทำงานก็ตาม แต่ในด้านที่เกี่ยวกับการพัฒนาระบบงานนั้น ออบเจ็กต์ จะหมายถึง สิ่งที่เราสนใจทั้งที่จับต้องได้ จับต้องไม่ได้ รวมไปถึงเหตุการณ์ต่างๆ ที่เกิดขึ้นได้ ที่ประกอบด้วย ข้อมูล และพฤติกรรม (Behavior) ที่มีผลต่อการเปลี่ยนแปลงของข้อมูลของมันเอง โดยใช้หลักการมองทุกอย่างในระบบให้เป็นวัตถุที่ประกอบด้วย ข้อมูลและพฤติกรรม ซึ่งข้อมูลของออบเจ็กต์จะเป็นตัวแสดง คุณสมบัติของออบเจ็กต์ที่เรียกว่า แอทริบิวต์ (Attribute) หรือ พรอพเพอร์ตี้ (Properties) ซึ่งจะบอกให้เราทราบว่าออบเจ็กต์ที่เราสนใจอยู่ เก็บข้อมูลหรือข่าวสารใดไว้บ้าง ส่วนพฤติกรรมของออบเจ็กต์จะเป็นตัวแสดงให้ทราบว่าออบเจ็กต์นั้นมีความสามารถให้บริการอะไรได้บ้าง ซึ่งเราเรียกว่าเมทธอด (Method) หรือ ฟังก์ชั่น (Function) ลักษณะของออบเจ็กต์แสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงภาพของออบเจ็กต์

จากรูปที่ 2.1 โดยหลักการเชิงวัตถุนั้น การกระทำใดๆ ก็ตามกับออบเจ็กต์ เช่นการเปลี่ยนแปลงสถานะ หรือการเปลี่ยนแปลงข้อมูลของออบเจ็กต์ จะต้องเรียกให้เมทธอดของออบเจ็กต์นั้นๆ เป็นผู้ดำเนินการเท่านั้น เรียกว่าการใช้บริการ(Call Service) โดยการส่งผ่านข้อความ(Message Passing) ไปยังเมทธอดของออบเจ็กต์นั้น ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการเรียกใช้บริการโดยการส่งข้อความระหว่างออบเจ็กต์

2.1.2 คลาส (Classes)

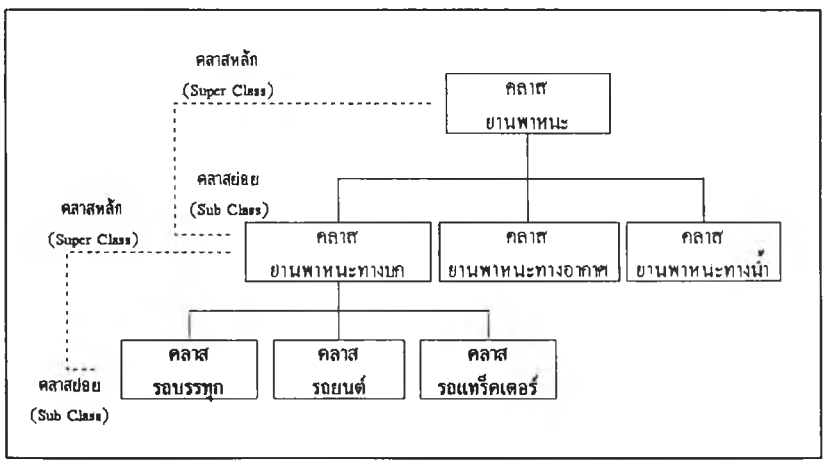
คลาส คือพิมพ์เขียวที่เอาไว้สร้างออบเจ็กต์ เปรียบเหมือนแบบพิมพ์เขียวที่เราเอาไว้สร้างบ้าน ซึ่งเราไม่สามารถเข้าอยู่อาศัยในแบบพิมพ์เขียวของบ้านได้ จนกว่าเรานำแบบพิมพ์เขียวมาสร้างเป็นบ้านแล้วจึงจะเข้าอยู่อาศัยใช้ประโยชน์ได้ เช่นเดียวกับที่เราไม่สามารถนำคลาส มาใช้งานได้โดยตรง จะต้องนำคลาสมาสร้างเป็นออบเจ็กต์เสียก่อน

2.1.3 การหุ้มห่อและการซ่อนข่าวสาร (Encapsulation and Information Hiding)

คือ การกำหนดให้ทุกๆ ปฏิบัติการที่จะกระทำกับข้อมูลของออบเจ็กต์ จะต้องดำเนินการโดยผ่าน เมทธอด ของออบเจ็กต์นั้นเท่านั้น ซึ่งหมายถึง การแยกส่วนติดต่อกับภายนอก (Interface) ออกจากส่วนดำเนินการภายในของออบเจ็กต์ ทำให้สามารถปกป้องข้อมูลของออบเจ็กต์ จากการกระทำจากภายนอกได้ เป็นผลให้บำรุงรักษาออบเจ็กต์ทำได้ง่ายขึ้น

2.1.4 การสืบทอด (Inheritance)

คือการที่คลาส ทุกๆ คลาส ในกลุ่ม มีการใช้โครงสร้างและพฤติกรรมร่วมกัน โดยคลาสเหล่านี้จะมีความสัมพันธ์แบบ อีสอะ (Is-a Relationship) กับอีกคลาสหนึ่งที่อยู่เหนือกว่า และเราเรียกคลาสที่อยู่เหนือกว่านี้ว่า คลาสพ่อ (Parent Class) หรือ คลาสหลัก (Super Class) ซึ่งจะถ่ายทอดคุณสมบัติทั้งแอททริบิวต์และเมทธอด มายังคลาสที่อยู่ต่ำกว่าที่เรียกว่า คลาสย่อย (Sub-Class) ซึ่งคลาสย่อย นี้อาจจะมีแอททริบิวต์และเมทธอดเพิ่มเติมจากคลาสหลักก็ได้ ตัวอย่างการกำหนดคลาสหลักและคลาสย่อยแสดงดังรูปที่ 2.3 แสดงให้เห็นการจัดกลุ่มยานพาหนะเป็นคลาสหลักที่มีคลาสย่อยประกอบด้วย คลาสยานพาหนะทางบก คลาสยานพาหนะทางอากาศ และคลาสยานพาหนะทางน้ำ ขณะที่ในส่วนของคลาสย่อยยานพาหนะทางบกนั้น ยังสามารถกำหนดคลาสที่ย่อยกว่าลงไปได้อีกเป็น คลาสรถบรรทุก คลาสรถยนต์ และคลาสรถแทร็คเตอร์



รูปที่ 2.3 แสดงการกำหนดโครงสร้างคลาสหลักและคลาสย่อย

2.1.5 การทำโพลิมอฟีซึม (Polymorphism)

คำว่าโพลิมอฟีซึม แปลว่ามีหลายฟอร์ม ซึ่งหมายถึงการที่เราสามารถใช้คำสั่งเดียวกันกับหลายๆ ออบเจ็กต์ ที่ทำงานแตกต่างกัน เพื่อให้ออบเจ็กต์เหล่านั้น แสดงพฤติกรรมที่ตอบสนองต่อคำสั่งเดียวกันนี้ออกมา โดยผลลัพธ์ที่ได้จากแต่ละออบเจ็กต์อาจแตกต่างกันไป เช่น เมื่อเราสั่งให้โปรแกรมประมวลผลคำพิมพ์เอกสารออกทางเครื่องพิมพ์ โดยที่ในเอกสารนั้นอาจประกอบด้วย ส่วนที่เป็น ข้อความและส่วนที่เป็นรูปภาพ ซึ่งเมื่อเราสั่งพิมพ์เอกสารออบเจ็กต์รูปภาพก็จะพิมพ์ข้อมูลที่เป็นรูปภาพด้วยวิธีการของออบเจ็กต์รูปภาพ ออบเจ็กต์ข้อความก็จะพิมพ์ข้อมูลที่เป็นข้อความด้วยวิธีการของออบเจ็กต์ข้อความ รวมกันเป็นเอกสารฉบับนั้น

2.1.6 การส่งข้อความ

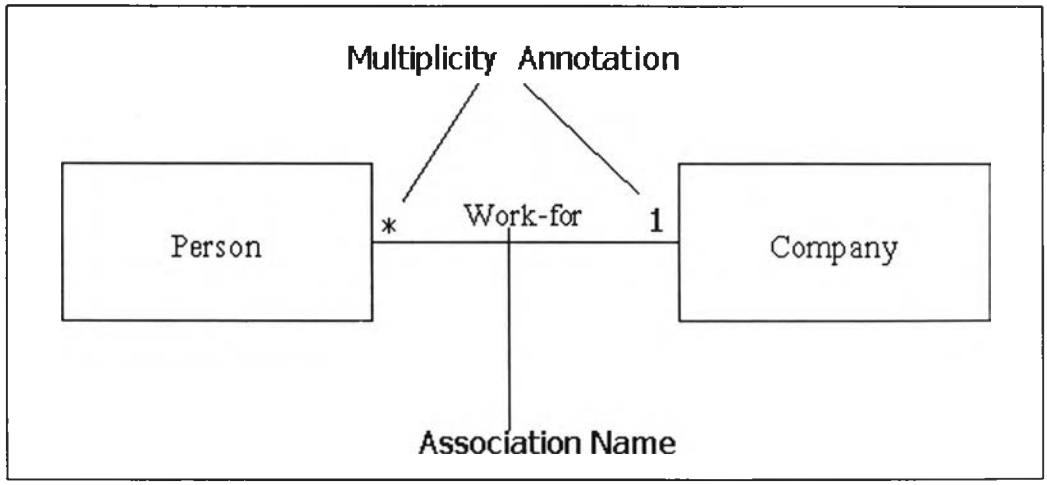
ออบเจ็กต์หนึ่งสามารถเรียกใช้บริการจากอีกออบเจ็กต์ได้โดยการส่งข้อความเรียกใช้บริการไปให้ โดยผ่านส่วนติดต่อภายนอกของออบเจ็กต์นั้น เราเรียกวิธีการส่งคำขอใช้บริการแบบนี้ว่า “การส่งข้อความ”

2.1.7 ความสัมพันธ์ระหว่างออบเจ็กต์

ความสัมพันธ์ระหว่างออบเจ็กต์ มี 4 แบบคือ

1) แบบแอสโซซิเอชัน (Association)

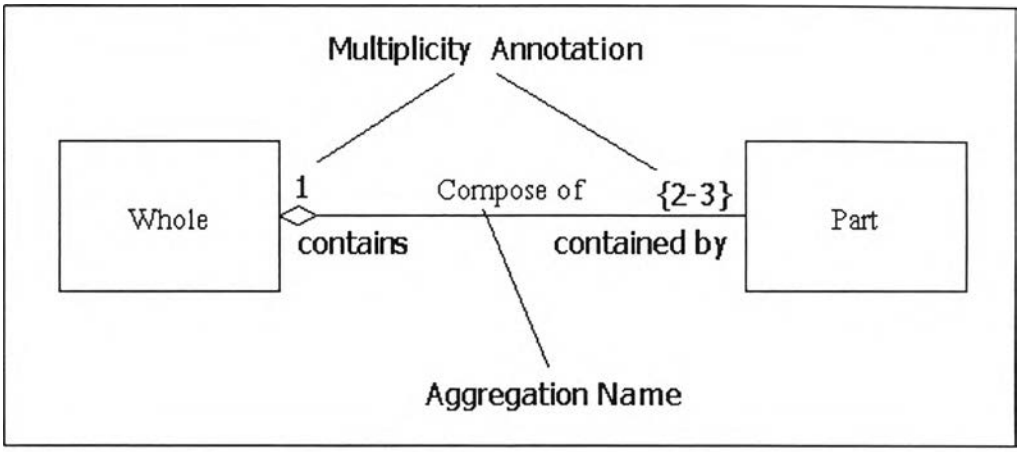
เป็นการอธิบายถึงความสัมพันธ์ระหว่างโครงสร้างของคลาส โดยมีการระบุเงื่อนไขของความสัมพันธ์ทั้งสองด้านและมีการระบุชื่อความสัมพันธ์อย่างชัดเจน ดังรูปที่ 2.4



รูปที่ 2.4 แสดงความสัมพันธ์แบบแอสโซซิเอชัน

2) แบบแอกกริเกชัน (Aggregation)

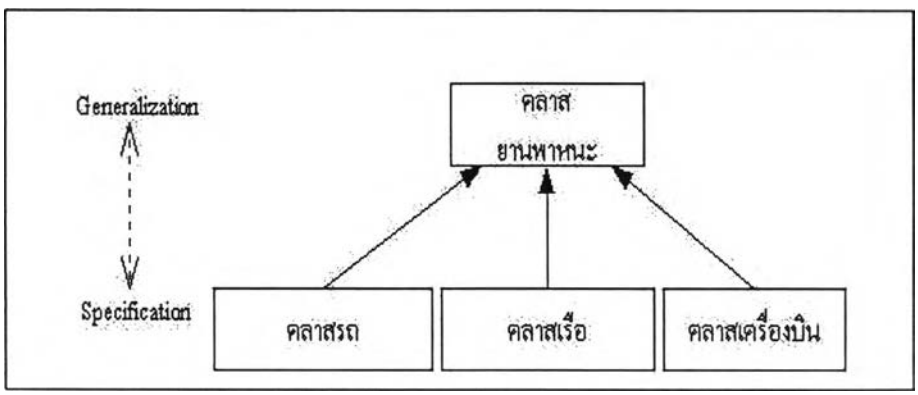
เป็นรูปแบบแสดงความสัมพันธ์อย่างหนึ่งของแบบแอสโซซิเอชัน ซึ่งกำหนดให้มีส่วนที่เป็นโฮล (Whole) และ พารท (Part) ซึ่งจะได้ความสัมพันธ์แบบโฮล-พารท (Whole-Part Relationship) ดังรูปที่ 2.5



รูปที่ 2.5 แสดงความสัมพันธ์แบบแอกกริเกชัน

3) แบบเจนเนอร์รัลไรเซชัน (Generalization)

เป็นความสัมพันธ์ที่เกิดจากกระบวนการ การจัดกลุ่มสิ่งที่เหมือนกันระหว่างคลาส โดยพิจารณาจากโครงสร้างของข้อมูลและพฤติกรรมที่เหมือนๆ กันในกลุ่มของคลาสเหล่านั้น แล้วนำ มาสร้างเป็น คลาสใหม่ ตัวอย่างเช่น ถ้าเรามี คลาสรถ คลาสเรือ และคลาสเครื่องบิน อยู่เราจะสามารถสร้าง คลาสยานพาหนะที่เป็นคลาสที่มีคุณสมบัติร่วมของทั้ง 3 คลาสดังกล่าวขึ้นมาใหม่ได้ ดังรูปที่ 2.6



รูปที่ 2.6 แสดงการเจนเนอร์รัลไรเซชันของคลาส

4) แบบขึ้นต่อกัน (Depends On)

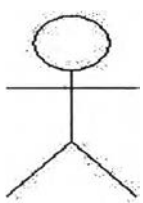
เป็นความสัมพันธ์ที่แสดงให้เห็นว่า ออบเจ็กต์คลาสหนึ่งจะทำปฏิบัติการใด ๆ อย่างไรจะขึ้นอยู่กับค่าของอีกออบเจ็กต์คลาสหนึ่งด้วย โดยเวลาทำงานจะมีการอ้างถึงอีกออบเจ็กต์คลาสด้วยการส่งข้อความไปขอใช้บริการ จากออบเจ็กต์คลาสปลายทางพร้อมส่งและรับ อากิวเมนท์(Argument) หรือ พารามิเตอร์(Parameter) กัน

2.2 ภาษายูเอ็มแอล (UML : Unified Modeling Language)

ยูเอ็มแอล (GCC,1998) เป็นภาษาสำหรับสร้างรูปแบบจำลองของซอฟต์แวร์ ถูกพัฒนาขึ้นในเดือนตุลาคม ค.ศ. 1994 โดยนักพัฒนาของบริษัทเรชันแนลซอฟต์แวร์คอร์ปอเรชัน เพื่อใช้ศึกษารูปแบบวิธีการในเชิงวัตถุ จนกระทั้งปัจจุบัน ยูเอ็มแอล ได้รับการรับรองจาก โอเอ็มจี (OMG : Object Management Group) ให้เป็นมาตรฐานหนึ่งในการพัฒนาระบบงานเชิงวัตถุ และมีหลายองค์กรผลิตเครื่องมือที่สนับสนุนมาตรฐานยูเอ็มแอลออกมา แผนภาพและสัญลักษณ์ที่ใช้ในภาษายูเอ็มแอลมีดังนี้

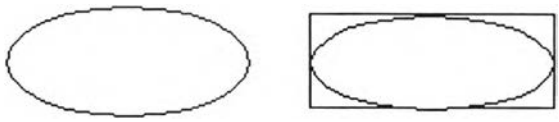
1) แผนภาพยูสเคส (Use Case Diagram) เป็นแผนภาพที่ใช้ในการแสดงความต้องการของระบบโดยรวม สามารถเข้าใจได้ง่าย มีสัญลักษณ์ต่างๆ ดังต่อไปนี้

-แอคเตอร์ (Actor) ใช้สัญลักษณ์รูปคน ดังรูปที่ 2.7 ใช้แทนสิ่งที่อยู่ภายนอกระบบที่จะพัฒนาทั้งหมด ที่ต้องการแลกเปลี่ยนข้อมูลกับระบบ โดยที่แอคเตอร์ อาจจะเป็น คน ระบบ หรือโปรแกรมอื่นๆ ก็ได้ โดยมากจะเป็นผู้เริ่มทำงานกับยูสเคส เช่นผู้ใช้งาน เป็นต้น



รูปที่ 2.7 แสดงสัญลักษณ์ของแอคเตอร์

-ยูสเคส (Use Case) ใช้สัญลักษณ์รูปวงรีหรือวงกลม ดังรูปที่ 2.8ใช้แทนงานที่เกิดขึ้นในแต่ละขั้นตอน ถ้ามีกรอบสี่เหลี่ยมล้อมรอบแสดงว่ามีแผนภาพย่อยที่ใช้อธิบายรายละเอียดของยูสเคสอยู่อีก



รูปที่ 2.8 แสดงสัญลักษณ์ของยูสเคส

-ระบบใช้สัญลักษณ์รูปสี่เหลี่ยม ดังรูปที่ 2.9 ใช้แทนระบบที่ถูกกระทำโดยแอคเตอร์ โดยที่ภายในของระบบจะประกอบด้วยยูสเคสต่างๆ

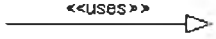
รูปที่ 2.9 แสดงสัญลักษณ์ของระบบ

-การติดต่อสื่อสาร ใช้สัญลักษณ์เส้นที่มีหัวลูกศรทึบหรือหัวก้างปลา ดังรูปที่ 2.10ใช้แสดงความสัมพันธ์หรือการสื่อสารกัน (การรับและการให้ข้อมูลข่าวสารแก่กัน) ระหว่างแอคเตอร์และยูสเคส ซึ่งอาจเป็นการสื่อสารแบบทางเดียวหรือสองทางก็ได้

รูปที่ 2.10 แสดงสัญลักษณ์ของการติดต่อสื่อสาร

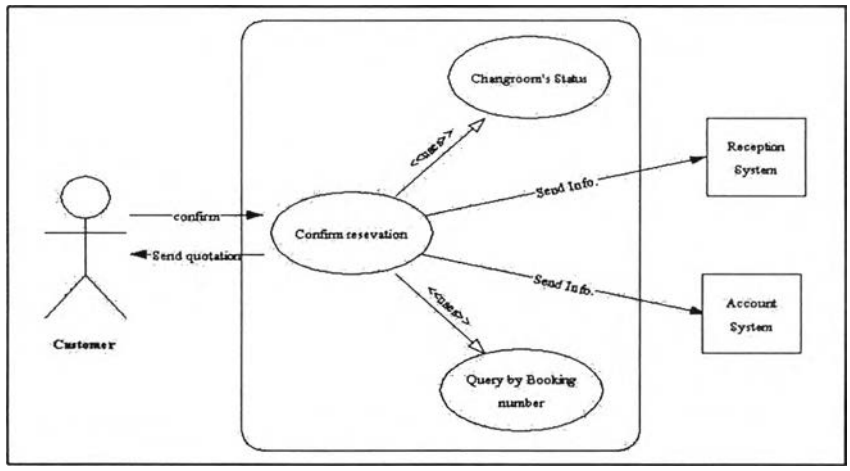
-แสดงความสัมพันธ์ ใช้สัญลักษณ์เส้นที่มีหัวลูกศรโปร่ง ใช้แสดงความสัมพันธ์ระหว่าง ยูสเคสกับยูสเคสด้วยกัน แบ่งออกได้เป็น 2 แบบ คือ

(ก) เพิ่มเติม (Extends) เป็นการเพิ่มความสามารถในการทำงานให้กับยูสเคสโดย การเรียกใช้ความสามารถจากอีกยูสเคสหนึ่ง การเขียนสัญลักษณ์เพื่อแสดงความสัมพันธ์แบบนี้ สามารถเขียนโดยให้ลูกศรชี้ออกจากยูสเคสที่ถูกเรียกใช้งาน ชี้ไปยังยูสเคสที่เป็นตัวเรียกงานที่ต้องการเพิ่มขีดความสามารถ และเขียนข้อความว่า <<extends>> กำกับอยู่ข้างๆ เส้นแสดงความสัมพันธ์นั้น



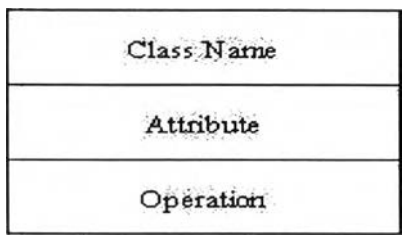
(ข) ใช้ (Uses) เป็นการแสดงการถ่ายทอดหน้าที่การทำงานจากยูสเคสหนึ่งไปยังอีกยูสเคสหนึ่ง การเขียนสัญลักษณ์เพื่อแสดงความสัมพันธ์แบบนี้สามารถเขียนโดย ให้ลูกศรชี้เข้าหายูสเคสที่เป็นเจ้าของความสามารถนั้น (ผู้ถ่ายทอด) และเขียนข้อความว่า <<uses>> กำกับอยู่ข้างๆ เส้นแสดงความสัมพันธ์นั้น

ตัวอย่างแผนภาพยูสเคสแสดงได้ดังรูปที่ 2.11



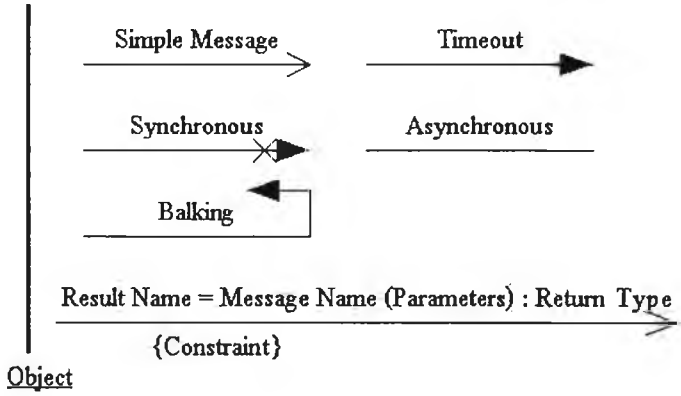
รูปที่ 2.11 แผนภาพยูสเคสแสดงวิธีการจองห้องพักโรงแรม(CDG,1998)

2) แผนภาพคลาส (Class Diagram) เป็นแผนภาพที่ใช้อธิบายโครงสร้างของคลาสและความสัมพันธ์ระหว่างคลาสในระบบ โดยใช้สัญลักษณ์รูปสี่เหลี่ยมแทนคลาส และรูปสี่เหลี่ยมนี้จะถูกแบ่งตามแนวนอนเป็น 3 ส่วน ส่วนที่อยู่บนสุดสำหรับระบุชื่อของคลาสส่วนกลางเอาไว้กำหนดคุณลักษณะของคลาส และส่วนล่างสุดสำหรับกำหนดหน้าที่การทำงานของคลาส ดังรูป 2.12



รูปที่ 2.12 ส่วนประกอบของสัญลักษณ์ของคลาส

3) แผนภาพซีเควน (Sequence Diagram) เป็นแผนภาพอินเตอร์แอ็กชั่นที่ใช้แสดงลำดับขั้นตอนการทำงาน ของออบเจ็กต์ที่เกิดจากการแลกเปลี่ยนข้อความซึ่งกันและกัน หรือเมื่อมีเหตุการณ์อย่างหนึ่งอย่างใดเกิดขึ้น โดยใช้สัญลักษณ์ต่างๆ ดังรูปที่ 2.13 ดังนี้



รูปที่ 2.13 แสดงสัญลักษณ์ที่ใช้ในแผนภาพซีเควน

จากรูปที่ 2.13

Object : คืออินสแตนซ์ของคลาสหรือออบเจ็กต์ที่ทำหน้าที่รับ-ส่งข้อความจากออบเจ็กต์อื่นๆ และทำการตอบสนองต่อข้อความที่รับเหล่านั้น เพื่อให้เกิดการทำงานในขั้นตอนต่างๆ ของระบบ โดยใช้สัญลักษณ์เส้นตรงแนวตั้งที่ระบุชื่อออบเจ็กต์หรือชื่อคลาสด้านล่าง หรือด้านบน

Message : คือข้อความที่ส่งไปมา ระหว่างเส้นแสดง Object ซึ่ง Message มีด้วยกันหลายชนิด ดังต่อไปนี้

-Simple Message : แสดงการส่งข้อความตามปกติ

-Synchronous Message : แสดงการส่งข้อความและรอจนกว่าจะได้รับคำตอบกลับคืน

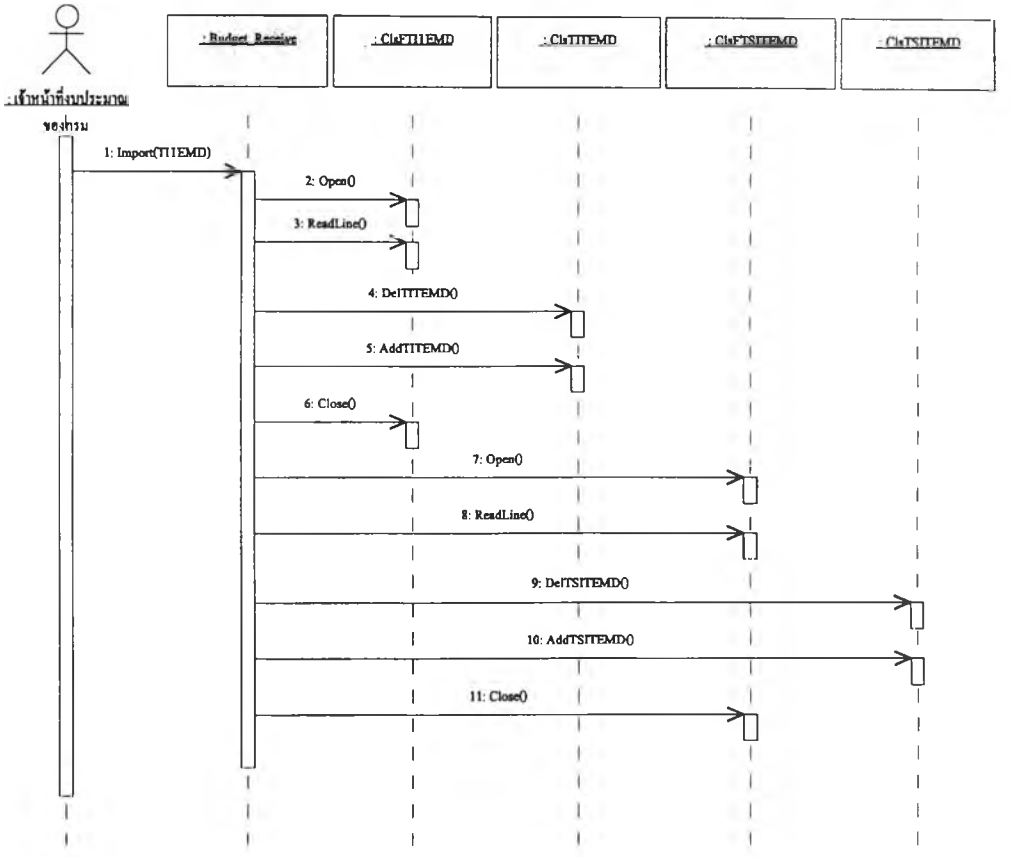
-Balking Meaasage : แสดงการส่งข้อความ ที่ต้องการ ให้ทางฝ่ายผู้รับดำเนินการทันที หากทางผู้รับไม่ดำเนินการทันทีจะถือว่าไม่มีการส่งข้อความ

-Timeout Message : แสดง การส่งข้อความ ที่ต้องการ ให้ทางฝ่ายผู้รับดำเนินการภายในระยะเวลาที่กำหนด หากทางผู้รับไม่ดำเนินการตามกำหนดเวลาจะถือว่าไม่มีการส่งข้อความ

-Asynchronous Message : แสดงการส่งข้อความ และดำเนินการอย่างอื่นต่อไปโดยไม่รอคำตอบ

ตัวอย่างแผนภาพซีเควนที่ใช้ในงานวิจัยนี้ แสดงดังรูปที่ 2.14

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.14 ตัวอย่างแผนภาพซีเควน

2.3 วงจรการพัฒนาระบบแบบเรชันแนลออบเจ็กต์ทอรีโพรเซส (Rational Objectory Process Software Development Life Cycle)

เป็นการพัฒนาซอฟต์แวร์ที่ใช้แนวคิดของการวนซ้ำ (Iteration) หลายรอบ และการพัฒนาแบบเพิ่มพูน (Incremental) โดยทำการแยกซอฟต์แวร์ออกเป็นส่วน ๆ แล้วทำการพัฒนารอบละส่วนจนครบจำนวนรอบเท่ากับซอฟต์แวร์ที่แบ่งไว้ จะประกอบไปด้วย 4 ระยะดังนี้

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

2.3.1 ระยะอินเซ็ปชัน

เป็นขั้นตอนในการกำหนดขอบเขตของปัญหา โครงสร้างของผลิตภัณฑ์ ตลอดทั้งหน้าที่ และความสามารถของผลิตภัณฑ์ เพื่อแบ่งส่วนของซอฟต์แวร์ออกเป็นส่วน ๆ ศึกษาข้อจำกัดต่าง ๆ ของระบบตลอดทั้งศึกษาแนวทางการนำเทคโนโลยีต่าง ๆ มาใช้งาน ศึกษาถึงความสัมพันธ์ที่มีต่อระบบอื่น ๆ ตลอดทั้งระบุข้อมูลที่จะนำเข้าและผลลัพธ์ที่จะได้จากระบบ ในขั้นตอนนี้จะใช้แบบจำลองยูสเคส (Use case Model) เป็นเครื่องมือสำหรับกำหนดขอบเขตของปัญหาเพื่อให้ผู้พัฒนาและผู้ที่เกี่ยวข้องเข้าใจได้ง่ายสะดวกในการพัฒนาระบบ

2.3.2 ระยะอีแลบบอเรชัน

เป็นระยะที่จะทำการออกแบบสถาปัตยกรรมเบื้องต้นของซอฟต์แวร์ โดยศึกษาและวิเคราะห์จากขอบเขตของปัญหาและโครงสร้างของซอฟต์แวร์เบื้องต้นที่ได้ทำการศึกษาไว้จากระยะอินเซ็ปชัน นำมาประเมินความเสี่ยงในด้านต่าง ๆ เช่น ทางด้านเทคนิค ด้านงบประมาณ เป็นต้น แล้วนำมากำหนดเป็นแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันเพื่อช่วยในการประเมินความเสี่ยงต่าง ๆ ดังกล่าว โดยแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันจะต้องสัมพันธ์กับยูสเคสที่ได้ทำการศึกษาไว้จากระยะอินเซ็ปชัน

2.3.3 ระยะคอนสตักชัน

เป็นระยะที่จะพัฒนาซอฟต์แวร์ต่อจากระยะอีแลบบอเรชัน เพื่อพัฒนาต้นแบบของผลิตภัณฑ์ โดยใช้แนวคิดการวนซ้ำและการพัฒนาแบบเพิ่มพูน แบ่งซอฟต์แวร์ออกเป็นส่วน ๆ แล้วแยกพัฒนาทีละส่วน ซึ่งในระยะนี้จะแบ่งออกเป็น 5 ขั้นตอนดังนี้

- วางแผนเพื่อแจกแจงรายละเอียดของผลิตภัณฑ์ ได้แก่ ความสามารถหน้าที่เพื่อให้ผู้ร่วมทีมพัฒนาเข้าใจ
- การวิเคราะห์ทำการศึกษารายละเอียดโดยใช้แผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันที่ได้จากระยะที่ 2 เพื่อนำมาวิเคราะห์หาแผนภาพคลาสและแผนภาพอินเตอร์แอ็กชันที่เหมาะสมกับระบบย่อยส่วนที่ 1
- การออกแบบเป็นระยะที่นำแผนภาพต่าง ๆ ที่ได้จากการวิเคราะห์ในขั้นตอนที่ 2 มาออกแบบส่วนที่ติดต่อกับผู้ใช้ รวมทั้งการออกแบบโครงร่างฐานข้อมูลเชิงตรรกะ (Logical Database Schema)
- การพัฒนาต้นแบบโปรแกรมโดยทำการแปลงแผนภาพต่าง ๆ ที่ได้จากการศึกษาดังกล่าวข้างต้นออกมาเป็นโปรแกรม รวมทั้งทำการแปลงโครงร่างฐานข้อมูลเชิงตรรกะเป็นโครงร่างฐานข้อมูลเชิงกายภาพ (Physical Database Schema)

- การทดสอบโปรแกรมที่ได้จากการพัฒนา เป็นการค้นหาข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นจากโปรแกรมที่พัฒนาเพื่อนำมาปรับปรุงแก้ไขให้โปรแกรมทำงานอย่างมีประสิทธิภาพ ประกอบไปด้วย การทดสอบหน่วยของโปรแกรม การทดสอบ การเชื่อมต่อของโปรแกรม การตรวจสอบก่อนการรับมอบโปรแกรม

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

2.3.4 ระยะทรานสิชัน

เป็นระยะที่ทำการปรับปรุงต้นแบบที่ได้จากพัฒนาในระยะคอนสตักชัน เพื่อนำมาปรับปรุงแก้ไขข้อบกพร่องต่าง ๆ พร้อมทั้งพัฒนาส่วนประกอบอื่นเพื่อให้ซอฟต์แวร์ทำงานได้อย่างมีประสิทธิภาพ แบบวิธีการติดตั้ง วิธีการบำรุงรักษาวิธีการแก้ไขเมื่อพบข้อบกพร่อง และรวบรวมส่วนประกอบอื่นของซอฟต์แวร์เพื่อบรรจุลงในผลิตภัณฑ์พร้อมที่จะนำไปใช้งาน

2.4 แนวคิดเกี่ยวกับการออกแบบฐานข้อมูล

2.4.1 ฐานข้อมูล

คือ การจัดเก็บรายละเอียดของข้อมูลที่เกี่ยวข้องกันอย่างเป็นระบบ เพื่อให้เกิดประโยชน์ในการใช้งานข้อมูลร่วมกันอย่างมีประสิทธิภาพ ทั้งการเพิ่ม การแก้ไข การลบ และการเรียกดูข้อมูล อย่างเป็นอิสระไม่ขึ้นกับวิธีการออกแบบของระบบงานใดระบบงานหนึ่ง

เหตุที่การจัดเก็บข้อมูลเป็นฐานข้อมูลได้รับความนิยมในปัจจุบัน เพราะ

1) สามารถลดความขัดแย้งของข้อมูลภายในระบบและในองค์กร

2) สามารถใช้ข้อมูลร่วมกันลดความซ้ำซ้อนในการจัดเก็บและบำรุงรักษาข้อมูล

3) สามารถควบคุมและจัดการความถูกต้องเชื่อถือได้ของข้อมูลได้ง่าย

4) สามารถกำหนดมาตรฐานการใช้งานข้อมูลภายในองค์กรได้สะดวก

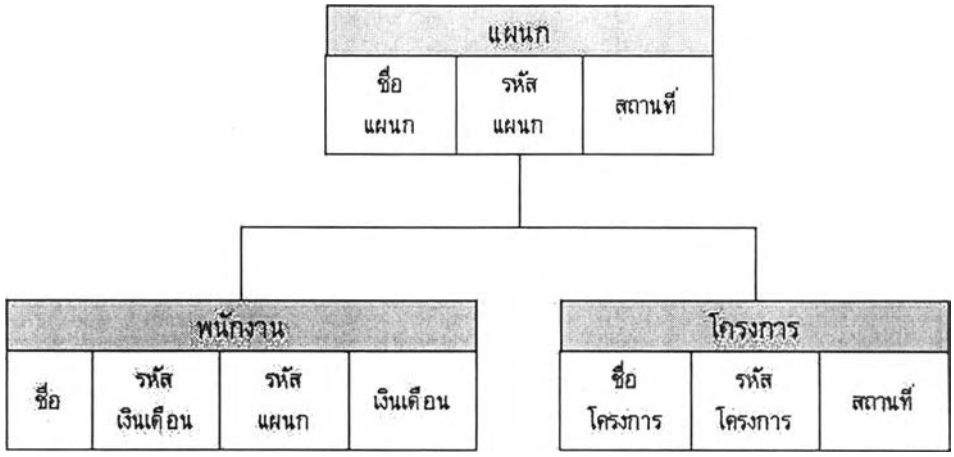
5) กำหนดระบบรักษาความปลอดภัยของข้อมูลได้ง่าย

6) มีความเป็นอิสระไม่ขึ้นต่อกันในการเปลี่ยนแปลงแก้ไขข้อมูลและโปรแกรม

2.4.2 รูปแบบจำลองของฐานข้อมูล (Database Model)

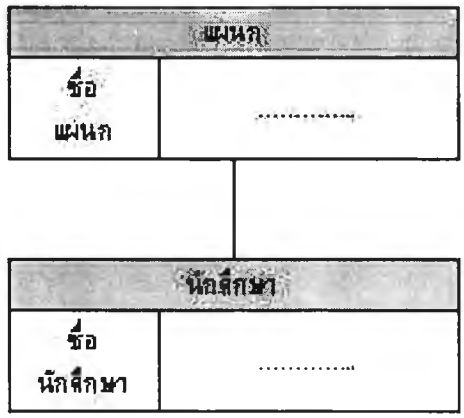
รูปแบบของฐานข้อมูลที่เป็นที่รู้จักกันดีในปัจจุบัน มี 3 รูปแบบคือ

1) ฐานข้อมูลแบบลำดับชั้น (Hierarchical Model) เป็นโครงสร้างที่มีฟิลด์เป็นลำดับชั้นย่อยลงไปโดยจะมีฟิลด์พ่อเป็นฟิลด์ใหญ่ และมีฟิลด์ลูกเป็นฟิลด์เรียงลำดับย่อยเรื่อยไป หรือมีลักษณะความสัมพันธ์เป็นแบบพ่อ-ลูก ดังรูปที่ 2.15



รูปที่ 2.15 ฐานข้อมูลแบบลำดับชั้น

2) ฐานข้อมูลแบบเครือข่าย (Network Model) เป็นโครงสร้างแบบที่มีฟิลด์เชื่อมโยงกันหมด การอ้างอิงความสัมพันธ์ระหว่างกันจะกำหนดเป็น เซ็ทไทพ์ (Set Type) ซึ่งประกอบด้วยชื่อของ เซ็ทไทพ์ ชื่อของประเภทเร็คคอร์ดหลัก (Owner Record Type) ชื่อของประเภทเร็คคอร์ดที่เป็น สมาชิก (member Record Type) ดังรูปที่ 2.16



รูปที่ 2.16 ฐานข้อมูลแบบเครือข่าย

3) ฐานข้อมูลแบบเชิงสัมพันธ์ (Relational Model) มีการจัดเก็บข้อมูลในรูปแบบตาราง 2 มิติ คือ เป็นแถว และเป็นคอลัมน์ ดังรูปที่ 2.17

DEPTNO	DNAM	LOC
10	Sale	Boston
20	Account	New York
30	Production	Chicaco
40	Admin	Texus

EMPNO	EMPNAME	HIREDATE	SALARY	POSITION	DEPTNO	MGRNO
1001	PRASIT	06/15/99	3000	Clerk	10	1002
3001	SOMCHAI	01/01/00	17000	Salesman	30	3004
4001	WICHAI	01/10/99	33000	Manager	40	2002
1002	TANIN	05/31/99	30000	Controller	10	1003

รูปที่ 2.17 ฐานข้อมูลเชิงสัมพันธ์

2.4.3 ฐานข้อมูลเชิงสัมพันธ์

ฐานข้อมูลเชิงสัมพันธ์เป็นรูปแบบฐานข้อมูลที่ได้รับความนิยม นำมาจัดทำเป็นระบบจัดการฐานข้อมูลในปัจจุบันมากที่สุด และในการพัฒนาระบบการจัดทำงบประมาณรายจ่ายนี้ ก็ใช้รูปแบบการออกแบบข้อมูลแบบฐานข้อมูลเชิงสัมพันธ์นี้ด้วย

โดยทั่วไปในระบบฐานข้อมูลหนึ่ง ๆ จะเก็บข้อมูลต่าง ๆ ตามโครงสร้างของข้อมูลที่ระบบงานต้องใช้ทั้งหมด และจะยอมให้ผู้ใช้หลายกลุ่มหลายระดับสามารถเรียกใช้ข้อมูลร่วมกันได้ โดยผ่านระบบงาน หรือภาษาเรียกค้นข้อมูลของระบบจัดการฐานข้อมูลเอง ในทางปฏิบัติจึงจำเป็นต้องมีการแบ่งระดับ
ของข้อมูลตามระดับสิทธิ์การเรียกใช้ข้อมูล เพื่อจัดระเบียนการใช้ข้อมูลให้เหมาะสมต่อผู้ใช้ข้อมูลแต่ละกลุ่ม ดังนี้

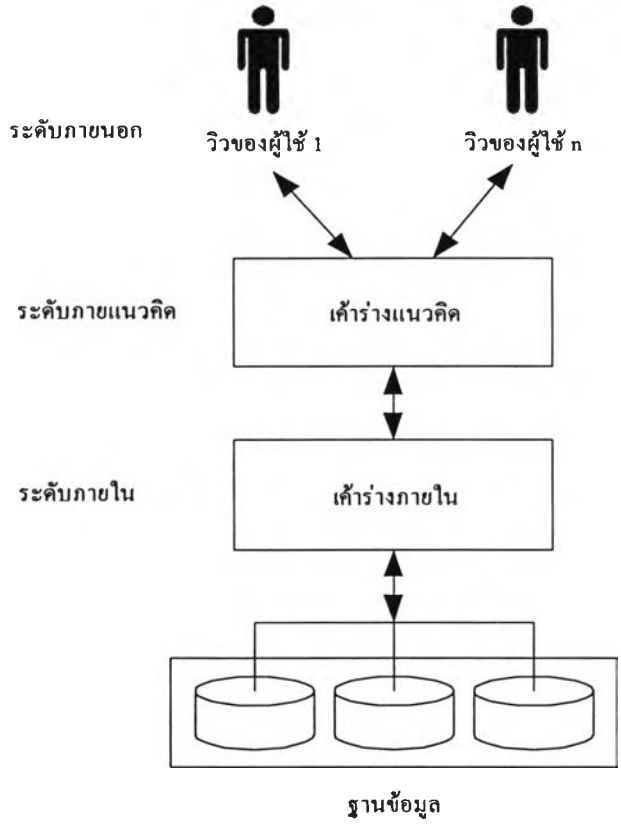
1) ข้อมูลระดับภายนอกหรือวิว (External Schema หรือ View) เป็นระดับของข้อมูลที่กำหนดให้ผู้ใช้แต่ละคน หรือโปรแกรมแต่ละโปรแกรมสามารถมองเห็นหรือใช้งานข้อมูลเฉพาะที่เกี่ยวข้องกับหน้าที่รับผิดชอบของผู้ใช้หรือโปรแกรมนั้นเท่านั้น

2) ข้อมูลระดับแนวคิด (Conceptual Schema) เป็นระดับของข้อมูลที่กำหนดขึ้นจากการออกแบบข้อมูลที่ต้องการใช้งาน ตามที่นักวิเคราะห์ระบบออกแบบไว้ ประกอบด้วยโครงสร้างข้อมูล ความสัมพันธ์ของข้อมูล กฎเกณฑ์และข้อจำกัดต่าง ๆ ข้อมูลระดับนี้ถูกสร้าง

ขึ้นมาเพื่อให้ผู้ใช้ข้อมูลต่าง ๆ ในระดับภายนอกสามารถเรียกใช้ข้อมูลได้หลากหลายรูปแบบตามความจำเป็นและหน้าที่รับผิดชอบ

3) ข้อมูลระดับภายใน (Internal Schema) เป็นระดับการจัดเก็บโครงสร้างของฐานข้อมูลที่เก็บอยู่จริงมีวิธีการจัดเก็บ และเข้าถึงข้อมูลอย่างไรจะขึ้นอยู่กับความสามารถของระบบจัดการฐานข้อมูลที่ใช้

ซึ่งข้อมูลทั้ง 3 ระดับจะมีความสัมพันธ์กันดังรูปที่ 2.18



รูปที่ 2.18 แสดงความสัมพันธ์ของข้อมูลทั้ง 3 ระดับ

2.4.4 องค์ประกอบของฐานข้อมูลเชิงสัมพันธ์

1) รีเลชั่นเป็นคำนามที่แทนข้อมูลเรื่องใดเรื่องหนึ่ง บางครั้งเรียกว่า ตาราง (Table)

2) ทูเพิล (Tuple) เป็นค่าของข้อมูลในแต่ละแถว บางครั้งเรียกว่า เรคคอร์ด

3) แอททริบิวต์ เป็นรายละเอียดของข้อมูลในแต่ละคอลัมน์ ซึ่งเป็นสิ่งแสดงให้รู้ว่ารีเลชั่นนั้น ๆ ประกอบด้วยรายละเอียดอะไรบ้าง บางที่เรียกว่า คอลัมน์ หรือ ฟิลด์

4) คาดิแนลลิตี (Cardinality) เป็นค่าจำนวนแถวของข้อมูลในแต่ละรีเลชั่น

5) คีย์หลัก (Primary Key) เป็นแอททริบิวต์ที่มีค่าของข้อมูลที่เป็นเอกลักษณ์หรือสามารถใช้ชี้เฉพาะเจาะจงถึงทูเพิลนั้น ๆ ได้ (Unique Identifier) บางที่ก็เรียกว่า ยูนีคคีย์

6) โดเมน คือขอบเขตของข้อมูลที่เป็นไปได้ในแต่ละแอทริบิวต์ เช่น โดเมนของแอทริบิวต์เพศจะมีค่าเป็น ชาย หรือ หญิง เท่านั้น เป็นต้น

2.4.5 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลมีขั้นตอนดังนี้

1) ค้นหาเอนทิตี (Entity) ทั้งหมด โดยศึกษาถึงลักษณะหน้าที่ของระบบงาน (Business Function) ว่ามีรายละเอียดการทำงานอย่างไร มีข้อมูลใดเกี่ยวข้องบ้าง มีข้อสมมุติฐาน (Business Rule) ของงานต่าง ๆ อย่างไร โดยพิจารณาดูจากผลการออกแบบระบบงานที่ได้ออกแบบไว้ จะได้ว่ามีเอนทิตีใดบ้างในระบบ (เอนทิตี หมายถึง ชื่อของสิ่งหนึ่งสิ่งใด ทั้งที่จับต้องได้หรือจับต้องไม่ได้ เช่น คน สถานที่ สิ่งของ การลงทะเบียนเรียน เป็นต้น)

2) คัดเลือกเอนทิตีที่ควรให้มีอยู่ในฐานข้อมูล โดยพิจารณาจากคุณสมบัติของเอนทิตีนั้น ๆ ว่า เป็นเอนทิตีประเภทซุปเปอร์ไทพ์ (Supertype) หรือ สับไทพ์ (Subtype) ซึ่ง ซุปเปอร์ไทพ์ และ สับไทพ์ หมายถึงการที่เอนทิตีหนึ่งมีรายละเอียดร่วมเหมือนกับเอนทิตีอื่น และรายละเอียดร่วมถูกกำหนดเป็นเอนทิตีใหม่ เรียกว่า ซุปเปอร์ไทพ์ ส่วนเอนทิตีเดิมจะอ้างอิงถึงเอนทิตีที่เป็น ซุปเปอร์ไทพ์ และเพิ่มรายละเอียดเฉพาะเข้าไป เรียกว่า สับไทพ์ ดังนั้นเมื่อมีเอนทิตีสับไทพ์ จะต้องมี เอนทิตีซุปเปอร์ไทพ์ ด้วยเสมอ

3) กำหนดประเภทของความสัมพันธ์ระหว่างเอนทิตี โดยพิจารณาจากกฎเกณฑ์ ข้อสมมุติฐานต่าง ๆ ของระบบงานที่ได้จากข้อ 1. โดยที่รูปแบบความสัมพันธ์ระหว่างเอนทิตี มีด้วยกัน 3 แบบคือ

- ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One–to-One Relationship (1:1))
- ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One–to-Many Relationship (1:m))
- ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many–to-Many Relationship (m:n))

4) กำหนดคุณลักษณะของเอนทิตี กำหนดว่าเอนทิตีแต่ละเอนทิตี ควรมีรายละเอียดแอทริบิวต์อะไรบ้าง มีคุณลักษณะของแอททริบิวต์อย่างไรซึ่งคุณลักษณะต่าง ๆ ของแอททริบิวต์ โดยทั่วไปจะประกอบด้วย

ชื่อแอททริบิวต์

ประเภทของข้อมูลที่จัดเก็บ

ความยาวของข้อมูลสูงสุด

ยอมให้มีค่าว่าง (Null) ได้หรือไม่

คำอธิบาย

5) กำหนดคีย์หลักของแต่ละเอนทิตี กำหนดว่าแอททริบิวต์ใดบ้างประกอบกันเป็นคีย์หลักของเอนทิตีแต่ละเอนทิตี ซึ่งแอททริบิวต์ ที่ประกอบกันเป็นคีย์หลักจะต้องเป็นแอททริบิวต์ที่เก็บค่าข้อมูลที่เป็นเอกลักษณ์ กล่าวคือจะต้องไม่มีค่าซ้ำกันในเอนทิตีนั้น ๆ

6) นำผลจากข้อ 1 ถึง 5 มาทบทวน และเขียนเป็น แบบจำลอง อี อาร์ (E-R Model) และแปลงให้อยู่ในรูปแบบฐานข้อมูลเชิงสัมพันธ์ ดังขั้นตอนต่อไปนี้

(1) แปลงเอนทิตีต่าง ๆ ในแบบจำลอง อี อาร์ ให้เป็นรีเลชั่น และแปลงประเภทความสัมพันธ์ของเอนทิตีเป็นความสัมพันธ์ของรีเลชั่นด้วย ในกรณีที่ความสัมพันธ์ระหว่างเอนทิตีเป็นแบบกลุ่มต่อกลุ่ม มักนิยมสร้างรีเลชั่นใหม่ขึ้นมาแทรกกลางและทำให้ความสัมพันธ์ระหว่างเอนทิตีใหม่ที่สร้างนี้กับเอนทิตีเดิมทั้งสองเป็นแบบหนึ่งต่อกลุ่ม (One-to-Many)

(2) พิจารณาคีย์หลักและคีย์นอก(Foreign Key)ของแต่ละรีเลชั่น ว่าต้องมีคีย์หลักและคีย์นอกอะไรบ้าง ซึ่งคีย์นอก จะหมายถึงแอททริบิวต์ในรีเลชั่นหนึ่งที่ใช้ในการอ้างอิงถึงแอททริบิวต์เดียวกันที่ทำหน้าที่เป็นคีย์หลักของอีกรีเลชั่นหนึ่ง เพื่อใช้ในการเชื่อมโยงข้อมูลซึ่งกันและกัน

(3) แปลงรายละเอียดของเอนทิตี ให้เป็นแอททริบิวต์ของรีเลชั่น ในกรณีที่แอททริบิวต์หนึ่งมีค่าหลายค่า (Multivalued Attribute) จะต้องแยกแอททริบิวต์นั้นเป็นรีเลชั่นใหม่

(4) ทำให้รีเลชั่นอยู่ในรูปแบบบรรทัดฐาน (Normalization) ซึ่งเป็นการทำให้รีเลชั่นอยู่ในรูปแบบที่เป็นบรรทัดฐาน (Normal Form) เพื่อ

(4.1) ลดเนื้อที่ในการจัดเก็บข้อมูล ลดการเก็บข้อมูลที่ซ้ำกัน

(4.2) ลดปัญหา ความไม่ถูกต้องตรงกัน ของข้อมูล (Data Inconsistncy) ที่เกิดจากการเก็บข้อมูลเดียวกับไว้หลายรีเลชั่น ทำให้มีความซ้ำซ้อนในการเก็บข้อมูล (Redundancy)

(4.3) ลดปัญหาที่เกิดจากการเพิ่ม ปรับปรุง และลบข้อมูล (Insert Update and Delete Anomalies) ช่วยแก้ปัญหาที่อาจเกิดจากการแก้ไข ปรับปรุงข้อมูลไม่ครบถ้วน

รูปแบบบรรทัดฐาน แบ่งออกได้ดังนี้ คือ

(ก.) รูปแบบบรรทัดฐานขั้นที่ 1 (First Normal Form : 1NF) รีเลชั่นหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 ก็ต่อเมื่อ "ค่าของแอททริบิวต์ใด ๆ ของแต่ละทูเพิลเก็บค่าได้เพียงค่าเดียว"

(ข.) รูปแบบบรรทัดฐานขั้นที่ 2 (Second Normal Form : 2NF) รีเลชั่นหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 2 ก็ต่อเมื่อ "รีเลชั่นนั้น ๆ อยู่ในรูปแบบบรรทัด

ฐานขั้นที่ 1 และมีคุณสมบัติอีกประการหนึ่งคือ แอททริบิวต์ทุกแอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะต้องมีความสัมพันธ์ระหว่างค่าของแอททริบิวต์แบบฟังก์ชั่นกับคีย์หลัก (Fully Function Dependency) กล่าวอีกนัยหนึ่งคือค่าของแอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะสามารถระบุค่า โดยแอททริบิวต์ที่เป็นคีย์หลัก หรือแอททริบิวต์ทั้งหมดที่ประกอบกันเป็นคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม(Composite Key)" (Prime FFD to Non_prime)

(ค.) รูปแบบบรรทัดฐานบอยส์และคอดด์ (Boyce/Codd Normal Form : BCNF) รีเลชั่นหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานบอยส์และคอดด์ ก็ต่อเมื่อ "รีเลชั่นนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 3 และไม่มีแอททริบิวต์อื่นในรีเรชั่นที่สามารถระบุค่าของแอททริบิวต์ที่เป็นคีย์หลักหรือส่วนหนึ่งส่วนใดของคีย์หลักในกรณีที่คีย์หลักเป็นคีย์ผสม" (No Cadidate Key FFD to Prime)

(ง.) รูปแบบบรรทัดฐานขั้นที่ 4 (Fourth Normal Form : 4NF) รีเลชั่นหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 4 ก็ต่อเมื่อ "รีเลชั่นนั้น ๆ อยู่ในรูปแบบ BCNF และเป็นรีเลชั่นที่ไม่มีความสัมพันธ์ในการระบุค่าของแอททริบิวต์แบบหลายค่า โดยที่แอททริบิวต์ที่ถูกระบุค่าเหล่านี้ ไม่มีความสัมพันธ์กัน (Independently Multivalued Dependency)" (No MVD)

(จ.) รูปแบบบรรทัดฐานขั้นที่ 5 (Fifth Normal Form : 5NF) รีเลชั่นหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 5 ก็ต่อเมื่อ "รีเลชั่นนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 4 และเป็นรีเลชั่นที่มีคีย์หลักเป็นคีย์ผสมที่ประกอบด้วยแอททริบิวต์ตั้งแต่ 3 แอททริบิวต์เป็นต้นไป หากมีการแตกรีเลชั่น ออกเป็นรีเลชั่นย่อย สามรีเลชั่น(หรือมากกว่า) ซึ่งเกิดจากการจับคู่แอททริบิวต์แต่ละคู่ของรีเลชั่นเดิมเป็นคีย์ผสม และเมื่อทำการเชื่อมโยงรีเลชั่นย่อยทั้งหมดจะไม่ก่อให้เกิดข้อมูลใหม่ที่ไม่เหมือนรีเลชั่นเดิม"

2.5 สถาปัตยกรรมของเทคโนโลยีเว็บ

2.5.1 อินเตอร์เนต (Internet)

อินเตอร์เนตในวันนี้ คือเครือข่ายคอมพิวเตอร์สาธารณะที่มีขนาดใหญ่ และสำคัญที่สุดของโลก เป็นบริการเชื่อมจอคอมพิวเตอร์/เครือข่ายคอมพิวเตอร์ด้วยโปรโตคอล ทีซีพี/ไอพี (TCP/IP) โดยคอมพิวเตอร์ที่เข้ามาเป็นส่วนหนึ่งของอินเตอร์เนตจะต้องมี ไอพีแอดเดรส (IP Address) ไว้เป็นสิ่งอ้างอิงเมื่อเราจะติดต่อกับคอมพิวเตอร์เครื่องนั้นๆ

2.5.2 บริการของอินเตอร์เนต

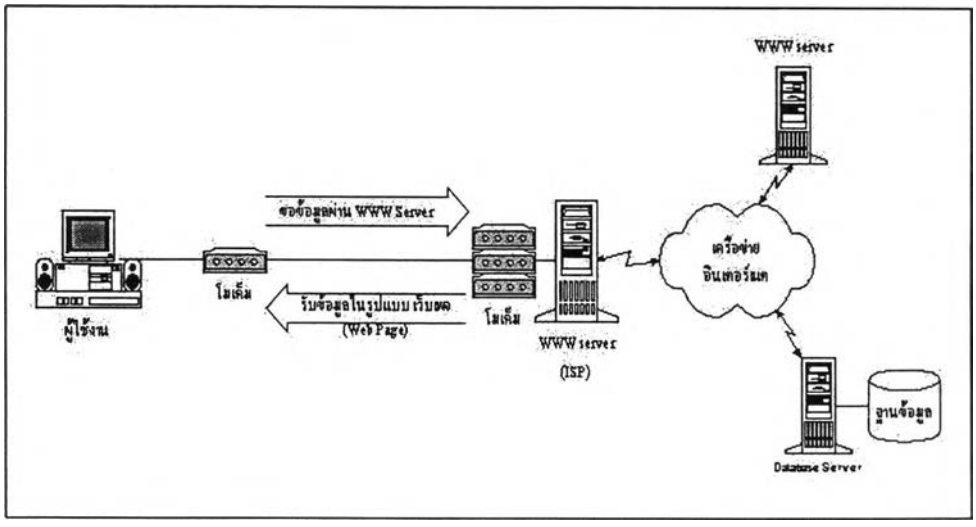
ในอินเตอร์เนต มีบริการหลากหลายรูปแบบให้ผู้ใช้เลือกใช้งานตามความเหมาะสมและความจำเป็น โดยมีบริการที่สำคัญๆ ได้แก่

- อีเมล์ (E-MAIL : Electronic Mail) เป็นบริการส่งไปรษณีย์อิเล็กทรอนิกส์
- เอฟทีพี (FTP : File Transfer Protocol) ใช้ในการส่ง/รับไฟล์ผ่านอินเตอร์เนต
- ไออาร์ซี (IRC : Internet Reply Chat) เป็นการสนทนาผ่านอินเตอร์เนตกับกลุ่มที่สนใจในเรื่องเดียวกัน
- นิวส์กรุ๊ป (NewsGroup) กลุ่มข่าวของผู้ที่มีความสนใจในเรื่องเดียวกัน
- โกเฟอร์ (Gopher) เป็นบริการค้นหาไฟล์ที่เก็บไว้ โดยค้นหาจากเมนูของหัวข้อต่างๆ ที่ได้จัดกลุ่มไว้
- เวิลด์ไวด์เว็บ (World Wide Web หรือ WWW) เป็นระบบที่เชื่อมต่อข่อมูลในคอมพิวเตอร์เครื่องต่างๆ ของอินเตอร์เนตไว้ด้วยดัน

บริการที่ได้รับความนิยมและมีผู้ใช้งานมากที่สุดคือ เวิลด์ไวด์เว็บ หรือที่มักเรียกกันว่า เว็บ ซึ่งในงานวิจัยนี้ก็ใช้แนวทางนี้ในการพัฒนาระบบงานด้วยเช่นกัน

2.5.3 เวิลด์ไวด์เว็บ

เป็นระบบการสื่อสารสารสนเทศที่เป็นไฮเปอร์เท็กซ์(Hypertext) ที่ได้รับความนิยมสูงสุดในการใช้งานอินเตอร์เนตในปัจจุบัน โดยมีการสื่อสารข้อมูลในรูปแบบไคลเอนต์/เซิร์ฟเวอร์ ซึ่งมีองค์ประกอบที่สำคัญดังรูปที่ 2.19 ดังต่อไปนี้



รูปที่ 2.19 องค์ประกอบของ เวิลด์ไวด์เว็บ

1) เว็บบราวเซอร์

เป็นแอปพลิเคชันที่นำผู้ใช้งานอินเตอร์เนตไปสู่แหล่งข้อมูลต่างๆ โดยบราวเซอร์จะทำหน้าที่แสดงเอกสารตามที่ผู้ใช้ต้องการ นอกจากนี้ยังเพิ่มความสามารถในการบันทึกชื่อและแหล่งข้อมูลที่เคยค้นหามาก่อนหน้านี้ หรือแนะนำแหล่งข้อมูลที่น่าสนใจให้กับผู้ใช้งาน บราวเซอร์มีให้เลือกใช้มากมาย ซึ่งส่วนใหญ่จะแจกจ่ายให้โดยไม่เสียค่าใช้ เช่น อินเตอร์เนตเอกซ์โพเร่อร์ (Internet Exploror), เนสเคปเนวิเกเตอร์ (Netscape Navigator) เป็นต้น บราวเซอร์ในปัจจุบันมีความสามรถที่จะเข้าใจข้อมูลทั้งที่เป็นข้อความ ภาพนิ่ง (Image) ภาพเคลื่อนไหว (Video) หรือเสียงได้ดี และยังมีความสามารถพิเศษเพิ่มเติมอีกมากมายเพื่อดึงดูดผู้ใช้ให้เลือกใช้บราวเซอร์นั้นๆ

2) เว็บเซิร์ฟเวอร์

เป็นแอปพลิเคชันที่ทำหน้าที่รับและประมวลผลเอกสาร ที่ถูกร้องขอจากผู้ใช้บริการอินเตอร์เนตผ่านบราวเซอร์ ซึ่งอาจเป็นการร้องขอเพื่อให้แสดงเอกสาร ข้อมูลจากฐานข้อมูล หรือทำการคำนวณ ซึ่งเว็บเซิร์ฟเวอร์จะส่งเอกสารกลับไปแสดงผลให้ผู้ใช้บริการผ่านบราวเซอร์ที่ร้องขอนั้น ในรูปเว็บเพจ นอกจากเว็บเซิร์ฟเวอร์จะถูกนำมาใช้เพื่อให้บริการในอินเตอร์เนตดังกล่าวแล้ว ยังอาจนำไปประยุกต์ใช้กับเครือข่ายภายในองค์กรที่เรียกว่า "อินทราเนต" ได้เช่นกัน

3) ไฮเปอร์ลิงก์

เป็นวิธีการเชื่อมโยงจากแหล่งข้อมูลหนึ่งไปยังอีกแหล่งข้อมูลหนึ่ง ซึ่งมีข้อมูลที่เกี่ยวข้องกัน ไฮเปอร์ลิงก์จะอยู่เป็นส่วนหนึ่งภายในเอกสารเอ็ชทีเอ็มแอล(HTML) เพื่อให้ผู้ใช้บริการสามารถเชื่อมโยงไปยังเอกสารอื่นที่มีความสัมพันธ์กันได้

4) เอ็ชทีเอ็มแอล

ย่อมาจาก HyperText Markup Language เป็นภาษาที่ใช้ในการสร้างเอกสารชนิดพิเศษ ซึ่งมีความสามารถในการเชื่อมโยงกับเอกสารหรือแหล่งข้อมูลอื่นๆ ที่มีความสัมพันธ์กันได้ โดยเอ็ชทีเอ็มแอลถูกนำมาใช้ในการสร้างเอกสารที่จะใช้ในเวิลด์ไวด์เว็บที่เรียกว่า เว็บเพจ นั่นเอง โครงสร้างของเอกสารเอ็ชทีเอ็มแอล แบ่งเป็น 2 ส่วน คือ

- ส่วนหัวของเอกสาร เป็นส่วนที่ใช้บอกข้อมูลสรุป หรือภาพรวมของเอกสาร ซึ่งต่อมามักใช้เพื่อเพิ่มความสามารถในการโปรแกรมเพื่อทำให้เอกสารเอ็ชทีเอ็มแอลสามารถแสดงผลได้น่าสนใจและน่าใช้งานมากขึ้น

- ส่วนเนื้อหาของเอกสาร เป็นส่วนที่ใช้เก็บเนื้อหาของเอกสารทั้งหมด ซึ่งเราสามารถใส่เนื้อหาทั้งที่เป็นข้อความ ภาพ หรือแม้กระทั่งเสียงเข้าไปในเอกสารเอ็ชทีเอ็มแอลตามรูปแบบที่กำหนด และยังเพิ่มความสามารถในการโต้ตอบกับผู้ใช้งานได้ด้วย

5) ทีซีพี/ไอพี

ย่อมาจาก Transport Control Protocol / Internet Protocol เป็นโปรโตคอลมาตรฐานที่ใช้สำหรับสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ที่เชื่อมต่อกันภายในเครือข่ายอินเตอร์เนต

6) ไอเอสพี (ISP)

ย่อมาจาก Internet Service Provider คือหน่วยงานที่ให้บริการเชื่อมคอมพิวเตอร์เข้าสู่เครือข่ายอินเตอร์เนต นอกจากนี้ยังมักให้บริการเช่าเนื้อที่เพื่อเก็บเว็บเพจสำหรับผู้ที่ต้องการมีเว็บเพจของตัวเองในอินเตอร์เนต

7) เซิร์ฟเวอร์ให้บริการฐานข้อมูลหรือดาต้าเบสเซิร์เวอร์

เป็นระบบคอมพิวเตอร์ที่ทำหน้าที่ให้บริการเรียกค้นและจัดการฐานข้อมูลในอินเตอร์เนต ดาต้าเบสเซิร์เวอร์จะถูกเรียกใช้จากเว็บเซิร์ฟเวอร์อีกต่อหนึ่ง เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอให้ค้นหาข้อมูลจากฐานข้อมูลโดยบราวเซอร์

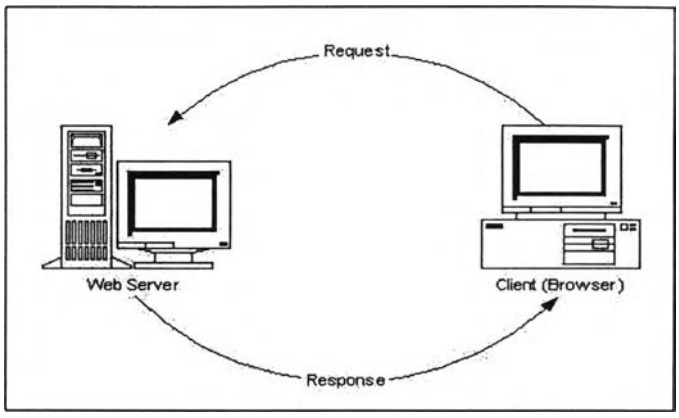
2.5.4 เว็บเพจ

คือเอกสารที่เว็บเซิร์ฟเวอร์ทำการประมวลขึ้นตามคำร้องขอของบราวเซอร์ โดยมีรูปแบบการทำงานระหว่างเวิลด์ไวด์เว็บกับเว็บเพจ แบบไคลเอนต์/เซิร์ฟเวอร์ ซึ่งจะประกอบด้วย คอมพิวเตอร์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ และคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์ ซึ่งโดยทั่วไประบบแบบ ไคลเอนต์/เซิร์ฟเวอร์ จะมีเครื่องคอมพิวเตอร์ ที่ทำหน้าที่เป็นเซิร์ฟเวอร์เพียงเครื่องเดียว ส่วนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์ จะมีจำนวนตั้งแต่หนึ่งเครื่องขึ้นไป ดังนั้นการที่จะนำเว็บเพจไปใช้งานได้ จะต้องมีโปรแกรมที่ทำงานสัมพันธ์กันอยู่ทั้งบนเครื่องเซิร์ฟเวอร์ และบนเครื่องไคลเอนต์ และในทางปฏิบัติ เราสามารถใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียว ทำหน้าที่เป็นทั้งเซิร์ฟเวอร์และไคลเอนต์ พร้อมๆ กันก็ได้

ในการใช้บริการเวิลด์ไวด์เว็บนั้น โปรแกรมที่ทำงานอยู่บนเครื่องเซิร์ฟเวอร์ เราเรียกว่าเว็บเซิร์ฟเวอร์ ส่วนโปรแกรมที่ทำงานบนเครื่องไคลเอนต์ เราเรียกว่าเว็บบราวเซอร์ การทำงานของโปรแกรมเว็บเซิร์ฟเวอร์ และโปรแกรมบราวเซอร์ จะมีการทำงานที่สัมพันธ์กัน กล่าวคือ โปรแกรมบราวเซอร์ จะเป็นส่วนที่ใช้สำหรับติดต่อกับผู้ใช้ผ่านทางจอภาพที่ผู้ใช้ใช้งานอยู่ ดังนั้นหน้าที่หลักของโปรแกรม บราวเซอร์ จึงได้แก่ การรับข้อมูลจากผู้ใช้และการนำข้อมูลที่ส่งมาจากโปรแกรมเว็บเซิร์ฟเวอร์ มาแสดงผล ส่วนหน้าที่หลักของโปรแกรมเว็บเซิร์ฟเวอร์ได้แก่ การจัดเก็บเว็บเพจและนำเว็บเพจตามที่ผู้ใช้ร้องขอโดยผ่านโปรแกรมบราวเซอร์ ส่งกลับไปให้บราวเซอร์แสดงผล

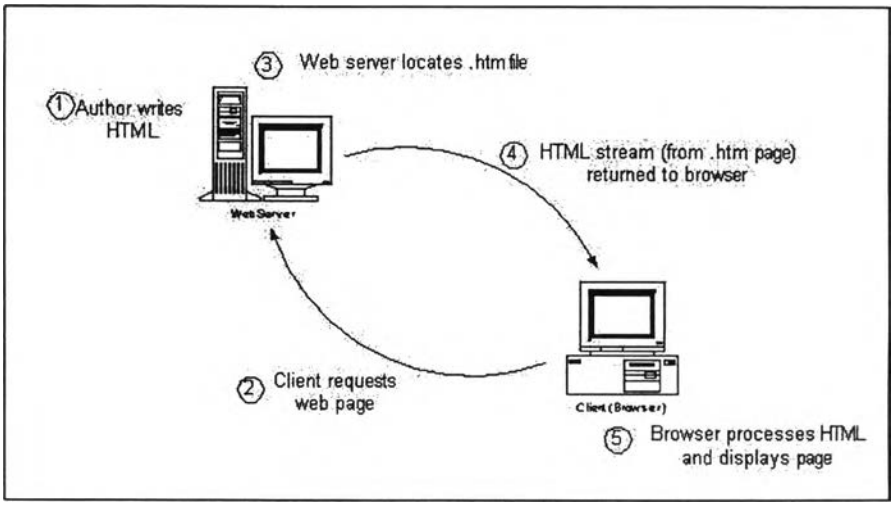
ข้อมูลที่ผู้ใช้ส่งผ่านโปรแกรมบราวเซอร์ จะถูกแปลงให้อยู่ในรูปของคำสั่งที่โปรแกรมเว็บเซิร์ฟเวอร์เข้าใจ ซึ่งเรียกว่า “การร้องขอ” (Request) แล้วจึงส่งไปให้ยังโปรแกรม

เว็บเซิร์ฟเวอร์ เพื่อนำไปประมวลผลหรือนำเว็บเพจที่โปรแกรมบราวเซอร์ต้องการมาแปลงให้อยู่ในรูปของคำสั่งที่โปรแกรม บราวเซอร์ เข้าใจ ซึ่งเรียกว่า "การตอบสนอง" (Response) แล้วจึงส่งไปให้ยังโปรแกรมบราวเซอร์ เพื่อนำไปแสดงผลต่อไป ดังรูปที่ 2.20



รูปที่ 2.20 การสื่อสารข้อมูลระหว่างบราวเซอร์และเว็บเซิร์ฟเวอร์(สัจจะ,2542)

เว็บเพจในยุคแรกๆ ยังจำกัดการตอบสนองต่อคำขอของผู้ใช้อยู่ โดยจะยอมให้ผู้ใช้ร้องขอได้เฉพาะสิ่งที่ได้จัดทำตรียมไว้ล่วงหน้าเท่านั้นดังนั้นรูปแบบของเพจประเภทนี้จะมีลักษณะคงเดิมเสมอ ไม่ว่าผู้เรียกจะเป็นใคร เรียกในเวลาใดก็ตาม เราเรียกเพจประเภทนี้ว่า สแตติกเว็บเพจ(Static Web Page) ซึ่งมีหลักการทำงาน ดังรูปที่ 2.21 ดังนี้



รูปที่ 2.21 แสดงหลักการทำงานแบบสแตติกเว็บเพจ(สัจจะ,2542)

1) ผู้สร้างเว็บจะสร้างเอกสารเอ็ชทีเอ็มแอลหรือเว็บเพจ เก็บไว้ที่เว็บเซิร์ฟเวอร์

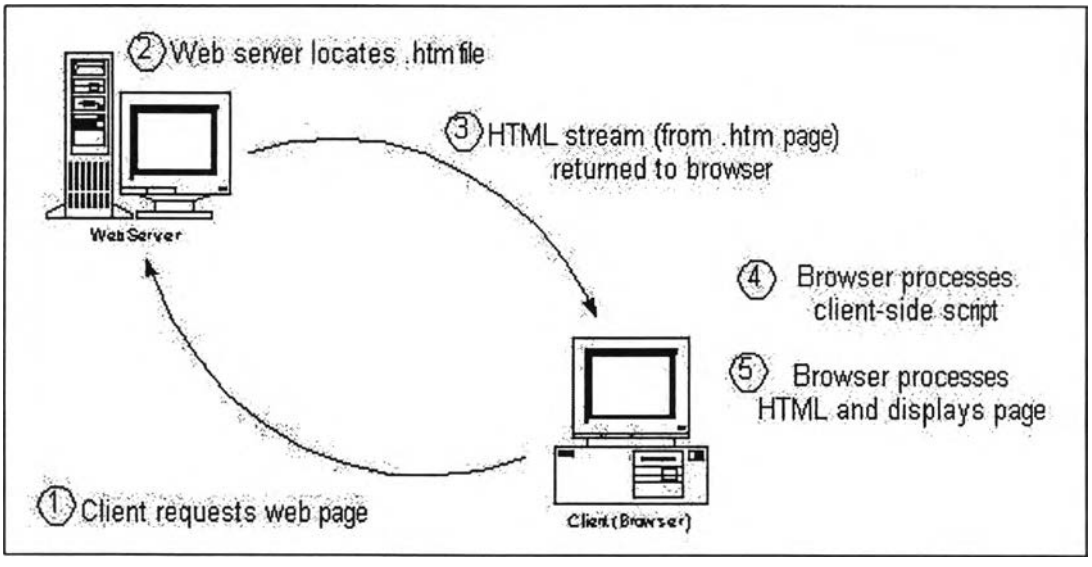
2) ไคลเอนต์ จะส่งการร้องขอ เพื่อเรียก เว็บเพจ ที่สร้างไว้แล้วตามข้อ 1.

3) เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอจากไคลเอนต์ ก็จะทำการค้นหาที่ตั้งของเว็บเพจ ตามที่ไคลเอนต์ ร้องขอ

4) เมื่อหาพบแล้วก็ส่งเอ็ชทีเอ็มแอลสตรีม (HTML Stream) ซึ่งประกอบด้วย เอ็ชทีเอ็มแอลแท็ก(HTML tag) ต่างๆ ที่ประกอบกันเป็นเว็บเพจ ส่งกลับให้บราวเซอร์ที่เครื่องไคลเอนต์

5) เมื่อเครื่องไคลเอนต์ ได้รับเอ็ชทีเอ็มแอลสตรีม ที่ เว็บเซิร์ฟเวอร์ ส่งมา ก็จะนำ เอ็ชทีเอ็มแอลสตรีมเหล่านั้นมาแปลความภาษา เอ็ชทีเอ็มแอล แล้วแสดงผลลัพธ์

ต่อมาได้มีการเพิ่มโปรแกรมสคริปต์ (Script) เข้าไปในเว็บเพจ ทำให้เว็บเพจมีความสามารถในการทำงานเพิ่มเติมในขณะที่แสดงผล เช่น มีการคำนวณขณะแสดงผล แสดงภาพเคลื่อนไหว หรือสามารถตอบสนองการกดปุ่มของผู้ใช้ เป็นต้น เราเรียกเว็บเพจประเภทนี้ว่า ไดนามิคเว็บเพจ(Dynamic Web Page) โดยมีหลักการทำงานเป็นสคริปต์ฝั่งไคลเอนต์ ดังแสดงในรูปที่ 2.22 และสคริปต์ฝั่งเซิร์ฟเวอร์ในรูป 2.23 ดังนี้



รูปที่ 2.22 สคริปต์ฝั่งไคลเอนต์

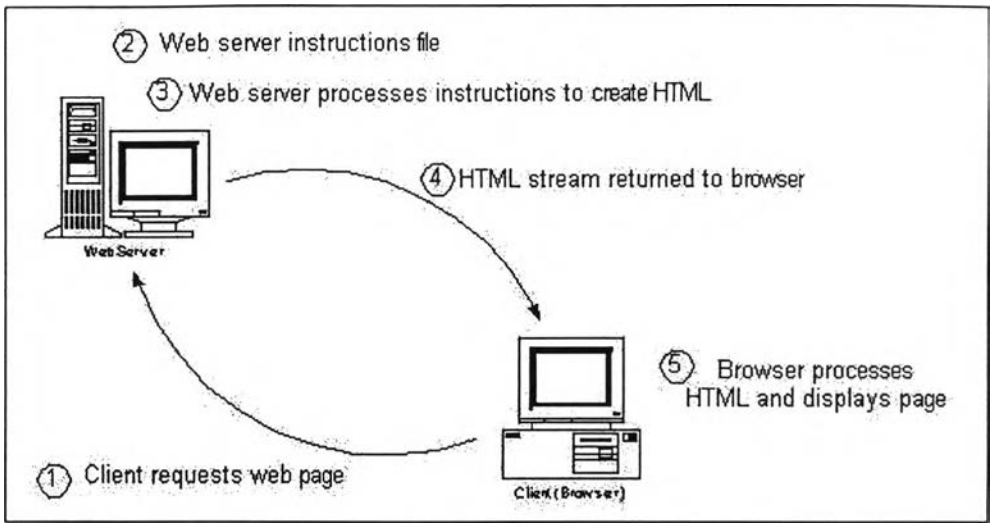
ขั้นที่1. ไคลเอนต์ ส่งการร้องขอ เพื่อเรียกเว็บเพจไปยัง เว็บเซิร์ฟเวอร์

ขั้นที่2.เว็บเซิร์ฟเวอร์ ทำการค้นหาที่ตั้งของเว็บเพจ ตามที่ไคลเอนต์ ร้องขอ

ขั้นที่3.เมื่อพบเว็บเพจตามที่ต้องการ ก็จะนำ เอ็ชทีเอ็มแอลสตรีม ของ เว็บเพจ นั้น ส่งกลับไปยังโปรแกรมบราวเซอร์

ขั้นที่4. โปรแกรมบราวเซอร์ประมวลผล สคริปต์ฝั่งไคลเอนต์ ที่ส่งมากับ เอ็ชทีเอ็มแอลสตรีม

ขั้นที่5.จากนั้นโปรแกรมบราวเซอร์จะทำการประมวล เพื่อแสดงผลลัพธ์



รูปที่ 2.23 สคริปต์ฝั่งเซิร์ฟเวอร์

ขั้นที่1. ไคลเอนต์ ส่งการร้องขอ เพื่อเรียกเว็บเพจ ไปยัง เว็บเซิร์ฟเวอร์

ขั้นที่2.เว็บเซิร์ฟเวอร์ ทำการค้นหาที่ตั้งของเว็บเพจ ที่มีโปรแกรม สคริปต์ฝั่งเซิร์ฟเวอร์ เก็บอยู่ ตามที่ไคลเอนต์ ร้องขอ

ขั้นที่3.เมื่อพบเว็บเพจ ตามที่ต้องการแล้ว เว็บเซิร์ฟเวอร์ก็จะนำมา โปรแกรม สคริปต์ฝั่งเซิร์ฟเวอร์ ที่เก็บไว้ใน เว็บเพจ นั้น มาประมวลผลสร้างเป็น เอ็ชทีเอ็มแอล แท็ค ให้อยู่ภายใน เอ็ชทีเอ็มแอลสตรีม

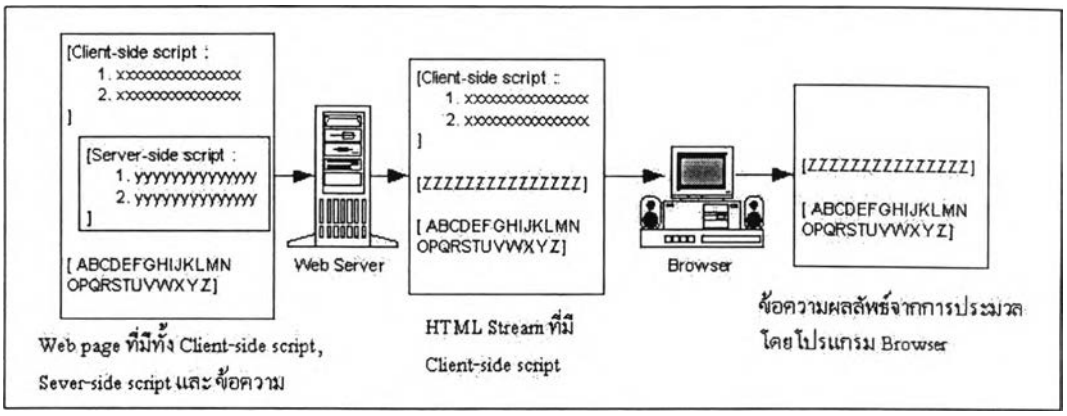
ขั้นที่4.จากนั้นส่ง เอ็ชทีเอ็มแอลสตรีม ที่สร้างขึ้นในขั้นที่ 3 นั้น ส่งกลับไปยังโปรแกรมบราวเซอร์

ขั้นที่5.เมื่อโปรแกรมบราวเซอร์บนเครื่องไคลเอนต์ได้รับ เอ็ชทีเอ็มแอลสตรีม ดังกล่าวแล้ว ก็จะทำการประมวลผลเพื่อแสดงผลลัพธ์

2.5 เอเอสพี (Active Server Page :ASP)

เอเอสพีเป็นเทคโนโลยีที่ใช้เพื่อสร้างโปรแกรมสคริปต์ทั้งที่เป็น สคริปต์ฝั่งเซิร์ฟเวอร์ และ สคริปต์ฝั่งไคลเอนต์ ที่บริษัทไมโครซิฟต์คิดค้นขึ้น โดยตัว เอเอสพีสคริปต์ จะถูกจัดเก็บโดยมีนามสกุล .asp ซึ่งเมื่อถูก เว็บเซิร์ฟเวอร์ นำไปประมวลผล จะถูกประมวลเฉพาะส่วนที่เป็น สคริปต์ฝั่งเซิร์ฟเวอร์ เท่านั้น เพื่อแปลงให้เหลือเฉพาะส่วนของโปรแกรมสคริปต์ฝั่งไคลเอนต์ และผลลัพธ์ต่างๆ ที่ได้จากการประมวลผล ที่อยู่ในรูปของ เอ็ชทีเอ็มแอลแท็ก ก่อนที่จะส่งต่อไปยังโปรแกรมบราวเซอร์ เพื่อทำงานต่อไป ดังรูปที่ 2.24

I 19968978



รูปที่ 2.24 หลักการทำงานของ เอเอสพีสคริปต์

จากรูปแสดงให้เห็นว่า ในเบื้องต้น เว็บเพจจะประกอบด้วยโปรแกรมสคริปต์ฝั่งไคลเอนต์ (คำสั่ง xxxxxxx), สคริปต์ฝั่งเซิร์ฟเวอร์ (คำสั่ง yyyyyyy) และข้อความ (ABCD…Z) สำหรับแสดงด้วย เอ็ชทีเอ็มแอล เมื่อ เว็บเซิร์ฟเวอร์ นำไปประมวลผล ส่วนที่อยู่ภายในโปรแกรม สคริปต์ฝั่งเซิร์ฟเวอร์ จะถูกประมวลผลออกมาเป็นผลลัพธ์ (ZZZZZZZZ) ซึ่งจะถูกส่งไปยังบราวเซอร์ พร้อมกับส่วนอื่นที่เหลือ ต่อมาเมื่อบราวเซอร์ได้รับผลลัพธ์ที่ผ่านจากเว็บเซิร์ฟเวอร์มา ก็จะนำไปประมวลผลส่วนที่เป็นสคริปต์ฝั่งไคลเอนต์ เสร็จแล้วจึงแสดงผลลัพธ์ที่ฝั่งไคลเอนต์ ตามผลลัพธ์ที่ประมวลได้