

# รายงานโครงการวิจัยประจำปีงบประมาณ 2545

เรื่อง

เรื่อง การปรับปรุงประสิทธิภาพและคุณภาพในการพัฒนาระบบสารสนเทศ  
อันเนื่องมาจากการใช้ CASE-Tools

Efficiency and Quality Improvement in Information System  
Development using CASE-Tools

โดย

ผู้ช่วยศาสตราจารย์ ดร. ประจวบ วาณิชชัชวาล

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ 10520

พ.ศ. 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสาร

งานเขียนและแผน  
วันที่... 9 เม.ย. 2546  
เวลา... 10.30 ชม.

แบบรายงานผลโครงการวิจัย

โดยใช้เงินรายได้คณะ

ชื่อโครงการวิจัย (ภาษาไทย) การปรับปรุงประสิทธิภาพและคุณภาพในการพัฒนาระบบสารสนเทศ  
อันเนื่องมาจากการใช้ CASE-Tools

(ภาษาอังกฤษ) Efficiency and Quality Improvement in Information System  
Development using CASE-Tools

รายนามหัวหน้าโครงการและคณะผู้วิจัยพร้อมทั้งหน่วยงานที่สังกัด หมายเลขโทรศัพท์และโทรสาร

หัวหน้าโครงการวิจัย: ผู้ช่วยศาสตราจารย์ ดร. ประจวบ วานิชชัชวาล

อาจารย์ประจำ คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โทรศัพท์ 326-8020, 737-2551-4 ต่อ โทรสาร. 326-9074

E-mail: Prachuab@it.kmitl.ac.th

ได้รับทุนอุดหนุนการวิจัยประเภทเงินรายได้ ประจำปีงบประมาณ 2545

จำนวนเงิน 51,500 บาท เริ่มทำการวิจัยเมื่อ มีนาคม 2545 ถึง 30 กันยายน 2545

รายละเอียดเกี่ยวกับผลงานความก้าวหน้าของการวิจัย

(1) วัตถุประสงค์ของโครงการ (โดยสรุป)

การวิจัยครั้งนี้มีจุดมุ่งหมายที่จะศึกษาถึงผลกระทบของการใช้ CASE-Tools ต่อประสิทธิภาพในกระบวนการพัฒนาระบบสารสนเทศซึ่งมักจะเป็นโครงการระยะยาว ที่จำเป็นจะต้องใช้เวลาพัฒนาและออกแบบเป็นเวลานาน รวมไปถึงการประเมินและปรับปรุงคุณภาพของระบบสารสนเทศที่ได้ถูกจัดทำขึ้นนั้นด้วย เนื่องมาจากการใช้งาน CASE-Tools อย่างเต็มประสิทธิภาพของ CASE-Tools ต่างๆ นั้น ยังไม่ค่อยมีให้เห็นมากนัก อีกทั้งราคาของ CASE-Tools ต่างๆ ก็มีราคาที่ค่อนข้างสูง องค์กรต่างๆ มักจะจัดหาแต่ไม่สามารถใช้งานได้อย่างเต็มประสิทธิภาพ และคุ้มกับราคาที่ต้องจัดหา CASE-Tools นั้นมา จุดที่สำคัญของการวิจัยนี้ นอกจากที่ได้กล่าวมาแล้ว ยังพยายามจะหาวิธีการพัฒนาความสามารถที่จะนำ CASE-Tools มาใช้ในองค์กรเพื่อให้เกิดประโยชน์สูงสุดอีกด้วย

(2) สรุปตารางเปรียบเทียบระหว่างแผนงานวิจัยตามที่เสนอไว้ในโครงการกับงานวิจัยที่ได้ดำเนินการไปแล้ว.

แผนการดำเนินงาน	ผลที่คาดว่าจะได้รับ	ผลจากการปฏิบัติจริง
ช่วงที่ 1 ถึง 3	ศึกษาการใช้ CASE-Tools จัดการอบรมการใช้ CASE-Tools พัฒนาโปรแกรมต้นแบบ	การใช้งาน CASE-Tools นั้น จะเป็นกระบวนการที่ใช้เวลานาน จึงจะเห็นผล ในขั้นแรกนี้ได้ ความก้าวหน้าประมาณ 45%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขหมู่ RCH OA 46.9 :Dsb 2/2147

เลขทะเบียน 54611

วันเดือนปี 24 มี.ค. 2548

อ้างอิงจากแบบ ขส.4 ก บางส่วน

b.11346656

(3) งบประมาณที่ได้ใช้จ่ายไปแล้วนับตั้งแต่เริ่มโครงการเป็นเงินทั้งสิ้น จำนวน 21,500 บาท

(4) งานตามโครงการที่จะทำต่อไป

- การใช้งาน CASE Tools ให้คุ้มกับค่าใช้จ่ายในการจัดหา CASE Tools นั้นๆ ให้มากยิ่งขึ้น
- พัฒนาระบบสารสนเทศต้นแบบที่จะสามารถใช้งานได้จริง
- เพิ่มประสิทธิภาพในการพัฒนาระบบสารสนเทศ

(5) คำชี้แจงเกี่ยวกับอุปสรรคหรือปัญหา (ถ้ามี)

การพัฒนาระบบสารสนเทศเป็นกระบวนการที่ต้องใช้เวลาค่อนข้างยาว ความก้าวหน้าขณะนี้สามารถประเมินได้ประมาณ 45 ถึง 50% ของโครงการทั้งหมด ดังนั้นจึงยังไม่สามารถสรุปผลสุดท้ายได้ในการดำเนินการวิจัยสำหรับปีงบประมาณนี้

(ลงชื่อ)



หัวหน้าโครงการ

(ผู้ช่วยศาสตราจารย์ ดร. ประจวบ วาณิชชวัล)

20 พฤศจิกายน 2545

\* **Repository Administration Utility** ใช้สำหรับการสร้างและดูแล Repository รวมถึงการควบคุมการเข้าถึง Repository

- **Online Documentation** เป็นเอกสารฉบับไลต์เกี่ยวกับวิธีการใช้และหลักการทำงานของ Oracle Designer 6i

### 2.3.1.2 การทำงานฝั่งเซิร์ฟเวอร์

มีส่วนประกอบหลักในการจัดการอยู่ 2 ส่วน คือ

#### 1) Repository

เปรียบเสมือนเป็นหัวใจของ Oracle Designer เนื่องจาก Repository อำนวยความสะดวกในการออกแบบและสร้างแอปพลิเคชันที่รวดเร็วและมีประสิทธิภาพ

Oracle Repository เป็นชุดเครื่องมือสำหรับเก็บและจัดการ Meta Data ของแอปพลิเคชัน ซึ่งสนับสนุนการทำงานในสภาพแวดล้อมแบบ Multi-user ทำให้สนับสนุนการพัฒนาแบบเป็นทีม เพราะสามารถเข้าถึง Repository ร่วมกันได้ นอกจากนี้เมื่อมีการจัดการเกิดขึ้นกับอ็อบเจกต์ใดก็ตามใน Repository จะไม่เกิดการเปลี่ยนแปลงกับ Definition ของอ็อบเจกต์นั้นจริงๆ การเปลี่ยนแปลงจะเป็นแบบ Logical ไม่ใช่ Physical ผู้ที่ทำหน้าที่สร้าง, ดูแล และกำหนดสิทธิ์ในการเข้าถึง Repository เรียกว่า Repository Administration

#### 2) API (Application Programmatic interface)

เป็นส่วนที่ทำหน้าที่ในการ Update Repository ซึ่งจะ Update ทันที เมื่อมีการเปลี่ยนแปลงและแก้ไขข้อมูลใดๆผ่าน Tools ต่างๆของ Oracle Designer 6i

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มา

เทคโนโลยีทางด้านซอฟต์แวร์คอมพิวเตอร์ได้มีการเปลี่ยนแปลงรวดเร็วเป็นอย่างมาก จากที่เคยเขียนโปรแกรมด้วย Third-Generation Language ต่อมาเปลี่ยนเป็น Fourth-Generation Language จนเมื่อหลายปีที่ผ่านมา CASE Tools ได้เริ่มเข้ามามีบทบาทสำคัญเพิ่มขึ้นเรื่อยๆ จนเป็นที่ยอมรับกันว่าเป็นเทคนิคในการพัฒนาระบบงานคอมพิวเตอร์ที่ฉลาดที่สุดในยุคปัจจุบัน และน่าจะเป็นเครื่องมือที่ใช้กันอย่างแพร่หลายในอนาคต เนื่องจาก CASE Tools เป็นเครื่องมือที่ช่วยให้ผู้พัฒนาระบบสามารถพัฒนาระบบงานที่ซับซ้อนได้ง่ายและรวดเร็วขึ้น และสะดวกต่อการปรับปรุงแก้ไข

สำหรับโครงการนี้ได้นำเอา CASE Tools ที่ชื่อว่า Oracle Designer6i มาช่วยในการออกแบบและพัฒนาระบบงานทะเบียนนักศึกษา ซึ่งเป็นระบบงานที่ค่อนข้างใหญ่และซับซ้อน มีหน่วยงานที่เกี่ยวข้องและข้อมูลจำนวนมาก การนำเอา CASE Tools มาช่วยในการพัฒนาระบบจัดการข้อมูลและจัดทำเอกสาร จึงเป็นสิ่งจำเป็นอย่างยิ่ง เนื่องจาก Oracle Designer6i นี้สนับสนุนการทำงานแบบ Spiral ทำให้ลดงานที่ซ้ำซ้อนลงได้ และมีส่วนการติดต่อกับผู้ใช้ (User Interface) ที่ง่ายต่อการใช้งาน โดย Oracle Designer6i นี้จะช่วยในการจัดการฐานข้อมูลของระบบ, ช่วยในการวิเคราะห์กิจกรรมต่างๆของระบบ และทำการออกแบบโครงสร้างของระบบ เพื่อนำไปสร้างเป็นโปรแกรมประยุกต์ใช้งานต่อไป อีกทั้งยังช่วยในการจัดทำเอกสารคู่มือและรายงานตลอดทุกขั้นตอนของการพัฒนาระบบ เรียกได้ว่าในขั้นตอนการพัฒนาระบบทุกขั้นตอนสามารถนำ Oracle Designer6i นี้ไปช่วยทั้งในการเขียน Diagram ต่างๆ การออกรายงาน จนกระทั่งสร้างตัวโปรแกรมที่เป็นโปรแกรมต้นแบบโดยง่ายให้

### 1.2 วัตถุประสงค์ของโครงการ

ในการออกแบบและพัฒนาระบบงาน มีวัตถุประสงค์ ดังนี้

1.2.1 ศึกษากระบวนการพัฒนาระบบสารสนเทศและการทำงานของ CASE Tools เพื่อใช้เป็นเครื่องมือในการออกแบบและพัฒนาระบบงาน

1.2.2 ออกแบบและพัฒนาระบบทะเบียนนักศึกษาโดยใช้ Oracle Designer6i CASE Tool

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขอบเขตของโครงการ

สำหรับโครงการพัฒนาระบบงานนี้ จะทำการวิเคราะห์ ออกแบบ และพัฒนาระบบสารสนเทศที่เกี่ยวเนื่องนักศึกษาในส่วนงานการลงทะเบียนและการจัดการข้อมูลคณะและภาควิชา ซึ่งระบบงานนี้จะใช้เป็นกรณีศึกษา เพื่อที่จะแสดงให้เห็นว่าการใช้ Oracle Designer6i CASE Tool ช่วยให้การพัฒนาระบบง่าย รวดเร็วและมีประสิทธิภาพขึ้น อีกทั้งยังสะดวกต่อการปรับปรุงแก้ไข

### 1.4 ขั้นตอนและวิธีการดำเนินการ

1.4.1 ศึกษาหลักการและทฤษฎีต่างๆที่เกี่ยวข้อง เช่น ทฤษฎี CASE Method, ระบบฐานข้อมูลแบบ Relational, ภาษา PL/SQL เป็นต้น

1.4.2 ศึกษาหลักการงานและวิธีการใช้งาน Oracle Designer6i CASE Tool

ออกแบบและพัฒนาระบบทะเบียนนักศึกษาโดยใช้ Oracle Designer6i CASE Tool ในการออกแบบ และทำการปรับปรุงฟอร์มแอปพลิเคชัน รวมทั้งออกรายงานต่างๆ โดยใช้ Oracle Form6i และ Oracle Report6i

1.4.3 สำหรับขั้นตอนในการพัฒนา เริ่มตั้งแต่การวิเคราะห์และออกแบบระบบโดยใช้ Entity Relationship Diagram, Data Flow Diagram, Function Hierarchy Diagram ลงบนเครื่องมือ Oracle Designer6i หลังจากนั้นจะทำการสร้างตัวโปรแกรมและฐานข้อมูลขึ้นมาโดยอัตโนมัติ จากนั้นทำการปรับปรุงตัวโปรแกรมโดยใช้ Oracle Form6i และ Oracle Report6i ส่วนฐานข้อมูลจะถูกนำไปสร้างบนระบบจัดการฐานข้อมูล Oracle8i

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ในการพัฒนาระบบ โดยใช้ CASE Tools คาดว่าจะได้รับประโยชน์ ดังนี้

1.5.1 จัดเก็บข้อมูลที่ใช้ในการพัฒนาระบบรวมไว้ที่เดียวกันอย่างครบถ้วน เพื่อลดเวลาในการสืบค้นและเวลาในการดูแล และปรับปรุงแก้ไขระบบ เช่น Requirement Specification, Program Specification, Source Code, และมาตรฐานการเขียนโปรแกรม เป็นต้น

1.5.2 เกิดความคล่องตัวในการประสานงานระหว่างผู้ที่เกี่ยวข้องกับการพัฒนาระบบ

1.5.3 ทำให้การพัฒนาระบบมีขั้นตอนการทำงานที่มีรูปแบบเป็นมาตรฐานเดียวกันทั้งองค์กร

1.5.4 แนวคิดการใช้ CASE Tools ไปพัฒนาระบบอื่นๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 CASE Tools

CASE Tools (Computer-aided Software Engineering) เป็นเครื่องมือที่พัฒนาขึ้นมาเพื่อใช้ในการออกแบบและสร้างระบบสารสนเทศ ประกอบด้วยการวิเคราะห์ระบบ การออกแบบระบบ และการสร้างระบบ โดยจะเริ่มตั้งแต่การรวบรวมความต้องการทางธุรกิจของระบบที่จะพัฒนา แล้วนำความต้องการเหล่านั้นมาออกแบบเป็นโครงสร้างไว้ แล้วนำโครงสร้างดังกล่าวมาสร้างเป็นระบบสารสนเทศขึ้นมา ซึ่งวิธีการพัฒนาระบบสารสนเทศโดยใช้ CASE Tools นี้เรียกว่า CASE Method

ปัจจุบันนี้มีการนำเอา CASE Tools มาช่วยในการออกแบบ วิเคราะห์ ควบคุมและทำเอกสารในระบบงานใหญ่ๆมากขึ้น เนื่องจากบริษัท และองค์กรต่างๆเห็นว่า CASE Tools ช่วยในการสร้าง Diagram และช่วยในการแก้ไข ปรับปรุงงานในระบบงานที่มีขนาดใหญ่ได้เป็นอย่างดี อีกทั้ง CASE Tools ต่างๆที่พัฒนาขึ้นมาในปัจจุบัน มีองค์ประกอบที่ทำให้สามารถใช้งานได้ง่ายและมีประสิทธิภาพมากขึ้น ตัวอย่างเช่น มีการใช้ Graphics ช่วยในการออกแบบและสร้าง Diagram ต่างๆ , มีตัว Generators สำหรับสร้างหน้าจอเพื่อใช้ในการแสดงผล Input/Output ของระบบ, มี Repository ที่ใช้เก็บข้อมูลเกี่ยวกับการออกแบบทั้งหมดไว้ในที่เดียวกัน เหมาะกับระบบงานใหญ่ๆที่มีการใช้งานแบบ Multi-user

ในการนำ CASE Tools มาใช้ในการวิเคราะห์และออกแบบระบบนั้น ช่วยให้ขั้นตอนในการพัฒนาระบบมีระเบียบวิธีที่เป็นมาตรฐาน ลดระยะเวลา ค่าใช้จ่าย ตลอดจนการดูแลรักษาระบบที่พัฒนาขึ้น ทำให้สามารถปรับปรุงเปลี่ยนแปลงได้ง่าย

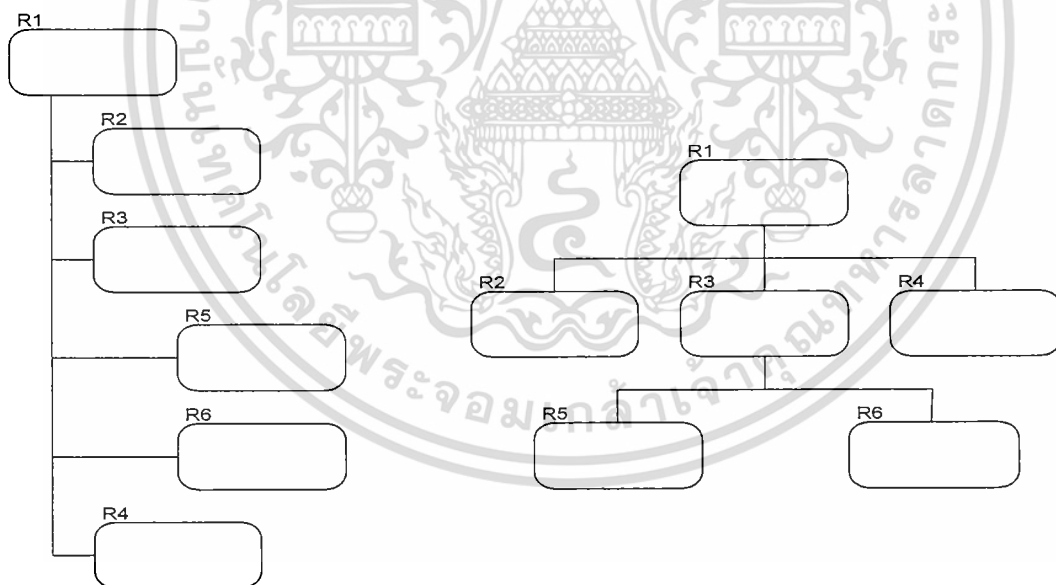
ในขั้นตอนการวิเคราะห์ระบบโดยใช้ CASE Tools มีการใช้เทคนิคในการสร้างไดอะแกรมต่างๆ ดังนี้

##### 2.1.1 Function Hierarchy Diagram

เป็นการเขียนกิจกรรมทั้งหมดที่ต้องทำภายในองค์กรหรือในหน่วยงานหนึ่งๆ โดยการนำงานย่อยแต่ละอย่างหรือฟังก์ชันทางธุรกิจ (Business Function) มาเขียนไล่เป็นลำดับชั้น ฟังก์ชันที่อยู่บนจะถูกแตกเป็นฟังก์ชันย่อยๆในลำดับชั้นถัดลงมา จนกระทั่งได้โครงสร้างขององค์กรเป็น

Hierarchy ที่สมบูรณ์ และครอบคลุมการทำงานทั้งหมด รวมทั้งสามารถเข้าใจการทำงานนั้นได้ การแตกย่อยของฟังก์ชันก็จะสิ้นสุด ในการแตกฟังก์ชันย่อยนี้ลำดับก่อนหลังที่ปรากฏในฟังก์ชันไม่ใช่สิ่งสำคัญ (ไม่เน้นว่าจะต้องเรียงลำดับก่อนหลัง) แต่จะเน้นถึงลำดับชั้นของฟังก์ชันในแต่ละระดับ (level) มากกว่า ฟังก์ชันที่อยู่ในชั้นที่สูงกว่าจะถูกอธิบายโดยฟังก์ชันที่อยู่ด้านล่างที่แตกออกมาเรื่อยๆ นอกจากนี้การเขียน

สัญลักษณ์ที่ใช้แสดงฟังก์ชันทางธุรกิจ (Business Function) จะเป็นรูปกรอบสี่เหลี่ยมที่มีมุมขอบมน (แต่ใน Oracle Designer6i CASE Tools จะใช้เป็นรูปสี่เหลี่ยมผืนผ้า) ภายในกรอบคือรายละเอียดของฟังก์ชัน ที่มุมซ้ายบนแสดงชื่อสั้นๆไว้เป็นเลเบล (Label) เพื่อให้สะดวกในการเรียก และอาจมีเครื่องหมาย  $\oplus$  ที่มุมขวาบน เพื่อแสดงว่าฟังก์ชันดังกล่าวยังสามารถแตกรายละเอียดฟังก์ชันนั้นออกเป็นฟังก์ชันย่อยๆลงไปได้อีก แต่รายละเอียดไม่ได้ปรากฏให้เห็นบนแผนภาพในขณะนั้น การเลือกใช้ชื่อฟังก์ชันจะต้องบ่งบอกถึงคำจำกัดความว่าฟังก์ชันนั้นมีหน้าที่ทำอะไรอย่างชัดเจน ชื่อของฟังก์ชันจะต้องขึ้นต้นด้วยคำกริยาและจะต้องไม่ระบุถึงวิธีปฏิบัติ นอกจากนี้ควรหลีกเลี่ยงการใช้คำศัพท์เฉพาะทาง, อักษรย่อ, คำย่อที่อาจสื่อความหมายกำกวม



รูปที่ 2.1 Function Hierarchy Diagram

อีกองค์ประกอบหนึ่งที่แสดงอยู่ในโมเดล (Model) คือ เหตุการณ์ (Event) จะแสดงด้วยสัญลักษณ์ลูกศรที่มีข้อความภายในลูกศร เหตุการณ์ดังกล่าวอาจเป็นเหตุการณ์ผลลัพธ์ (Outcome) ใดๆ หรือเป็นเหตุการณ์ใดๆที่กระตุ้น (Trigger) ให้แต่ละฟังก์ชันเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการแสดงการย่อยของฟังก์ชันอาจจะแสดงได้ทั้งแนวนอน (Horizontal Layout), แนวตั้ง (Vertical Layout) หรือแบบผสม (Hybrid Layout) ก็ได้ ซึ่งตามทฤษฎีแล้วถือว่าเหมือนกัน ไม่ว่าจะแสดงในรูปแบบใด ขึ้นอยู่กับเนื้อหาในการใช้งานและความสะดวกในการใช้งาน

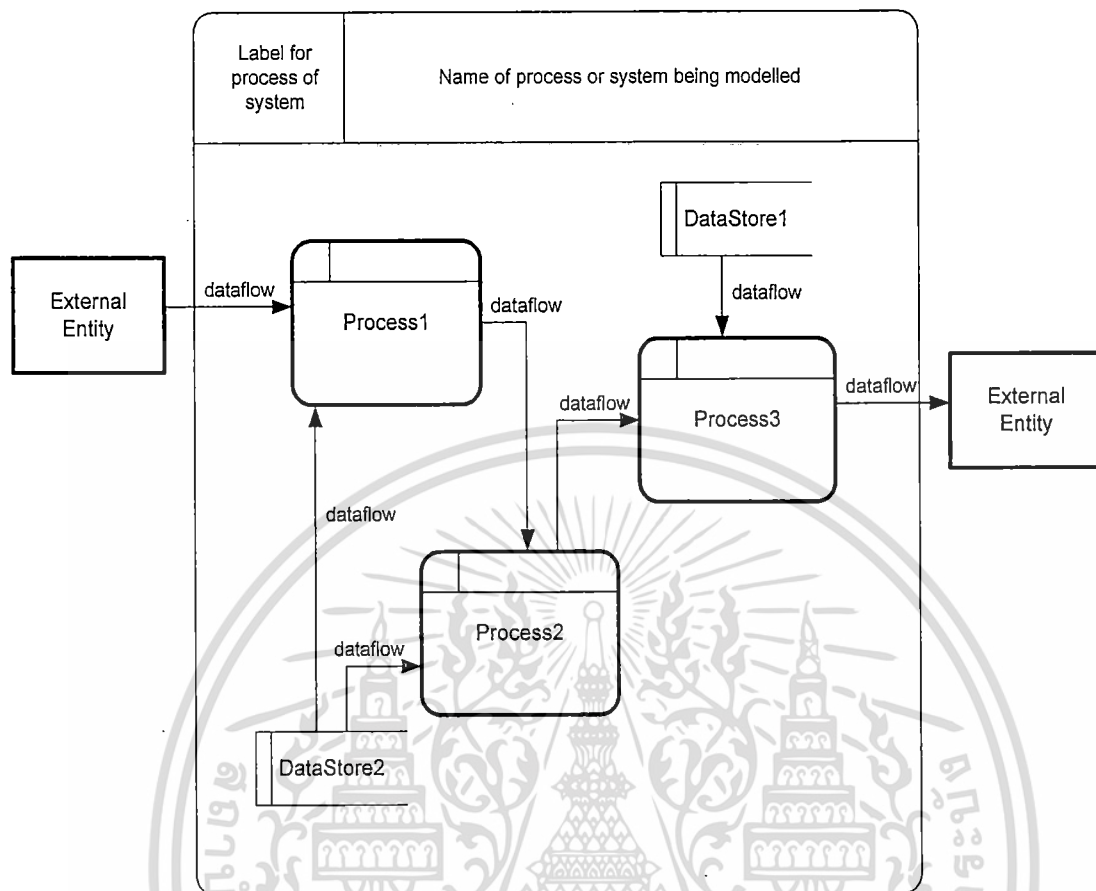
### 2.1.2 Dataflow Diagram

ไดอะแกรมชนิดนี้ใช้แสดงรายละเอียดว่ากิจกรรมในองค์กรนั้นต้องใช้ข้อมูลอย่างไร โดยจะแสดงให้เห็นว่าข้อมูลใดจะถูกจัดเก็บ และแสดงให้เห็นว่าข้อมูลต่างๆจะถูกดำเนินการผ่านโพรเซสใดบ้าง

ประโยชน์ของไดอะแกรมชนิดนี้ คือ ใช้ช่วยในการตัดสินใจว่าควรจะทำการแตกโพรเซสลงไปเป็นโพรเซสย่อยอีกหรือไม่ จนกระทั่งได้ไดอะแกรมในระดับที่เป็นรายละเอียดที่สุดแล้ว ไดอะแกรมนั้นก็เหมาะที่จะใช้ในการกำหนด Specification สำหรับงานออกแบบ, เขียน โปรแกรม, เขียนเอกสารคู่มือ หรือใช้ออกแบบอุปกรณ์ต่างๆเพื่อใช้ในระบต่อไป มีข้อสังเกตว่า ไดอะแกรมนี้ไม่สามารถอธิบายการทำงานภายในโพรเซสหรือฟังก์ชันได้ และไม่สามารถบอกได้ว่าเมื่อใดจึงจะเกิดการ ทำงานของ โพรเซสใด เพราะไม่มีการแสดง Event ไว้ในไดอะแกรมนี้เลย

นอกจากโพรเซสแล้ว ไดอะแกรมนี้จะมีส่วนประกอบสำคัญอีก 3 ส่วน คือ

- **Dataflow** ใช้แสดงข้อมูลใดๆที่ถูกแลกเปลี่ยนระหว่าง 2 โพรเซส หรือระหว่างโพรเซสกับ Datastore สัญลักษณ์ของ Dataflow คือเส้นตรงที่มีหัวลูกศรชี้ไปที่ทิศทางที่เกิดการไหลของข้อมูล ซึ่งอาจเกิดได้ทั้ง 2 ทาก็จะเป็นเส้นตรงที่มีหัวลูกศรทั้ง 2 ปลาย
- **Datastore** ใช้แทนหน่วยข้อมูลที่ถูกจัดเก็บ โดยจะมีการกำหนดชื่อไว้ใช้อย่างถึงจากโพรเซสต่างๆ สัญลักษณ์ของ Datastore จะเป็นรูปกรอบสี่เหลี่ยมตัดมุมที่เปิดด้านหนึ่ง ภายในกรอบแสดงชื่อย่อและคำอธิบาย
- **External Entity** เป็นสัญลักษณ์ที่ใช้แสดงสิ่งใดๆที่อยู่นอกขอบเขตระบบงาน แต่สัมพันธ์กับระบบด้วยการส่งผ่านของข้อมูลถึงกันอยู่



รูปที่ 2.2 องค์ประกอบของ Dataflow Diagram

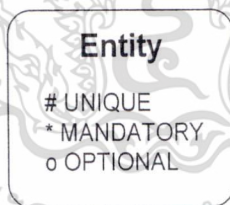
### 2.1.3 Entity Relationship Diagram

เป็นโมเดลที่ใช้แสดงความสัมพันธ์ของข้อมูลระหว่าง Entity เพื่อใช้กำหนดข้อมูลที่ระบบต้องการ เป็นวิธีการ (Algorithm) ที่สามารถออกแบบฐานข้อมูลให้มีลักษณะเป็น โครงสร้าง (Data Structure) ที่มีข้อมูลอยู่บนนั้นเลย ซึ่ง ER Model นี้มีรูปแบบการนำเสนอที่สามารถมองจากโมเดลแล้วเข้าใจได้โดยง่าย

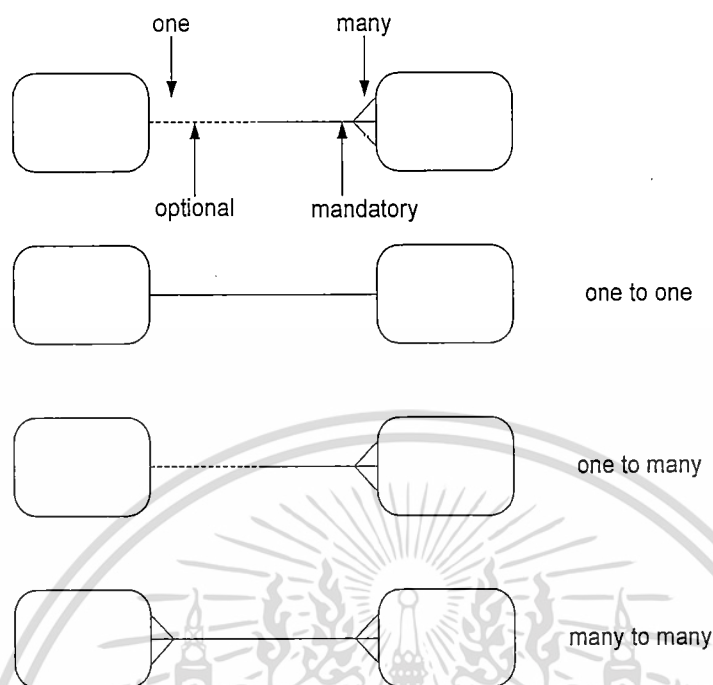
ส่วนประกอบของ Entity Relationship Diagram

- เอนทิตี (Entity) คือ สิ่งของหรือสิ่งที่เราสนใจหรือหัวข้อที่เป็นหลักสำคัญ อาจจะเป็นสิ่งที่มีอยู่จริง หรือถูกสมมติขึ้นมาก็ได้ จะเป็นสิ่งที่บรรจุหรือเกี่ยวข้องกับข้อมูลด้านต่างๆที่จำเป็นต้องรู้ เช่น คน, สถานที่, สิ่งของ, สินค้า, การสั่งซื้อ เป็นต้น โดยจะแสดงอยู่ในกรอบสี่เหลี่ยมขอบมน (Soft Box)

- แอททริบิวต์ (Attribute) คือ รายละเอียดต่างๆที่บ่งบอกได้ถึงคุณภาพ, ลักษณะ, ประเภท, จำนวน หรือข้อกำหนดคุณลักษณะที่เกี่ยวข้องกับสถานะของเอนทิตี โดย แอททริบิวต์สามารถแบ่งออกเป็น 3 ลักษณะ คือ
  - Unique Identifier (UID) เป็นแอททริบิวต์ที่ค่าในแอททริบิวต์นั้นไม่ซ้ำกัน และไม่สามารถที่จะเป็นค่า Null ได้ ใช้เครื่องหมาย # หน้าแอททริบิวต์นั้นๆ
  - Mandatory Attribute เป็นแอททริบิวต์ที่จะต้องมียู่อค่าอยู่เสมอ ไม่สามารถที่จะเป็นค่า Null ได้ ใช้เครื่องหมาย \* หน้าแอททริบิวต์นั้นๆ
  - Optional Attribute เป็นแอททริบิวต์ที่อาจจะมีค่าหรือไม่มีก็ได้ สามารถเป็นค่า Null ได้ ใช้เครื่องหมาย o หน้าแอททริบิวต์นั้นๆ
- ความสัมพันธ์ (Relationship) คือ สิ่งที่สำคัญในการเชื่อมโยงระหว่าง 2 เอนทิตีซึ่งสามารถแสดงความสัมพันธ์ได้ทั้งที่เป็น Optional (เป็นเส้นประ) หรือความสัมพันธ์แบบ Mandatory (เป็นเส้นทึบ) คือ จะต้องมียู่อค่าข้อมูลเสมอ ไม่สามารถเป็นค่า Null ได้ และยังสามารถแสดงถึงความสัมพันธ์ในลักษณะที่เป็นหนึ่งต่อหนึ่ง (One to One) หรือหนึ่งต่อกลุ่ม (One to Many) หรือกลุ่มต่อกลุ่ม (Many to Many)



รูปที่ 2.3 สัญลักษณ์ของ Entity, Attribute และสัญลักษณ์ที่ใช้



รูปที่ 2.4 สัญลักษณ์ของความสัมพันธ์ (Relationship)

## 2.2 วงจรการพัฒนาระบบงาน (System Development Life Cycle)

ระบบงานต่างๆ ไม่ว่าจะเป็นงานด้านธุรกิจ การศึกษา หรืออื่นๆ ในปัจจุบันนี้ มีความซับซ้อนเพิ่มมากขึ้น การวิเคราะห์ระบบงานต่างๆ เหล่านี้จึงเกิดความยุ่งยากอันเนื่องมาจากความซับซ้อนของระบบงานเอง ด้วยเหตุนี้การพัฒนาระบบงานควรจะมีมาตรฐานขึ้น เพื่อช่วยลดความซับซ้อนที่เกิดขึ้นภายในระบบงานได้ รวมไปถึงทำให้เกิดข้อผิดพลาดได้น้อยที่สุด

วงจรการพัฒนาระบบงานก็เป็นหลักการหนึ่งที่น่าสนใจในการพัฒนาระบบเพื่อตอบสนองความต้องการเหล่านั้น โดยในวงจรพัฒนาระบบงานนี้จะแบ่งออกเป็น 7 ขั้นตอน ดังนี้

### 1. Problem Definition

ในขั้นตอนนี้ เป็นขั้นตอนการศึกษาระบบ และการกำหนดขอบเขตในการศึกษา รวมไปถึงลักษณะของปัญหาที่ผู้ใช้ประสบอยู่ โดยจะต้องทำการค้นหาให้ได้อย่างชัดเจนว่า สิ่งที่ผู้ใช้ต้องการในระบบนี้คืออะไร หรือว่าปัญหาที่เกิดขึ้นในการทำงานคืออะไรบ้าง

### 2. Requirement Analysis

ในขั้นตอนนี้ จะต้องทำการวิเคราะห์การทำงานของระบบปัจจุบัน เพื่อทำการค้นหาวิธีการในการแก้ไขปัญหาที่เกิดขึ้นในระบบปัจจุบัน หรือวิธีการที่จะทำให้สามารถตอบสนองความต้องการที่เพิ่มขึ้นของผู้ใช้ได้

### 3. Specification

เป็นขั้นตอนที่จะต้องระบุให้ได้ว่า ระบบจะพัฒนาไปในแนวทางใด ด้วยวิธีการใด และมีความจำเป็นที่จะต้องมียุทธศาสตร์ประกอบอะไรบ้าง โดยจะต้องมีความชัดเจน เพื่อให้สามารถนำไปทำการออกแบบระบบได้

### 4. Design

เป็นขั้นตอนที่ให้นำสิ่งที่ได้ระบุเอาไว้มาทำการออกแบบระบบงาน โดยการออกแบบจะต้องให้มีความสอดคล้องกับการทำงานในระบบปัจจุบัน และจะต้องสามารถรองรับความต้องการ และสามารถแก้ไขปัญหาที่เกิดขึ้นของผู้ใช้ได้

### 5. Coding

เป็นขั้นตอนที่นักพัฒนาระบบเริ่มที่จะทำการเขียนโปรแกรม เพื่อให้มีความสอดคล้องกับความต้องการของผู้ใช้ และเป็นไปตามสิ่งที่ได้ออกแบบไว้

### 6. Testing

เป็นขั้นตอนในการทดสอบระบบที่พัฒนาขึ้นมา เพื่อเป็นการยืนยันว่าระบบที่พัฒนาขึ้นมา มีความถูกต้องสมบูรณ์ สอดคล้องกับที่ได้ตกลงกันเอาไว้หรือไม่

### 7. Operation and Maintenance

เป็นขั้นตอนที่ผู้ใช้เริ่มนำระบบไปใช้งานจริงภายใต้สภาวะแวดล้อมจริงๆ เมื่อใช้ระบบไปสักระยะหนึ่งอาจจะเกิดความผิดพลาดขึ้น ซึ่งจะเข้าสู่ส่วนของการบำรุงรักษาระบบ โดยในขั้นตอนนี้ก็จะมีกรปรับปรุงเปลี่ยนแปลง หรือแก้ไขให้รองรับกับความต้องการของผู้ใช้ที่เพิ่มขึ้น

## 2.3 Oracle Designer6i

บริษัท Oracle เป็นบริษัทหนึ่งที่ได้มีการพัฒนา CASE Tools ขึ้นมา ซึ่ง Oracle Designer6i ก็เป็น CASE Tool ตัวหนึ่งของบริษัท Oracle ที่ได้ทำการพัฒนาให้สามารถใช้พัฒนาระบบงานได้ง่าย รวดเร็ว และสร้าง Application ได้อย่างรวดเร็ว มีประสิทธิภาพ แล้วยังรองรับการออกแบบระบบฐานข้อมูลแบบ Object และฐานข้อมูลแบบ Relational Database ได้โดยไม่มีปัญหาใด ๆ

การเก็บรวบรวมข้อมูลต่าง ๆ ของ Oracle Designer6i นั้นจะเก็บไว้ที่ส่วนกลางที่เรียกว่า Repository ซึ่งจะเก็บ Object, Meta Data ที่เกี่ยวกับการวิเคราะห์ ออกแบบและการสร้างระบบ ของระบบที่ทำการพัฒนา

ความสามารถในการสร้างระบบสารสนเทศของ Oracle Designer6i นั้นมีความสามารถที่ครอบคลุมการออกแบบระบบ กล่าวคือ สามารถที่จะสร้าง Forms ต่าง ๆ, Web Applications ซึ่งมีการสนับสนุนภาษา Java, Reports ของระบบสามารถสร้าง Menu รวมไปถึงสามารถที่จะสร้าง Document ของระบบได้อีกด้วย

### 2.3.1 ส่วนประกอบของ Oracle Designer 6i

การทำงานของ Oracle Designer 6i มี 2 ส่วน คือ ฟังก์ชันไคลเอนต์ และฟังก์ชันเซิร์ฟเวอร์ ในส่วนของฟังก์ชันไคลเอนต์จะเกี่ยวข้องกับเครื่องมือ และ Utility ที่ใช้ในการออกแบบ การพัฒนา และการจัดการรายละเอียดต่าง ๆ ภายในระบบ และในส่วนของฟังก์ชันเซิร์ฟเวอร์จะมีความเกี่ยวข้องกับงานที่เกี่ยวกับการจัดการ Repository ที่เป็นศูนย์รวมของข้อมูลและออบเจกต์ทั้งหมดที่มีภายในระบบ

#### 2.3.1.1 การทำงานฟังก์ชันไคลเอนต์

บนฟังก์ชันไคลเอนต์มีเครื่องมือต่างๆที่สนับสนุนทุกขั้นตอนของการพัฒนาระบบ ดังนี้

- **Analysis and Design Tools**

เครื่องมือในการวิเคราะห์และออกแบบระบบ ประกอบด้วย 3 ส่วนหลักๆ ได้แก่

##### 1. Model System Requirements

เป็นเครื่องมือที่ใช้ในการวิเคราะห์ความต้องการของผู้ใช้ สร้างออกมาเป็น โมเดล ซึ่งจะเป็นอินพุตของ Transformer ต่อไป ซึ่งประกอบด้วย

##### - Process Modeler

เป็นเครื่องมือที่ช่วยให้ผู้ออกแบบระบบสามารถสร้าง โมเดลของธุรกิจตามวัตถุประสงค์ขององค์กร เพื่ออธิบาย Business Process ออกมาในลักษณะรูปภาพ และ Flow การทำงานของธุรกิจ และแสดงความสัมพันธ์ของแต่ละ Process เพื่อให้ผู้พัฒนาระบบและผู้ใช้ระบบมองเห็นระบบออกมาในภาพเดียวกัน ช่วยให้เกิดความร่วมมือในการสร้างและแก้ไข Model ร่วมกัน เพื่อให้ได้ Model ที่ตรงกับความต้องการทางธุรกิจที่จะพัฒนาระบบ และสามารถนำไปใช้พัฒนาต่อในขั้นตอนอื่นได้อย่างถูกต้อง

##### - Function Hierarchy Diagrammer

เป็นเครื่องมือที่ใช้แสดงการแบ่งย่อยของกิจกรรมต่างๆในองค์กร ว่าในองค์กรนั้นมีฟังก์ชันการทำงานอะไรบ้าง และแบ่งย่อยฟังก์ชันเหล่านี้ลงไปจนกระทั่งรู้รายละเอียดของงานครบถ้วน

#### - Dataflow Diagrammer

เป็นเครื่องมือที่ใช้ในการทำแผนภาพแสดงการไหลของข้อมูลภายในธุรกิจ ซึ่งจะแสดงรายละเอียดงานของระบบว่ามีการใช้ข้อมูลอะไรบ้าง และมีการแลกเปลี่ยนข้อมูลกันอย่างไร

#### - Entity Relationship Diagrammer

เป็นเครื่องมือที่ใช้แสดงความต้องการภายในระบบขององค์กร และกำหนดข้อมูลที่ระบบงานนั้นๆต้องการ โดยมีการสร้างโมเดลที่มีลักษณะเป็นกราฟิกเพื่อใช้แสดงความสัมพันธ์ระหว่าง Entity ว่าเกี่ยวข้องกันอย่างไรในระบบงาน และทำการกำหนด Entities, Attributes, Relationship, และ Domains ทั้งหมดของระบบงานลงใน Repository

## 2. Transform Preliminary Designs

เป็นเครื่องมือที่ใช้ในการเปลี่ยนแปลงการรูปแบบโครงสร้างอย่างง่ายที่ได้จากขั้นตอนการวิเคราะห์ระบบ ให้เป็นการออกแบบเบื้องต้น ซึ่งประกอบด้วย

#### - Database Design Transformer (DDT)

เป็นเครื่องมือที่ใช้ในการจัดการแผนผังการออกแบบฐานข้อมูลที่ได้จากการออกแบบ ER Diagram ตามความต้องการของระบบในขั้นตอนของ Entity Relationship Diagrammer ให้อยู่ในรูปของข้อมูลที่สามารถที่จะเปลี่ยนให้เป็นฐานข้อมูลได้

#### - Application Design Transformer (ADT)

เป็นเครื่องมือที่ใช้ในการจัดการข้อมูลที่ได้จากขั้นตอนของ Function Hierarchy Diagrammer และ Dataflow Diagrammer เพื่อให้อยู่ในรูปแบบที่สามารถทำการสร้างเป็นแอปพลิเคชันที่ตั้งอยู่บนพื้นฐานของฟังก์ชัน และ Business Process ที่ได้จากความต้องการของระบบได้

## 3. Design and Generate

กลุ่มเครื่องมือที่ใช้ในการออกแบบแนวทางของระบบให้เป็นฟอร์ม (Form) ที่สอดคล้องและตรงตามความต้องการขององค์กรสำหรับผู้ที่ทำหน้าที่จัดการระบบและผู้ที่ออกแบบระบบ โดยมีการสร้างองค์ประกอบต่างๆในฝั่งของเซิร์ฟเวอร์ และแอปพลิเคชัน

ในฝั่งของ ไคลเอนต์จากรายละเอียดเรคอร์ดที่อยู่ภายใน Repository เครื่องมือที่ทำหน้าที่ในส่วนนี้ คือ

**- Design Editor**

เป็นเครื่องมือที่ให้สภาพแวดล้อมในการออกแบบและการสร้างแอปพลิเคชันของทั้งฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ รวมไว้ในสภาพแวดล้อมเดียวกัน ใช้สำหรับทั้งการออกฐานข้อมูล และการออกแบบแอปพลิเคชัน

- **Generators** ใช้ในการสร้างส่วนต่างๆของระบบ โดยแบ่งเป็น 2 ส่วน คือ

- **Server Generators** สามารถสร้างฐานข้อมูลของออราเคิล, ODBC และฐานข้อมูลประเภทอื่นได้

- **Client Generators** สามารถสร้างแอปพลิเคชันได้ในหลายแพลตฟอร์ม ดังนี้

\* Oracle Forms และ Oracle Reports

\* Oracle Web PL/SQL Application

\* Visual Basic Project

\* MS Help Application

- **Utilities and Services**

นอกเหนือจาก Client tools และ Repository แล้ว Oracle Designer 6i ยังสนับสนุน Utilities และ Services ต่างๆ ได้แก่

- **Repository Services** ประกอบด้วย

\* **Repository Object Navigator** เป็นเครื่องมือที่ทำให้สามารถเข้าถึงออบเจกต์ทั้งหมดใน Repository ใช้ในการจัดเก็บและจัดการออบเจกต์ทั้งหมดที่มีในการพัฒนาระบบ และมีการจัดเก็บลงใน Repository โดยจะมีการใช้งานร่วมกับ Repository Administration Utility เพื่อใช้ในการดูแลความสัมพันธ์ทั้งหมดที่อยู่ภายในและการเข้าถึงออบเจกต์ของคลังข้อมูลส่วนกลาง นอกจากนี้ผู้ดูแลระบบยังใช้ Repository Object Navigator สำหรับควบคุมและกำหนดสิทธิ์ในการเข้าถึงระบบด้วย

\* **Matrix Diagrammer** เป็นเครื่องมือที่ใช้ทำ Cross-check ระหว่างอีลิเมนต์ที่มีรูปแบบต่างกัน ได้ เช่น สามารถทำ Cross-check ระหว่าง Business Functions และ Entities

\* **Repository Reports** เป็นเครื่องมือที่ใช้สำหรับสร้างรายงาน (Report) ของข้อมูลใน Repository

\* **Repository Administration Utility** ใช้สำหรับการสร้างและดูแล Repository รวมถึงการควบคุมการเข้าถึง Repository

- **Online Documentation** เป็นเอกสารอธิบายไปเกี่ยวกับวิธีการใช้และหลักการทำงานของ Oracle Designer 6i

### 2.3.1.2 การทำงานฝั่งเซิร์ฟเวอร์

มีส่วนประกอบหลักในการจัดการอยู่ 2 ส่วน คือ

#### 1) Repository

เปรียบเสมือนเป็นหัวใจของ Oracle Designer เนื่องจาก Repository อำนวยความสะดวกในการออกแบบและสร้างแอปพลิเคชันที่รวดเร็วและมีประสิทธิภาพ

Oracle Repository เป็นชุดเครื่องมือสำหรับเก็บและจัดการ Meta Data ของแอปพลิเคชัน ซึ่งสนับสนุนการทำงานในสภาพแวดล้อมแบบ Multi-user ทำให้สนับสนุนการพัฒนาแบบเป็นทีม เพราะสามารถเข้าถึง Repository ร่วมกันได้ นอกจากนี้เมื่อมีการจัดการเกิดขึ้นกับอิลิเมนต์ใดก็ตามใน Repository จะไม่เกิดการเปลี่ยนแปลงกับ Definition ของอิลิเมนต์นั้นจริงๆ การเปลี่ยนแปลงจะเป็นแบบ Logical ไม่ใช่ Physical ผู้ที่ทำหน้าที่สร้าง, ดูแล และกำหนดสิทธิ์ในการเข้าถึง Repository เรียกว่า Repository Administration

#### 2) API (Application Programmatic interface)

เป็นส่วนที่ทำหน้าที่ในการ Update Repository ซึ่งจะ Update ทันที เมื่อมีการเปลี่ยนแปลงและแก้ไขข้อมูลใดๆผ่าน Tools ต่างๆของ Oracle Designer 6i