บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ในอดีตข้อกำหนด (Specification) มีลักษณะเป็นคำบรรยายถึงกระบวนการทำงานของระบบ ซึ่งวิธีนี้ทำให้ยากแก่การทำความเข้าใจ และมีปัญหาในการตีความระหว่างผู้ใช้กับนักวิเคราะห์ระบบ อีกทั้งข้อกำหนดนี้ยังยากแก่การแก้ไขเนื่องจากความซับซ้อน ปัจจุบันการพัฒนาเทคโนโลยีทางด้านคอมพิวเตอร์เป็นไปอย่างรวดเร็ว ทำให้คอมพิวเตอร์มีความสามารถมากขึ้นนำไปสู่การพัฒนาระบบงานที่มีความสลับซับซ้อนมากขึ้นด้วย ดังนั้นวิธีการวิเคราะห์ระบบและข้อกำหนดแบบเดิมไม่สามารถตอบสนองความต้องการใหม่นี้ได้อย่างคล่องตัว ในช่วงสิบปีที่ผ่านมาวิธีการวิเคราะห์ระบบแบบโครงสร้างเริ่มเป็นที่ยอมรับกันอย่างแพร่หลาย (Yourdon, 1989)

การวิเคราะห์ด้วยวิธีนี้ได้เสนอแนะแนวทางการเขียนข้อกำหนดขึ้นใหม่เพื่อแก้ปัญหาที่มีอยู่ โดยข้อกำหนดใหม่มีคุณสมบัติดังนี้

1. มีลักษณะเป็นกราฟิก (Graphic) ซึ่งประกอบด้วยแผนภาพแบบต่าง ๆ ทำให้ผู้ใช้เข้าใจง่ายและลดปัญหาในการตีความ

2. ข้อกำหนดแบ่งออกเป็นหลายระดับ (Leveling) ทำให้นักวิเคราะห์ระบบสามารถเข้าใจขอบเขตของงานโดยดูจากข้อกำหนดระดับบน และเข้าใจรายละเอียดมากขึ้นเมื่อดูข้อกำหนดในระดับล่างลงมา และยังสามารถแบ่งงานเป็นส่วน ๆ ได้

การเขียนข้อกำหนดด้วยวิธีนี้ ก่อให้เกิดประโยชน์ดังต่อไปนี้

1. ทำความเข้าใจได้ง่าย เนื่องจากเป็นรูปภาพ
2. ลดปัญหาการตีความ เนื่องจากสัญลักษณ์มีความหมายที่แน่นอน
3. สามารถแก้ไขได้ง่าย เนื่องจากแบ่งข้อกำหนดเป็นหลายระดับและเป็นส่วน ๆ

ในการวิเคราะห์ข้อมูลแบบโครงสร้าง มักจะใช้เครื่องมือดังต่อไปนี้ (Bellin, 1990)

1. แผนภาพกระแสข้อมูล (Data Flow Diagram) เป็นแผนภาพที่แสดงถึงกระบวนการเปลี่ยนแปลงของข้อมูล

2. พจนานุกรมข้อมูล (Data Dictionary) เป็นแหล่งเก็บข้อมูลที่ได้จากการวิเคราะห์ระบบ ตลอดจนคำอธิบายคุณสมบัติของข้อมูลเหล่านั้น

3. ข้อกำหนดของกระบวนการ (Process Specification) แสดงถึงกระบวนการเปลี่ยนแปลงของข้อมูล ซึ่งอาจเขียนในรูปของรหัสเทียม (Pseudo Code)

4. แบบจำลองข้อมูล (Data Model) แสดงถึงความสัมพันธ์ของข้อมูล

5. แผนภาพโครงสร้าง (Structure Chart) เป็นแผนภาพที่แสดงถึงลำดับชั้นของโมดูล (Hierarchy of Module) ของระบบ

เครื่องมือเหล่านี้หากทำด้วยมือจะมีความยุ่งยากในการแก้ไข และถ้าระบบมีขนาดใหญ่และซับซ้อนก็จะทำให้มีความยุ่งยากยิ่งขึ้น ดังนั้นจึงได้มีการนำคอมพิวเตอร์มาช่วยในการพัฒนาระบบ (Computer-Aided Software Engineering, CASE) เอียน ซอมเมอร์วิลล์ ได้แบ่งการทำงานของซอฟต์แวร์เคสออกเป็นระบบย่อย 8 ระบบดังแสดงในรูปที่ 1.1 (Sommerview, 1989) ระบบย่อยนี้อาศัยฐานข้อมูลกลางเป็นตัวเชื่อม โดยรายละเอียดแต่ละระบบเป็น ดังนี้

1. โปรแกรมบรรณาธิการแผนภาพและผังงาน (Diagram Editing Tool) เช่น แผนภาพกระแสข้อมูล, แผนภาพโครงสร้าง เป็นต้น โปรแกรมบรรณาธิการนี้ไม่ใช่โปรแกรมกราฟิกหรือโปรแกรมแคด (Computer-Aided Design) ธรรมดา ๆ เพราะจะต้องทราบประเภทและคุณลักษณะของเอนทิตีที่แสดงด้วยภาพได้ เช่น แผนภาพกระแสข้อมูลรูปหนึ่ง แสดงการส่งข้อมูลจากกระบวนการ A ไปยังกระบวนการ B โดยใช้ลูกศรแสดงทางเดินของข้อมูล ถ้าหากลบกระบวนการ A ออกจากแผนภาพ ลูกศรแสดงการส่งข้อมูลนี้จะต้องถูกลบออกโดยอัตโนมัติด้วย

2. เครื่องมือวิเคราะห์และตรวจสอบการออกแบบ (Design and Checking Tools) เป็นเครื่องมือสำหรับตรวจสอบ และรายงานความผิดพลาด ความคลาดเคลื่อนในการออกแบบ โดยปกติมักจัดทำเป็นส่วนหนึ่งของโปรแกรมบรรณาธิการ

3. เครื่องมือภาษาสอบถาม (Query Langauge Facilities) ใช้สำหรับค้นหาข้อมูลที่บันทึกเก็บไว้และตรวจสอบแผนภาพที่ทำเสร็จแล้ว

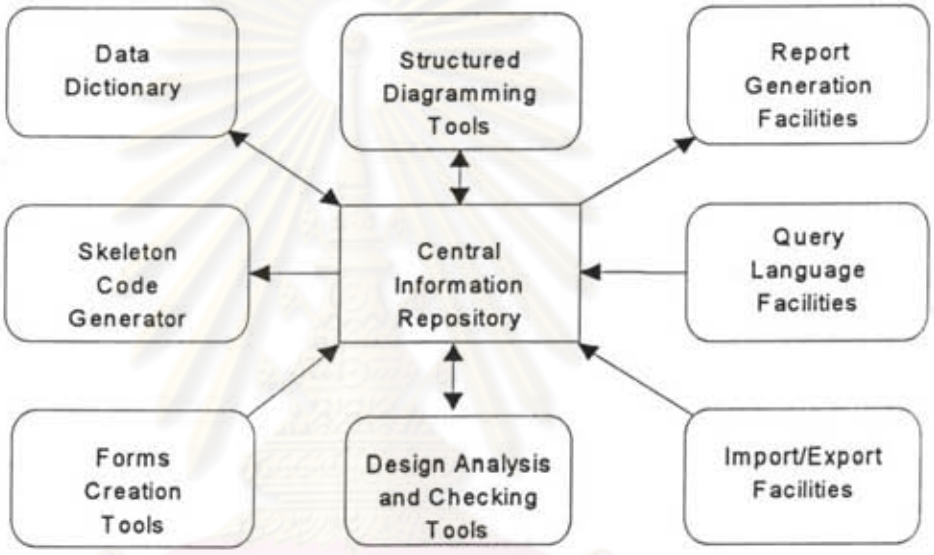
4. พจนานุกรมข้อมูล (Data Dictionary) ใช้สำหรับเก็บรายละเอียดเกี่ยวกับชื่อต่าง ๆ ที่ใช้ในระบบที่พัฒนา

5. เครื่องมือสร้างรายงาน (Report Generation Facilities) ใช้สำหรับนำข้อมูลที่เก็บไว้ที่ศูนย์กลางมาจัดทำเอกสารของระบบ เช่น ข้อกำหนด ได้โดยอัตโนมัติ

6. เครื่องมือสร้างแบบฟอร์ม (Forms Generation Tools) ใช้สำหรับกำหนดรูปแบบของจอภาพและเอกสารต่าง ๆ ของระบบ

7. เครื่องมือรับส่ง (Import/Export Facilities) ใช้สำหรับแลกเปลี่ยนข้อมูลกับเครื่องมือพัฒนาซอฟต์แวร์อื่น ๆ เช่น สามารถส่งโครงสร้างข้อมูลสำหรับใช้ในภาษาชั้นสูง เช่น C, COBOL, BASIC เป็นต้น

8. เครื่องมือสร้างโครงคำสั่ง (Skeleton Code Generator) ใช้สำหรับสร้างโครงคำสั่งตามแบบที่ออกแบบไว้



รูปที่ 1.1 แสดงการทำงานของซอฟต์แวร์เคส (Sommerview, 1989)

ตาราง 1.1 เป็นตัวอย่างของซอฟต์แวร์เคสที่ช่วยในการวิเคราะห์และออกแบบข้อมูล ในปัจจุบันซอฟต์แวร์เคสยังไม่เป็นที่แพร่หลายมากในประเทศไทยนัก ซึ่งเป็นผลมาจาก

- มีราคาสูงมากเมื่อเทียบกับโปรแกรมสำเร็จรูปทั่ว ๆ ไป จากตาราง 1.1 จะพบว่าซอฟต์แวร์เคสมีราคาตั้งแต่ $2,495 - $250,000 เมื่อเทียบกับ CA-Clipper 5.2 มีราคาเพียง $199 หรือโปรแกรมแคดซึ่งมีราคาประมาณ $2,000 - $5,000 ซึ่งนับได้ว่ามีราคาสูงแล้ว แต่เมื่อเทียบกับซอฟต์แวร์เคสจะเห็นได้ว่ามีราคาสูงกว่ามาก

- บริษัทผู้ผลิตไม่ยอมส่งมาขายในประเทศไทย เนื่องจากปัจจุบันยังไม่มีกฎหมายคุ้มครองลิขสิทธิ์ซอฟต์แวร์ ผู้ผลิตกลัวการละเมิดลิขสิทธิ์จากการทำสำเนาซอฟต์แวร์ออกจ่ายแจก

- อุตสาหกรรมซอฟต์แวร์ในประเทศไทยยังไม่ขยายตัวมากนัก เนื่องจากปัญหาลิขสิทธิ์ทำให้ผู้ผลิตซอฟต์แวร์ไม่กล้าเสี่ยงที่จะผลิตซอฟต์แวร์ออกมา

- ขาดบุคคลากรที่มีความรู้ทางด้านวิศวกรรมซอฟต์แวร์

ตาราง 1.1 ตารางซอฟต์แวร์เคส (Gane, 1990)

Product	Minimum Configuration	graphics	respository	price
Analyst/Designer Toolkit	Run on AT, PS/2 640K, 10MB, mouse Hercules monocrome	DFD, ERD, State Transition Diagram, structure chart	Data element, data structure, process, dataflow, data store, entity, relationship	$2495 for first copy
Backman Product Set	Run on Compaq 386, PS/2 m80 1MB RAM, 20 MB DISK monitem Viking or IBM8514	Backman Entity relationship diagram	DA: entities, attribute,relationship, key,dimension,domain,data type DAB: tables, table space, column, index	$25000 for 1st copy package
Cor Vision (diagramer tool)	Run on workstation PC, PS/2, AT, VT terminal 640K, 20MB, mouse, CGA	ERD, Menu diag, dataview diagram	Store on VAX;diagram, source code, system documentation	$250K for development license
Deft	MacPlus, WorkStation	DEF, ERD, Jackson structure diagram, form and report layout	all diagram object	Macintosh: $9000 for 1st copy VAX:$10000-$52000 depend on size
Designv1	Run on XT,AT 512K, 2 floppies	DFD, ERD, flowchart, Warnier-Orr diagram	All diagram Object, data element/structure, entities,relationship, screen/report description	$7000 for first copy
THE DEVELOPER	(PC workstation) XT / AT 3270 PC, 286, 386 (mainframe) MVS/TSO with DB2 or ORACLE VAX/VMS with ORACLE	DFD, ERD, Structure chart, Oranization chart, Operation Procedure diag., system flowchart	50 (called components: data element, data flow, process, etc.) and 22 relationship type provided as standard with the CUSTOMIZER	$5400 for first copy Mainframe repository $30000-$50000
ER-DESIGNER (ERD)	Run on XT / AT 320K, 2 floppies	ERD (chen notation)	Entities, entities attribute, relationship, relationship attribute, connection, cardinality	$495 for first copy
Excelerator	Run on XT / AT / VAX / SUN / APOLLO 640K, 10MB, mouse, EGA	DFD, ERD, Structure chart, Document graph, Presentation diagram, state-transition diag.	45 (data store, records, element, process, entity, relationship,etc)	$8400 for first copy

การวิจัยนี้จะเป็นการศึกษาเทคนิคในการสร้างแบบจำลองเชิงตรรกะโดยใช้แผนภาพกระแสข้อมูล และวิธีการของซอฟต์แวร์เคส เพื่อจัดสร้างโปรแกรมช่วยในการสร้างแบบจำลองเชิงตรรกะ เพื่อใช้เป็นข้อกำหนดในการออกแบบและพัฒนาโปรแกรม

ทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ มีดังต่อไปนี้

- แผนภาพกระแสข้อมูล
- ขั้นตอนการทำแบบจำลองเชิงตรรกะ
- เทคนิคการเขียนโปรแกรมประยุกต์บนวินโดว์ด้วยวิธีโปรแกรมเชิงวัตถุ

วัตถุประสงค์ของการวิจัย

1. สร้างเครื่องมือช่วยนักวิเคราะห์ระบบให้ทำงานได้สะดวกขึ้น โดยเครื่องมือที่สร้างขึ้นสามารถทำงานได้ดังนี้
   - เขียนแผนภาพกระแสข้อมูล
   - จัดเก็บรายละเอียดของข้อกำหนด
   - จัดพิมพ์แผนภาพกระแสข้อมูล และรายละเอียดของข้อกำหนด
2. เพื่อเป็นแนวทางในการวิจัยและพัฒนาซอฟต์แวร์เคส ซึ่งเป็นการรวมเครื่องมือซอฟต์แวร์ (Software Tools) เพื่อช่วยในการพัฒนาโปรแกรมแบบครบวงจร (Full-Life Cycle) ต่อไป

ขอบเขตของการวิจัย

1. พัฒนาบนเครื่องไมโครคอมพิวเตอร์ภายใต้โปรแกรมไมโครซอฟต์วินโดว์รุ่น 3.0 ขึ้นไป
2. การพัฒนาครั้งนี้ไม่รวมการแสดงผลภาษาไทย ถ้าต้องการแสดงผลภาษาไทยต้องใช้โปรแกรมไมโครซอฟต์วินโดว์ที่แสดงผลภาษาไทยได้
3. ใช้แผนภาพกระแสข้อมูลในการสร้างแบบจำลองเชิงตรรกะ (Logical Model)
4. สามารถตรวจสอบความครบถ้วนของการไหลข้อมูลระหว่างแผนภาพกระแสข้อมูลแต่ละระดับเท่านั้น
5. การเขียนข้อกำหนดของกระบวนการสามารถทำในลักษณะคำอธิบายเท่านั้น โดยใช้โปรแกรมบรรณาธิกร

ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีการสร้างแบบจำลองเชิงตรรกะ
2. ศึกษาความรู้ทางวินโดว์
3. ศึกษาภาษาซี / ซีพลัสพลัส และแนวทางการเขียนโปรแกรมไมโครซอฟต์วินโดว์
4. ออกแบบระบบ
   - ออกแบบตัวประสานกับผู้ใช้
   - ออกแบบวัตถุ

5. เขียนโปรแกรม

6. ทดสอบและประเมินผลการทำงานของโปรแกรม

7. สรุปผล เสนอแนะ และจัดทำรูปเล่มวิทยานิพนธ์

ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นเครื่องมือช่วยในการวิเคราะห์ระบบ

2. เป็นแนวทางในการวิจัยและพัฒนาซอฟต์แวร์ประเภทเคส ซึ่งเป็นการรวมเครื่องมือซอฟต์แวร์ (Software Tools) เพื่อช่วยในการพัฒนาโปรแกรมแบบครบวงจร (Full-Life Cycle) ต่อไป

3. ช่วยประหยัดเวลาและค่าใช้จ่ายในการวิเคราะห์ระบบ

4. ช่วยพัฒนาอุตสาหกรรมการผลิตซอฟต์แวร์

5. ช่วยลดการขาดดุลการค้ากับต่างประเทศ