บทที่ 2

ทฤษฎีและแนวความคิดที่นำมาใช้ในการวิจัย

ในการจัดสร้างโปรแกรมช่วยการพัฒนาแบบจำลองเชิงตรรกะ โดยในการวิจัยครั้งนี้ใช้แผนภาพกระแสข้อมูลในการทำแบบจำลองเชิงตรรกะ และเพื่อให้นักวิเคราะห์ระบบสามารถใช้โปรแกรมได้ง่ายจึงได้ทำการพัฒนาโปรแกรมให้ทำงานภายใต้วินโดว์ ดังนั้นผู้วิจัยจึงทำการศึกษาทฤษฎีการพัฒนาแบบจำลองเชิงตรรกะโดยใช้แผนภาพกระแสข้อมูล และเทคนิคการเขียนโปรแกรมบนวินโดว์ เพื่อที่จะใช้ในการพัฒนาโปรแกรม

แบบจำลองเชิงตรรกะ

การพัฒนาแบบจำลองเชิงตรรกะเป็นวิธีแสดงการทำงานของระบบงาน โดยไม่ต้องคำนึงถึงอุปกรณ์ที่ใช้งาน และเทคนิคของการเขียนโปรแกรม เพื่อให้แบบจำลองที่สร้างขึ้นแสดงถึงขั้นตอนการทำงานของระบบอย่างแท้จริง นักวิเคราะห์ระบบจะได้มีอิสระในการเลือกอุปกรณ์ในขั้นตอนการออกแบบระบบ ในวิทยานิพนธ์นี้ใช้แผนภาพกระแสข้อมูลในการสร้างแบบจำลองเชิงตรรกะ โดยมีวัตถุประสงค์ของระบบและรายการรายงานเป็นเครื่องมือใช้ในการตรวจสอบความถูกต้องของแบบจำลอง เพื่อที่จะได้ระบบงานที่ทำงานตามความต้องการของผู้ใช้ นอกจากนี้ยังมีข้อกำหนดของเอนทิตีที่อยู่ในแผนภาพ ซึ่งจะแสดงรายละเอียดของเอนทิตีแต่ละตัวที่อยู่ในแผนภาพ ซึ่งจะช่วยให้แบบจำลองมีความสมบูรณ์มากยิ่งขึ้น

1. แผนภาพกระแสข้อมูล

แผนภาพกระแสข้อมูล คือแผนภาพที่แสดงการเคลื่อนไหวของข้อมูลจากกระบวนการหนึ่งไปยังอีกกระบวนการหนึ่ง และกระบวนการแต่ละกระบวนการรับข้อมูลอะไรเข้าไป และประมวลผลให้ข้อมูลอะไรออกมา แผนภาพกระแสข้อมูลถูกแนะนำขึ้นมาใช้ประมาณปี 1974 - 1975 เพื่อใช้ในการออกแบบระบบโดยยืมเครื่องหมายมาจากทฤษฎีของกราฟ และต่อมาวิศวกรซอฟต์แวร์ได้นำมาใช้ในการทำแบบจำลองความต้องการของผู้ใช้ (Yourdon, 1989)

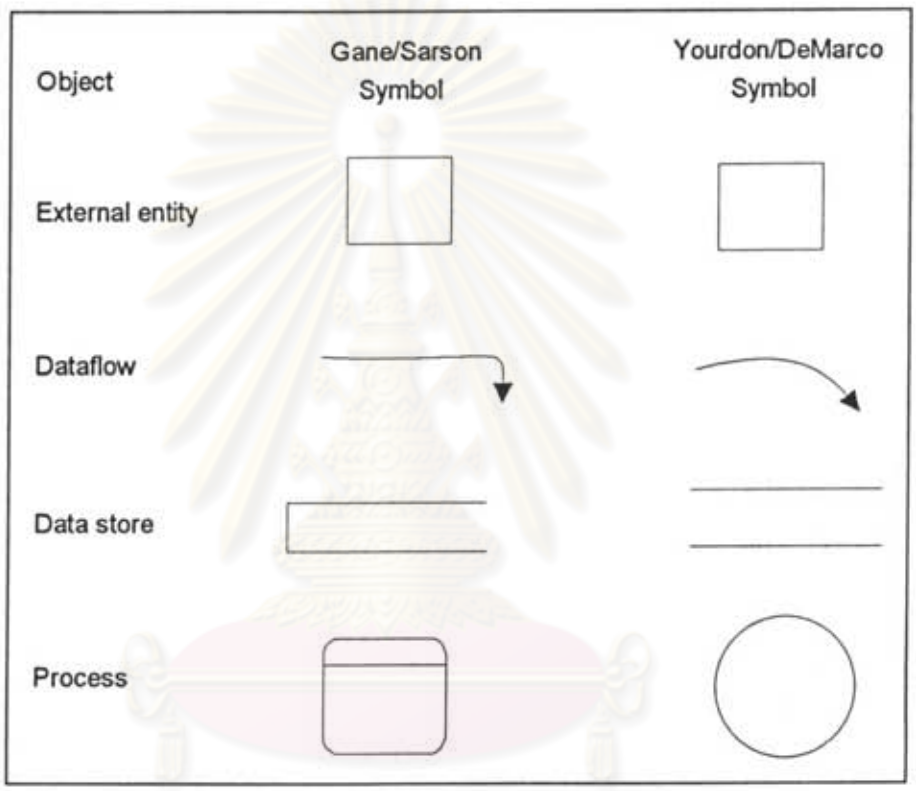
สัญลักษณ์ที่ใช้ในแผนภาพกระแสข้อมูล มีดังต่อไปนี้ (Gane, 1990)

- เอนทิตีภายนอก (External Entity) เรียกอีกอย่างว่า Source/Sink, Source/Destination เป็นแหล่งกำเนิดข้อมูล หรือจุดสิ้นสุดของข้อมูลของระบบ ซึ่งอาจจะเป็นบุคคล กลุ่มบุคคล หรือระบบงานก็ได้ เช่น ระบบเงินเดือนส่งรายงานเงินเดือนที่ต้องจ่ายให้กับระบบบัญชี ระบบบัญชีก็จะเป็นเอนทิตีภายนอกของระบบเงินเดือน เป็นต้น สัญลักษณ์ที่ใช้แทนเอนทิตีภายนอกใช้รูปสี่เหลี่ยมมีชื่ออยู่ภายใน
- กระแสข้อมูล (Data Flow) ใช้แสดงทิศทางการเคลื่อนที่ของข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยทิศทางที่ข้อมูลเคลื่อนที่จะไปในทิศทางเดียวกับลูกศร กระแสข้อมูลจะไม่แสดงลำดับการไหลของข้อมูล ลักษณะของกระแสข้อมูลเปรียบได้กับรางรถไฟ ซึ่งแสดงทิศทางการเดินรถแต่ไม่ได้บอกเวลารถไฟวิ่งออกจากสถานี
- กระบวนการ (Process) ใช้แสดงกริยาการกระทำต่อข้อมูลที่ไหลเข้ามา โดยไม่คำนึงว่าจะเป็นการกระทำด้วยคนหรือเครื่องก็ตาม และได้มาซึ่งผลลัพธ์ที่จะไหลออกจากกระบวนการ สัญลักษณ์ที่ใช้แทนกระบวนการใช้รูปวงกลมมีชื่ออยู่ภายใน การตั้งชื่อกระบวนการควรจะตั้งในลักษณะ "กริยา + กรรม"
- ส่วนเก็บข้อมูล (Data Store) ใช้แสดงที่เก็บข้อมูล ซึ่งอาจเก็บอยู่ที่ไหนก็ได้ เช่น ในแผ่นดิสก์เทป หรือแฟ้มเอกสาร เป็นต้น ถ้าส่วนเก็บข้อมูลมีกระแสข้อมูลเข้าและกระแสข้อมูลออกอย่างละ 1 เส้น ต้องพิจารณาว่าส่วนเก็บข้อมูลนี้มีความสำคัญมากน้อยแค่ไหน หรือส่วนเก็บข้อมูลนี้เป็นเพียงที่พักข้อมูลชั่วคราว เช่น สาขาส่งรายงานการขายให้สำนักงานใหญ่โดยใช้แผ่นดิสก์ แผ่นดิสก์นี้ไม่เป็นแหล่งเก็บข้อมูลม เป็นเพียงที่พักชั่วคราว ซึ่งในอนาคตบริษัทอาจเปลี่ยนรูปแบบการส่งข้อมูลโดยใช้โมเด็มก็ได้ ดังนั้นจึงไม่จำเป็นต้องมีส่วนเก็บข้อมูลนี้ในแผนภาพ แต่ถ้าในกรณีที่สาขาส่งรายละเอียดของลูกค้าที่มีปัญหาให้กับผู้จัดการที่สำนักงานใหญ่ ข้อมูลนี้มีความสำคัญในระบบงาน ก็อาจเขียนส่วนเก็บข้อมูลนี้ไว้ในแผนภาพก็ได้ สัญลักษณ์ที่ใช้แทนส่วนเก็บข้อมูลใช้เส้นขนานโดยมีชื่ออยู่ตรงกลาง

สัญลักษณ์ที่ใช้อธิบายในตอนต้นเป็นแบบ Yourdon / DeMarco ยังมีสัญลักษณ์อีกแบบที่นิยมใช้กันคือแบบ Gane / Sarson สัญลักษณ์ที่ใช้ทั้ง 2 แบบแสดงไว้ในรูปที่ 2.1

การเขียนแผนภาพกระแสข้อมูลจะเขียนแผนภาพเป็นหลายระดับ โดยระดับบนสุดจะแสดงขอบเขตของระบบงาน และจะแสดงรายละเอียดของกระบวนการโดยการเขียนแผนภาพกระแสข้อมูลระดับล่างลงมา การเขียนแผนภาพโดยแบ่งเป็นระดับและเป็นส่วน ๆ นี้ทำให้สามารถทำความเข้าใจระบบงานได้ง่าย ผู้ที่เข้ามาศึกษาภายหลังสามารถทำการศึกษาขอบเขตงานได้จากแผนภาพระดับบน และเข้าใจระบบมากขึ้น

เมื่อศึกษาแผนภาพในระดับล่าง และยังสามารถเลือกศึกษาเฉพาะส่วนที่ต้องการได้ การศึกษาแผนภาพกระแสข้อมูลสามารถทำความเข้าใจได้ง่ายเนื่องจากใช้สัญลักษณ์เพียง 4 แบบและสัญลักษณ์ที่ใช้สามารถทำความเข้าใจได้ง่าย ดังนั้นบุคคลที่ไม่มีความรู้ทางด้านคอมพิวเตอร์สามารถเรียนรู้และเขียนแผนภาพเองได้ นอกจากนี้แผนภาพกระแสข้อมูลไม่ได้แสดงอุปกรณ์ที่ใช้ในระบบ ทำให้นักวิเคราะห์ระบบมีอิสระในการออกแบบระบบ (Gane, 1990)



รูปที่ 2.1 แสดงสัญลักษณ์ที่ใช้ในแผนภาพกระแสข้อมูล (Gane,1990)

2. วัตถุประสงค์ของระบบ

วัตถุประสงค์ของระบบ เป็นเอกสารที่จะใช้กำหนดขอบเขตความรับผิดชอบของระบบ ซึ่งจะใช้เป็นตัวควบคุมให้การพัฒนาระบบเป็นไปตามแนวทางที่ต้องการ เป็นข้อตกลงของทุกฝ่ายที่ช่วยกันกำหนด ถ้าไม่สามารถเขียนวัตถุประสงค์ของระบบได้แสดงว่าแต่ละฝ่ายไม่ทราบจุดประสงค์ของระบบนี้อย่างชัดเจน ซึ่งอาจทำให้เกิดความขัดแย้งกันในอนาคต ดังนั้นการเขียนวัตถุประสงค์ของระบบจึงมีความสำคัญมาก

การเขียนวัตถุประสงค์ของระบบเริ่มต้นโดยการกำหนดชื่อระบบงานนั้น เขียนเหตุผลที่ต้องการระบบงานนี้ จากนั้นจึงเขียนหน้าที่ต่าง ๆ ที่ระบบจะต้องทำ และถ้ามีงานใดดูเหมือนจะรวมอยู่ในระบบงานนี้ แต่ได้มีการตกลงให้ไม่อยู่ในขอบเขตของงานนี้ ก็ให้ระบุไว้ในวัตถุประสงค์ของระบบเพื่อที่จะได้ไม่เกิดความ

ขัดแย้งในภายหลัง วัตถุประสงค์ของระบบควรมีความยาวไม่มากเกินไป ควรจะสามารถทบทวนได้ภายใน 1 ชั่วโมง รูปที่ 2.2 เป็นตัวอย่างการเขียนวัตถุประสงค์ของระบบของระบบควบคุมกล้องถ่ายรูปที่ติดตั้งอยู่บนดาวเทียม (Yourdon Systems Method Model-Driven System Development, 1993)

SYSTEM : Satellite Camera Control System

GENERAL DESCRIPTION :

The purpose of the system is to take and display pictures of land resources from a satellite. The satellite will contain an automatic electronic camera that will (when activated) collect light untill the exposure is complete. The image may then be read out before sending a signal to start another exposure.

RESPONSIBILITIES :

1. Accepting requests for photographs from accredited researchers and scheduling a suitable time for these photographs to be taken
2. Controlling operation of the camera and storing pictures.
3. Using the satellite attitude control instrumentation to point the camera at the target.
4. Display picture on request
5. Keeping an up to date record of satellite's orbit so scheduling can be achived. To do this an on-borad accelerometer is used, together with ground tracking information

SPECIFIC EXCLUSIONS :

1. Chaging the satellite orbit.
2. Tracking the satellite. This function is carried out by ground tracking stations.

รูปที่ 2.2 ตัวอย่างของวัตถุประสงค์ระบบ (Yourdon Systems Method Model-Driven System Development, 1993)

3. ข้อกำหนดของเอนทิตีในแผนภาพกระแสข้อมูล

3.1 ข้อกำหนดของเอนทิตีภายนอก เป็นข้อกำหนดแสดงรายละเอียดของเอนทิตีภายนอก ซึ่งประกอบด้วยข้อมูล ชื่อของเอนทิตีภายนอก (Name) คำอธิบายเอนทิตีภายนอก (Description) ใช้แสดงลักษณะและบทบาทของเอนทิตีภายนอกในระบบงานนี้ จำนวนเอนทิตี (Number of Instances) ใช้แสดงจำนวนเอนทิตีในระบบ ถ้ามีจำนวนหลายตัวจะต้องมีตัวระบุ (Identifier) รูป 2.3 เป็นตัวอย่างของข้อกำหนดของเอนทิตีภายนอก (Yourdon Systems Method Model-Driven System Development, 1993)

NAME : Lift

Description : The System controls the movement of lifts.
Each lift may be moved up or down by a motor.

Number of instances : 4

Identifier : <Lift>.number

NAME : Attending Physician

Description : A fully licensed MD employed by or practicing at this hospital.

Number of instances : many

Identifier : <Physician>.id

รูปที่ 2.3 ตัวอย่างข้อกำหนดของเอนทิตีภายนอก (Yourdon Systems Method Model-Driven System Development, 1993)

3.2 ข้อกำหนดของกระบวนการ เป็นข้อกำหนดแสดงรายละเอียดของกระบวนการ ซึ่งประกอบด้วย ชื่อกระบวนการ (Name) คำอธิบายกระบวนการ (Description) ใช้อธิบายวิธีการเปลี่ยนแปลงข้อมูล ซึ่งอาจจะอธิบายในรูปของรหัสเทียม (Pseudo Code) ตารางการตัดสินใจ (Decision Table) รูปที่ 2.4 เป็นตัวอย่างของข้อกำหนดกระบวนการ โดยเขียนคำอธิบายกระบวนการในรูปรหัสเทียม (Yourdon Systems Method Model-Driven System Development, 1993)

3.3 ข้อกำหนดของส่วนเก็บข้อมูล ใช้แสดงเอนทิตีหรือความสัมพันธ์ของข้อมูลที่รวมอยู่ในส่วนเก็บข้อมูลนี้ ซึ่งใช้ในการตรวจสอบซึ่งกันและกันระหว่างแผนภาพกระแสข้อมูลกับแบบจำลองข้อมูล ข้อมูล

ที่เก็บในส่วนเก็บข้อมูลประกอบด้วย ชื่อของส่วนเก็บข้อมูล (Name) คำอธิบายส่วนเก็บข้อมูล (Description) ใช้แสดงเอนทิตีที่อยู่ในส่วนเก็บข้อมูลนี้ ความสัมพันธ์ของข้อมูล (Relationship) .ใช้แสดงความสัมพันธ์ของข้อมูล ในกรณีข้อมูลที่เก็บในส่วนเก็บข้อมูลเป็นความสัมพันธ์ของข้อมูล เช่น <Instructor> is assigned to teach <Scheduled corse> และ กลุ่มของข้อมูล (Stores Included) ใช้แสดงกลุ่มของข้อมูล ในกรณีที่มีจำนวนเอนทิตีมาก และเอนทิตีสามารถแบ่งออกได้เป็นกลุ่ม ๆ และแต่ละกลุ่มได้มีการอธิบายรายละเอียดในข้อกำหนดอื่นแล้ว สามารถใช้ชื่อกลุ่มของเอนทิตีแทนเอนทิตีเหล่านั้นได้ (Yourdon Systems Method Model-Driven System Development, 1993)

```
PROCESS : Caculate Discount
DESCRIPTION :
        if (Discount Customer )
            if (order > 5000)
                discount = 40%
            else
                discount = 25 %
            endif
        else
            if  (order < 1000)
                discount = 0%
            elseif (order < 3000)
                discount = 15%
            else
                discount = 25 %
            endif
        endif
```

รูปที่ 2.4 ตัวอย่างข้อกำหนดของกระบวนการ

3.4 ข้อกำหนดของกระแสข้อมูล ใช้แสดงรายละเอียดของกระแสข้อมูล ประกอบด้วยข้อมูลชื่อของกระแสข้อมูล (Name) คำอธิบายกระแสข้อมูล (Description) และส่วนประกอบของกระแสข้อมูล

(Composition) การเขียนส่วนประกอบสามารถเขียนโดยใช้เครื่องหมายต่อไปนี้ (Yourdon Systems Method Model-Driven System Development, 1993)

= หมายถึง เท่ากันหรือมีความหมายเดียวกัน

+ หมายถึง เป็นการรวมเข้าด้วยกัน

[*data* | *data*] หมายถึง ให้เลือกในวงเล็บอย่างใดอย่างหนึ่ง

l{ *data* }u หมายถึง ข้อมูลในวงเล็บสามารถมีซ้ำได้ โดย l คือตัวเลขกำหนดจำนวนซ้ำน้อยที่สุดที่ต้องมี ถ้าไม่กำหนดให้มีค่าเท่ากับ 0 และ u คือตัวเลขกำหนดจำนวนซ้ำสูงสุด ถ้าไม่กำหนดให้ถือว่าไม่จำกัดจำนวน

(*data*) หมายถึง ข้อมูลในวงเล็บนั้นอาจใช้หรือไม่ก็ได้

DATA STORE : Solutions
ENTITIES : Solutions
RELATIONSHIPS : -
STORES INCLUDE : -

DATA STORE : Teachng assignments
ENTITIES : -
RELATIONSHIPS : <Instructor> is assigned to teach <Schedualed couse>
STORES INCLUDE : -

DATA STORE : Current solutions
ENTITIES : -
RELATIONSHIPS : -
STORES INCLUDE : Scheduled solutions
Solution in tank

รูปที่ 2.5 แสดงข้อกำหนดของส่วนเก็บข้อมูล (Yourdon Systems Method Model-Driven System Development, 1993)

DATAFLOW : patient admission note

MEANING : information recorded at the time the patient took a room at the hospital.

COMPOSITION : patient name

+ [next of kin | contract person | responsible agency]

+ (referring physician)

+ 1{medical complaint + complaint priority}

รูปที่ 2.6 แสดงตัวอย่างข้อกำหนดของกระแสข้อมูล (Yourdon Systems Method Model-Driven System Development, 1993)

4. ขั้นตอนในการทำแบบจำลองข้อมูลเชิงตรรกะ

การวิเคราะห์ระบบเริ่มตั้งแต่ผู้ใช้ต้องการให้นักวิเคราะห์ระบบเข้าไปช่วยหาวิธีแก้ปัญหาให้ นักวิเคราะห์ระบบต้องเข้าไปศึกษาความต้องการของผู้ใช้ การเข้าไปศึกษาระบบอาจได้จากการสัมภาษณ์ผู้ใช้ หรืออาจได้จากเอกสาร ซึ่งงานในส่วนนี้เป็นศิลปะขึ้นอยู่กับความสามารถของนักวิเคราะห์ระบบ ขั้นตอนต่อไป คือนำข้อมูลที่ได้มาสร้างเป็นแบบจำลองเชิงตรรกะ (Logical Model) โดยการใช้แผนภาพกระแสข้อมูล ซึ่ง พอจะแบ่งเป็นขั้นตอนได้นี้ (Bellin, 1990)

1. สร้างแผนภาพกระแสข้อมูลระดับบนสุด (Context Diagram) แผนภาพนี้ใช้ในการกำหนด ขอบเขตของระบบ และแสดงให้เห็นว่ามีใครเกี่ยวข้องกับระบบบ้าง โดยมีวิธีการเขียนดังนี้

1.1 กำหนดผู้เกี่ยวข้องกับระบบ ผู้ที่เกี่ยวข้องกับระบบหมายถึงบุคคล หรือกลุ่มบุคคล หรือหน่วยงาน ที่ให้ข้อมูลหรือรับข้อมูลจากระบบ เริ่มต้นโดยการเขียนสัญลักษณ์กระบวนการขนาดใหญ่รูปเดียว ตั้งชื่อกระบวนการ จากนั้นเขียนสัญลักษณ์ของเอนทิตีภายนอกรอบ ๆ กระบวนการ ใส่ชื่อสำหรับเอนทิตีภายนอก

1.2 กำหนดเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ทำรายการเหตุการณ์ที่เกิดขึ้นซึ่งเกิดจากผู้ใช้ให้ข้อมูลแก่ระบบ หรือผู้ใช้รับข้อมูลจากระบบ ในขั้นตอนนี้ถ้าไม่แน่ใจว่าเป็นเหตุการณ์หรือไม่ ให้ใส่ไว้ในรายการก่อนโดยจะมีขั้นตรวจสอบในภายหลัง

1.3 กำหนดข้อมูลเข้าและข้อมูลออกจากระบบ จากรายการเหตุการณ์ในข้อที่แล้ว ให้ใส่ข้อมูลที่เข้าและออกจากระบบเนื่องจากเหตุการณ์นี้ในรายการเหตุการณ์ ถ้าเหตุการณ์ไหนไม่มีข้อมูลเข้าหรือออก ให้ตัดเหตุการณ์นั้นจากรายการเหตุการณ์ ถ้ามีข้อมูลเข้าหรือออกจากระบบ แต่ไม่มีเหตุการณ์ในรายการเหตุการณ์ให้ใส่เหตุการณ์ใหม่เข้าไปใน

รายการเหตุการณ์ จากนั้นลากเส้นจากเอนทิตีมายังกระบวนการตามที่แสดงไว้ในรายการเหตุการณ์ จากนั้นใส่ข้อมูลนั้นในพจนานุกรมข้อมูล

1.4 ตรวจสอบความครบถ้วนของแผนภาพ ตรวจสอบแผนภาพกระแสข้อมูลทุก ๆ กระแสข้อมูลมีอยู่ในตารางเหตุการณ์ จะสังเกตุว่าเหตุการณ์สามารถมีกระแสข้อมูลได้หลายเส้น และกระแสข้อมูลอาจใช้ในหลายเหตุการณ์

1.5 เขียนวัตถุประสงค์ของระบบ จุดประสงค์ของการเขียนวัตถุประสงค์ของระบบ เพื่อใช้ตรวจสอบว่างานใดสมควรอยู่ในระบบหรือไม่ และใช้ตรวจสอบวัตถุประสงค์ของระบบตรงกับความต้องการของผู้ใช้หรือไม่ ถ้าไม่ใช่ให้นำแผนภาพกระแสข้อมูลระดับบนสุดมาพิจารณาร่วมกับผู้ใช้ และสรุปวัตถุประสงค์ของระบบใหม่

2. วาดแผนภาพกระแสข้อมูลในระดับต่าง ๆ ให้เขียนขยายรายละเอียดของกระบวนการลงในแผนภาพกระแสข้อมูลระดับล่างลงมา ถ้ากระบวนการไหนยังไม่สามารถเขียนข้อกำหนดได้ แสดงว่ากระบวนการนั้นมีความซับซ้อน ให้เขียนแผนภาพกระแสข้อมูลขยายรายละเอียด การเขียนแผนภาพในระดับต่าง ๆ ให้ตรวจสอบความสมบูรณ์ของแผนภาพตามวิธีเดิม และให้เขียนข้อมูลกระแสข้อมูลลงในพจนานุกรมข้อมูล

3. ตรวจสอบกระแสข้อมูล ทำการตรวจสอบกระแสข้อมูลที่เข้าออกในกระบวนการในแผนภาพกระแสข้อมูลในแต่ละระดับว่าเหมือนกันหรือไม่

4. เขียนรายละเอียดของข้อมูลให้สมบูรณ์ เพิ่มรายละเอียดของข้อมูลที่มีอยู่ในพจนานุกรมให้สมบูรณ์

5. พิจารณาความต้องการรายงาน สำรวจรายงานที่ผู้ใช้ต้องการมีรายงานอะไรบ้าง แต่ละรายงานต้องการข้อมูลอะไร ถ้าแผนภาพกระแสข้อมูลที่ทำขึ้นไม่สามารถจัดทำรายงานที่ต้องการได้ เช่น ขาดข้อมูลที่รายงานต้องการ ก็ให้ทำการแก้ไขแผนภาพ

6. เขียนข้อกำหนดของกระบวนการ (Process Specifications) เขียนบรรยายวิธีการที่ใช้ในการเปลี่ยนแปลงข้อมูลเข้าให้เป็นข้อมูลออก สามารถเขียนได้หลายรูปแบบ เช่น ตารางการตัดสินใจ, รหัสเทียม เป็นต้น

<u>เทคนิคการเขียนโปรแกรมบนวินโดว์</u>

1. ความรู้เบื้องต้นเกี่ยวกับวินโดว์

ในปี ค.ศ. 1983 บริษัทไมโครซอฟต์คอร์ปอเรชันได้ออกวินโดว์รุ่น 1 นับเป็นความพยายามในการสร้างระบบจัดการสภาพแวดล้อมแบบกราฟิกบนเครื่องไมโครคอมพิวเตอร์ ซึ่งในยุคนั้นฮาร์ดแวร์ยังไม่พร้อมที่จะทำงานแบบกราฟิก แต่อย่างไรก็ดีบริษัทไมโครซอฟต์ก็ได้ออกวินโดว์รุ่นใหม่ออกมาเรื่อย ๆ และ

ในปี ค.ศ. 1990 บริษัทฯก็ประสบความสำเร็จเมื่อออกวินโดว์รุ่น 3.0 อันเนื่องมาจากเครื่องไมโครคอมพิวเตอร์มีความสามารถมากขึ้น และมีโปรแกรมประยุกต์ที่ใช้งานบนวินโดว์มากขึ้น ซึ่งเป็นผลมาจากการส่งเสริมผู้ผลิตซอฟต์แวร์ของบริษัทฯ โปรแกรมวินโดว์รุ่นนี้มีปรับปรุงให้มีความสามารถมากขึ้น เช่น สนับสนุนการทำงานกับระบบเครือข่าย รูปแบบการแปลงบิทชนิดใหม่ที่ไม่ขึ้นกับอุปกรณ์ (Device Independent Bitmap : DIB) และสนับสนุนการใช้หน่วยความจำแบบขยาย (Extended Memory) ได้ถึง 16 เมกะไบต์ ถ้าหากทำงานบนซีพียู 80386 ขึ้นไปในภาวะเอนฮานด์แล้ว วินโดว์จะใช้การจัดการหน่วยความจำเสมือน (Virtual Memory) จะทำให้สามารถใช้หน่วยความจำได้มากขึ้น ในปี ค.ศ. 1992 บริษัทฯได้ออกวินโดว์รุ่น 3.1 โดยมีการพัฒนารูปแบบตัวอักษรใหม่เรียกว่า ทรูไทป์ (True Type Font) ซึ่งเป็นตัวอักษรแบบโครงร่าง ทำให้สามารถย่อ-ขยายตัวอักษรได้ตามความต้องการ นอกจากนี้ยังพัฒนาการแลกเปลี่ยนข้อมูลที่เรียกว่า OLE (Object Linked and Embedded) และยังปรับปรุงโปรแกรมส่วนต่าง ๆ ให้มีประสิทธิภาพดียิ่งขึ้น

การทำงานของวินโดว์แบ่งออกเป็น 3 ภาวะ ซึ่งแต่ละภาวะขึ้นกับชนิดและความสามารถของเครื่องคอมพิวเตอร์ที่ใช้งาน เมื่อเรียกใช้งานวินโดว์ วินโดว์จะทำการตรวจสอบเครื่องและจะเลือกทำงานในภาวะที่เหมาะสมกับเครื่องโดยอัตโนมัติ ภาวะต่าง ๆ มีรายละเอียดดังนี้ (จิรพัฒน์, 2536)

- ภาวะจริง (Real Mode) เป็นภาวะที่ใช้งานได้กับวินโดว์รุ่นเก่าที่สุด ทำให้โปรแกรมที่เขียนไว้สำหรับวินโดว์รุ่นเก่าสามารถทำงานในภาวะนี้ได้ เครื่องที่จะทำงานในภาวะนี้ต้องเป็นเครื่องที่มีหน่วยความจำไม่น้อยกว่า 1 เมกะไบต์

- ภาวะมาตรฐาน (Standard Mode) เป็นภาวะปกติในการเรียกใช้วินโดว์ เพราะในภาวะนี้วินโดว์จะใช้หน่วยความจำขยายได้เต็มที่

- ภาวะเอนฮานด์ (Enhanced Mode) เป็นภาวะที่สามารถใช้ความสามารถของซีพียู 80386 ได้เต็มที่ และสามารถใช้หน่วยความจำเสมือนได้ ทำให้โปรแกรมสามารถใช้หน่วยความจำได้มากกว่าที่มีอยู่จริง

1.1 คุณลักษณะเด่นของวินโดว์

- ตัวประสานกับผู้ใช้แบบกราฟิก (Graphic User Interface : GUI) ตัวประสานกับผู้ใช้ใช้รูปภาพในการสื่อความหมาย เช่น สัญรูป (Icon) แถบเลื่อน (Scroll Bar) ทำให้เกิดความสวยงาม มีความน่าใช้ สะดวกและง่ายต่อการเรียนรู้ นอกจากนี้โปรแกรมประยุกต์มีตัวประสานกับผู้ใช้เหมือน ๆ กัน ทำให้เป็นมาตรฐานง่ายแก่การเรียนรู้

- การทำงานแบบหลายภารกิจ (Multitasking) สภาพแวดล้อมของวินโดว์ยินยอมให้ผู้ใช้เรียกใช้โปรแกรมมาทำงานได้หลาย ๆ งานพร้อมกัน โดยเป็นแบบ Nonpreemptive คือโปรแกรม

ประยุกต์แต่ละตัวจะใช้ซีพียูไปจนกว่าจะไม่ต้องการ แล้วจึงผลัดให้โปรแกรมอื่นใช้งานต่อ ดังนั้นในการเขียนโปรแกรมจะต้องระวังในข้อนี้ด้วย

- ความเป็นอิสระต่ออุปกรณ์ เนื่องจากวินโดว์เป็นตัวกลางระหว่างโปรแกรมที่เขียนขึ้นกับอุปกรณ์ต่าง ๆ เพียงแต่เรียกใช้ฟังก์ชันต่าง ๆ ที่วินโดว์จัดเตรียมให้ ทำให้ไม่จำเป็นต้องแก้ไขโปรแกรมเมื่อมีการเปลี่ยนแปลงอุปกรณ์

- การแลกเปลี่ยนข้อมูลระหว่างโปรแกรม วินโดว์ได้สนับสนุนการโอนย้ายและแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ เช่น การคัดลอกข้อมูลจากโปรแกรมประยุกต์หนึ่งไปยังคลิปบอร์ด (Clipboard) แล้วคัดลอกจากคลิปบอร์ดไปยังอีกโปรแกรมประยุกต์หนึ่งได้

1.2 รูปแบบของโปรแกรมประยุกต์

จุดประสงค์หลักของวินโดว์คือการสร้างตัวประสานกับผู้ใช้แบบใหม่ที่ใช้งานง่ายกว่าเดิม โปรแกรมประยุกต์จะติดต่อกับผู้ใช้โดยผ่านระบบใหม่ซึ่งประกอบไปด้วย (จิรพัฒน์, 2536)

- วินโดว์ เป็นสิ่งที่ติดต่อกับผู้ใช้อย่างพื้นฐานที่ทุกโปรแกรมประยุกต์จะต้องมีใช้เสมอ วินโดว์จะประกอบด้วย แถบชื่อเรื่อง (Title Bar) แถบเมนู (Menu Bar) แถบเลื่อน (Scroll Bar) ฯลฯ

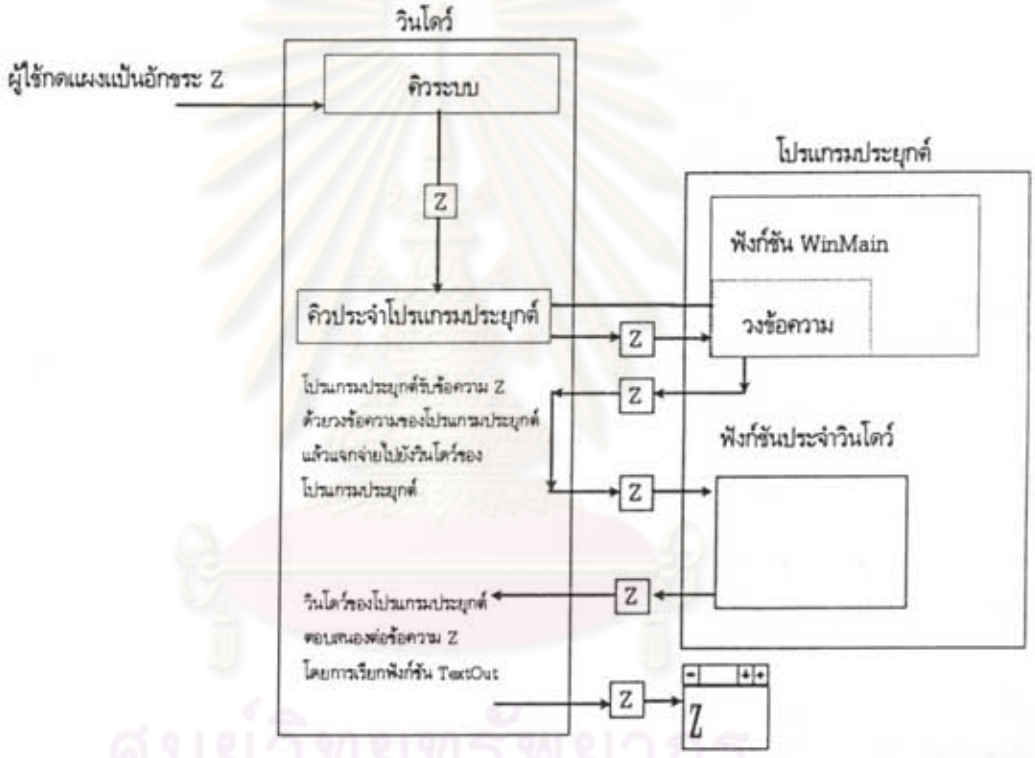
- เมนู เป็นสิ่งที่ใช้รับข้อมูลหลักของโปรแกรมประยุกต์ที่ทำงานบนวินโดว์ เมนูประกอบไปด้วยรายการคำสั่งเรียงกันไป ซึ่งจะให้ผู้ใช้ได้เลือกดูและใช้งานได้ง่าย ๆ

- กรอบโต้ตอบ (Dialog Box) กรอบโต้ตอบเป็นวินโดว์ชนิดหนึ่งที่ใช้งานชั่วคราวในการติดต่อกับผู้ใช้ เช่น ให้ผู้ใช้ป้อนชื่อแฟ้มที่จะทำการเก็บข้อมูลเมื่อเลือกเมนู SaveAs ภายในกรอบโต้ตอบประกอบด้วยตัวควบคุมหลายชนิด ตัวควบคุมแต่ละชนิดมีวิธีติดต่อกับผู้ใช้ต่าง ๆ กัน เช่น บรรณาธิกร (Edit) ใช้สำหรับป้อนข้อความ ปุ่มกด (Push Button) สำหรับให้ผู้ใช้กด (ทำได้โดยการเลื่อนเมาส์ไปยังปุ่มแล้วกดปุ่มที่เมาส์)

1.3 ลักษณะการทำงานของโปรแกรมประยุกต์บนวินโดว์

ในระบบดอสธรรมดาโปรแกรมประยุกต์จะทำการรับข้อมูลเข้าโดยตรงจากอุปกรณ์โดยใช้ฟังก์ชันต่าง ๆ เช่น getchar() เป็นต้น แต่เนื่องจากวินโดว์ทำงานแบบหลายภารกิจและสามารถมีอุปกรณ์รับข้อมูลเข้าได้หลายอุปกรณ์ เพื่อไม่ให้เกิดการแย่งกันรับข้อมูล วินโดว์ได้ออกแบบให้โปรแกรมประยุกต์รับข้อมูลเข้าเป็นข้อความ โดยวินโดว์จะเป็นตัวรับข้อมูลจากอุปกรณ์ต่าง ๆ แล้วแจกจ่ายไปยังคิวข้อความของโปรแกรมประยุกต์ที่เหมาะสม โปรแกรมประยุกต์จะรับข้อมูลเข้าโดยการอ่านข้อความจากคิวข้อความ ดังนั้นส่วนประกอบหลักของโปรแกรมประยุกต์จึงมีส่วนหนึ่งเรียกว่า วงข้อความ (Message Loop) ซึ่งเป็นส่วนที่รับข้อความแล้วแจกจ่ายไปยังวินโดว์ที่เหมาะสม

จากรูปที่ 2.7 เมื่อกดแผงแป้นอักขระ วินโดว์จะรับทราบและส่งข้อความสำหรับแผงแป้นอักขระลงในคิวระบบ (System Queue) แล้วส่งต่อไปยังคิวประจำโปรแกรมประยุกต์ (Application Queue) ของโปรแกรมประยุกต์ที่เหมาะสม จากนั้นวงข้อความก็จะอ่านข้อความแล้วตีความ ซึ่งก็จะได้เป็นข้อความตัวหนังสือ (WM_CHAR) แล้วทำการแจกจ่ายข้อความไปยังวินโดว์ที่เหมาะสม หากฟังก์ชันประจำวินโดว์นั้นตอบสนองข้อความด้วยการแสดงตัวหนังสือออกบนจอภาพ ก็จะใช้ฟังก์ชัน TextOut ในการแสดงตัวอักษรบนพื้นที่ทำงานของวินโดว์นั้น



รูปที่ 2.7 การจัดการข้อความของวินโดว์เมื่อกดแผงแป้นอักขระ (จิรพัฒน์, 2536)

2. การเขียนโปรแกรมประยุกต์บนวินโดว์ด้วยวิธีโปรแกรมเชิงวัตถุ

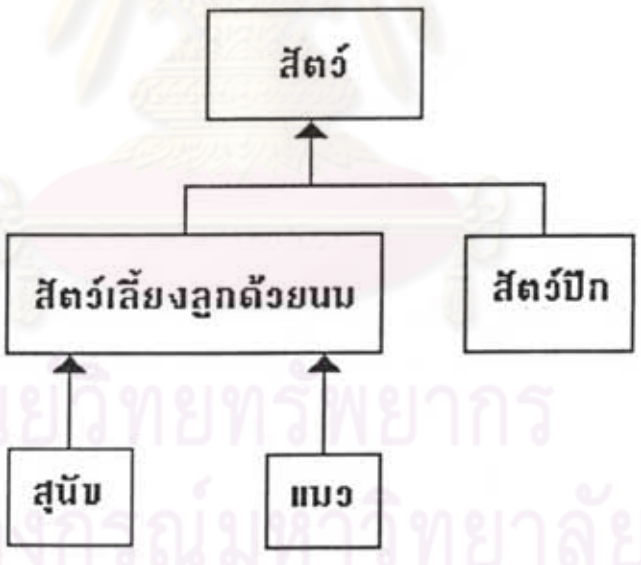
2.1 โปรแกรมเชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ เป็นการแก้ปัญหาแนวใหม่โดยมองปัญหาต่าง ๆ แยกออกเป็นกลุ่มย่อย ๆ ที่มีความสัมพันธ์กัน และใช้ภาษาคอมพิวเตอร์มาเขียนแก้ปัญหากลุ่มย่อยนั้นเป็นหน่วยที่เรียกว่าวัตถุ (Objects) คุณสมบัติที่สำคัญในการเขียนโปรแกรมเชิงวัตถุคือ

- วัตถุ ถือเป็นหน่วยซึ่งมีความสำคัญที่สุดของการเขียนโปรแกรมเชิงวัตถุ โดยวัตถุจะประกอบด้วยสมาชิกที่เป็นข้อมูลหรือฟังก์ชัน โดยข้อมูลหรือฟังก์ชันนั้นอาจถูกกำหนดให้เป็นแบบส่วนตัว (private) ซึ่งใช้ได้เฉพาะวัตถุนั้น หรืออาจกำหนดเป็นแบบสาธารณะ (public) ที่สามารถเรียกใช้จากส่วนไหนของโปรแกรมก็ได้ แนวคิดในการกำหนดแบบนี้เรียกว่า Encapsulation เป็นแนวคิดที่จะป้องกันการเรียกใช้หรือการแก้ไขข้อมูลอย่างไม่ถูกวิธี

- โพลีมอร์ฟิซึม คือการเขียนโปรแกรมที่สามารถใช้ชื่อฟังก์ชันซ้ำกันได้ เช่น ถ้าต้องการเขียนฟังก์ชันหาค่าสมบูรณ์ของตัวแปรเลขจำนวนเต็ม และเลขทศนิยม ถ้าเขียนด้วยภาษาซี จะต้องเขียนเป็น 2 ฟังก์ชันใช้ชื่อต่างกัน เช่น iabs และ fabs แต่ถ้าเขียนด้วยวิธีนี้สามารถใช้ชื่อเดียวกันได้ เมื่อเรียกใช้ตัวแปลภาษา จะทำการตรวจสอบว่าเป็นข้อมูลชนิดใด แล้วจะทำการเรียกใช้ฟังก์ชันที่เหมาะสมกับข้อมูลนั้น

- อินเฮอริแทนซ์ หมายถึงการที่วัตถุหนึ่งสามารถสืบทอดคุณสมบัติจากวัตถุอื่นมาใช้งานได้ ซึ่งเป็นสิ่งสำคัญที่ช่วยสนับสนุนแนวคิดของการกำหนดคลาส (classification) ตัวอย่างเช่น สุนัขลีซาวมีคลาสเป็นสุนัข ซึ่งสืบทอดคุณสมบัติของคลาสสัตว์เลี้ยงลูกด้วยนม และคลาสสัตว์เลี้ยงลูกด้วยนมก็สืบทอดคุณสมบัติมาจากคลาสสัตว์ ตามลำดับ เป็นต้น



รูปที่ 2.8 ตัวอย่างของอินเฮอริแทนซ์

2.2 ภาษาซีพลัสพลัส (Herbert, 1992)

ภาษาซีพลัสพลัส ถูกพัฒนาขึ้นโดย Bjarne Stroustrup ที่ Bell Laboratories ใน Murray Hill, New Jersey เมื่อปี ค.ศ. 1980 โดยตั้งชื่อว่า "C with classes" และภายหลังในปี 1983 จึง

เปลี่ยนชื่อเป็น C++ ต่อมาได้มีการปรับปรุงอีกสองครั้งในปี ค.ศ. 1985 และ 1989 จนถึงปัจจุบันเป็นรุ่น 2.1

ภาษาซีพลัสพลัสเป็นภาษาที่ใช้เขียนโปรแกรมเชิงวัตถุ และแนวคิดของการพัฒนาวินโดว์เป็นแบบโปรแกรมเชิงวัตถุเช่นกัน ดังนั้นภาษาซีพลัสพลัสจึงเหมาะสมที่จะใช้ในการพัฒนาโปรแกรมบนวินโดว์

ภาษาซีพลัสพลัสกำหนดวัตถุในลักษณะโครงสร้างที่เรียกว่าคลาส (class) โดยมีรูปแบบดังนี้

```
class class-name {
   private data and functions
public:
   public data and functions
} object-name-list;
```

ตัวอย่างการกำหนดคลาสของ queue ซึ่งเป็นคลาสของคิว ใช้เก็บข้อมูลเลขจำนวนเต็ม การเก็บและใช้ข้อมูลเป็นในลักษณะข้อมูลที่เข้าก่อนจะถูกนำมาใช้ก่อน การเก็บข้อมูลเข้าจะใช้ฟังก์ชัน qput() และการนำข้อมูลมาใช้ใช้ฟังก์ชัน qget()

```
#include <iostream.h>
// this creates the class queue
class queue {
   int q[100];
   int sloc, rloc;
public:
   void init();
   void qput(int i);
   int qget();
};
```

ภาษาซีพลัสพลัสใช้คุณสมบัติโพลิมอร์ฟิซึม ได้โดยการตั้งชื่อฟังก์ชันซ้ำกันมากกว่า 1 ฟังก์ชัน แต่มีเงื่อนไขว่าการประกาศพารามิเตอร์จะต้องต่างกัน วิธีการนี้เรียกว่า ฟังก์ชันโอเวอร์โหลดดิ้ง เช่น ฟังก์ชันหาค่าสัมบูรณ์ของเลขจำนวนเต็ม และเลขทศนิยม โดยใช้ชื่อฟังก์ชัน abs เหมือนกันแต่พารามิเตอร์ต่างกัน สามารถเขียนได้ดังนี้

```
abs (int & i);
abs (double & r);
```

ภาษาซีพลัสพลัสใช้คุณสมบัติอินเฮอริแทนซ์ ได้โดยการกำหนดคลาสใหม่โดยการถ่ายทอดคุณสมบัติจากคลาสเดิม คลาสใหม่เรียกว่าดีไรว์คลาส ส่วนคลาสเดิมเรียกว่าเบสคลาส การถ่ายทอดคุณสมบัติเรียกว่าอินเฮอริท การกำหนดดีไรว์คลาสมีรูปแบบดังนี้

```
class derived-class-name : access base-class
{
  // ... body of derived class
};
```

ตัวอย่าง คลาส TGObject เป็นคลาสของวัตถุกราฟิก ซึ่งมีสมาชิก Show ซึ่งใช้แสดงรูปวัตถุ และฟังก์ชัน Move ใช้ในการย้ายวัตถุ และคลาส TRectangle เป็นคลาสของวัตถุรูปสี่เหลี่ยม ซึ่งทำการสืบทอดคุณสมบัติจากคลาส TGObject โดยกำหนดฟังก์ชัน Show ใหม่ นอกนั้นใช้คุณสมบัติของ TGObject เช่น ฟังก์ชัน Move เป็นต้น การเขียนคลาส TRectangle สามารถเขียนได้ดังนี้

```
class TGObject {
  TPoint p;
public:
  Show();
  Move(TPoint &NewP);
};
class TRectangle : public TGObject {
public:
  Show();
};
```

2.3 การเขียนโปรแกรมประยุกต์ด้วย Object Windows ในบอร์แลนด์

คลาสที่ใช้ใน Object Windows สามารถแบ่งออกเป็นกลุ่ม ๆ ดังแสดงในรูปที่ 2.9 โดยกลุ่มคลาสที่ใช้ในวิทยานิพนธ์ฉบับนี้มีดังนี้

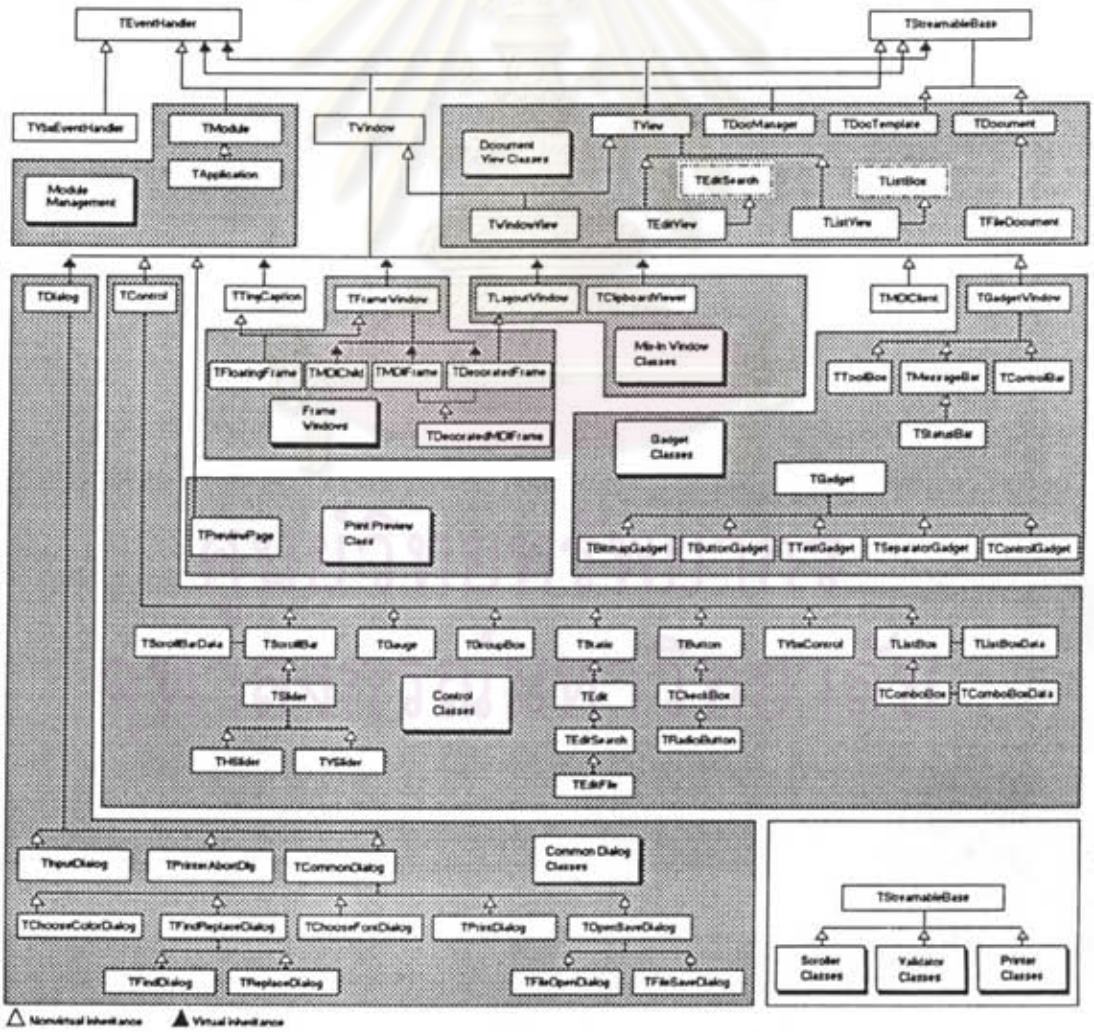
- FrameWindow Class เป็นกลุ่มของคลาสที่จัดการวินโดว์โครงแบบต่าง ๆ เช่น TDecoratedFrame เป็นวินโดว์โครงที่มีส่วนประกอบวินโดว์ทำงาน (Client Window) และวินโดว์ตกแต่ง (Decorated Window) เช่น แถบเครื่องมือ (Tool Bar) แถบสถานะ (Status Bar) เป็นต้น

- Control Class เป็นกลุ่มของตัวควบคุมที่ใช้ในวินโดว์ทั่วไป เช่น TButton เป็นตัวควบคุมชนิดกดปุ่ม ตัวควบคุมนี้ส่วนใหญ่จะใช้ในกรอบควบคุม

- Gadget Class เป็นกลุ่มของตัวควบคุมชนิดหนึ่งซึ่งใช้ในแถบเครื่องมือ แถบสถานะ เช่น TButtonGadget เป็นตัวควบคุมชนิดปุ่มกด

- Print Preview Class เป็นคลาสของวินโดว์ที่ใช้แสดงเอกสารก่อนพิมพ์

- Common Dialog Class เป็นกลุ่มคลาสของกรอบโต้ตอบ เช่น TFileOpenDialog เป็นกรอบโต้ตอบสำหรับป้อนชื่อแฟ้มข้อมูลที่จะทำการเปิด เป็นต้น



รูปที่ 2.9 แสดงคลาสที่ใช้ใน Object Windows (OWL.HLP, Borland International, 1993)