หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6

การพัฒนาโปรแกรม

สภาพแวดล้อมที่จำเป็นในการพัฒนาโปรแกรม

1) เครื่อง IBM PC หรือเครื่องที่เข้ากันได้ (compatible) ซึ่งมีรายละเอียด ดังนี้
   - ซีพียู 80386 หรือ 80386 SX
   - หน่วยความจำบนบอร์ด อย่างต่ำ 2 เมกะไบต์
   - ฮาร์ดดิกส์ความจุ อย่างต่ำ 40 เมกะไบต์
   - ดิกส์ไดร์ฟขนาด 5.25 นิ้ว ความจุ 1.2 เมกะไบต์ หรือ
     ดิกส์ไดร์ฟขนาด 3.5 นิ้ว ความจุ 1.44 เมกะไบต์
   - เมาส์

2) ระบบปฏิบัติการ MS-DOS รุ่น 5.0

3) ระบบจัดการวินโดว์ Microsoft WINDOWS รุ่น 3.1

4) ตัวแปลโปรแกรมภาษาซี Borland C/C++ รุ่น 3.1

5) คลังโปรแกรม (Library) สำหรับการจัดการฐานข้อมูลแบบรีเลชันนัล CodeBase รุ่น 4.5

การสร้างโปรแกรม

เทคนิคที่ใช้ในการวิจัยนี้ เป็นเทคนิคในแบบระบบกราฟิกเชิงโต้ตอบซึ่งมีรูปแบบในการทำงานที่สำคัญประกอบกัน 2 รูปแบบ คือ

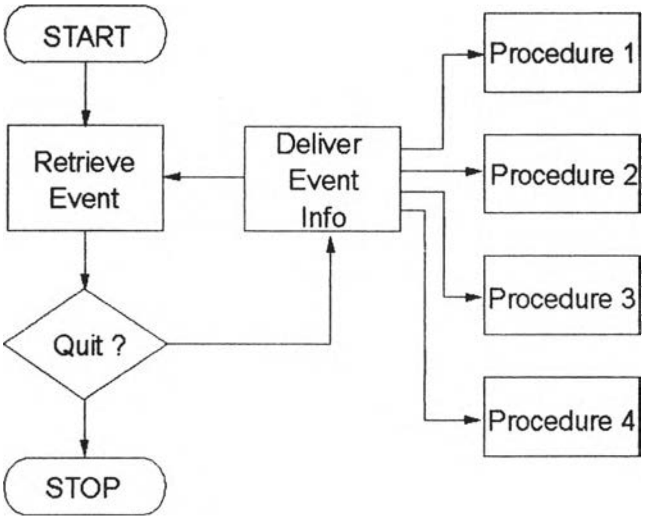
1) การเลือก (selecting) หรือการชี้ (pointing) เป็นการเลือกภาพ หรือวัตถุซึ่งปรากฏอยู่บนจอภาพ ซึ่งทำได้โดยการใช้อุปกรณ์นำข้อมูลเข้าเป็นเครื่องมือในการเลือกภาพหรือวัตถุบนจอภาพที่ต้องการ

2) การกำหนดตำแหน่ง (positioning) เป็นการกำหนดตำแหน่งของภาพหรือวัตถุบนจอภาพ โดยการรับข้อมูลตำแหน่งของพิกเซลบนจอภาพจากผู้ใช้ผ่านทางอุปกรณ์นำเข้า

เทคนิคทั้งสองนี้จะถูกใช้ร่วมกันเสมอ ตลอดการทำงานของโปรแกรม ทั้งนี้อุปกรณ์รับข้อมูลเข้าที่ใช้ในงานวิจัยนี้ กำหนดให้เป็น แป้นพิมพ์ และเมาส์

<u>หลักการทำงานของโปรแกรม</u>

การทำงานของโปรแกรมประเภทกราฟิกเชิงโต้ตอบโดยทั่วไปมักใช้หลัก การหยั่งสัญญาณ (polling) คือ จะต้องให้โปรแกรมเวียนตรวจสอบสถานะของอุปกรณ์รับข้อมูลเข้า คือแป้นพิมพ์ และเมาส์ว่าผู้ใช้มีการกดปุ่มแป้นพิมพ์ หรือกดปุ่มเมาส์ใดหรือไม่ ที่ตำแหน่งพิกัดใด หรือมีการเลื่อนเมาส์หรือไม่ และเลื่อนไปที่ตำแหน่งใดก่อนจะนำไปวิเคราะห์ต่อไปว่าจะต้องดำเนินการอย่างไรเพื่อตอบสนองต่อเหตุการณ์ดังกล่าว



รูปที่ 6.1 แสดงการหยั่งสัญญาณของโปรแกรมเชิงโต้ตอบ

สำหรับโปรแกรมที่สร้างขึ้นในการวิจัยนี้ เป็นโปรแกรมที่สร้างขึ้นในสภาพแวดล้อมของระบบจัดการวินโดว์ ซึ่งระบบจัดการวินโดว์นี้จะทำหน้าที่ในการหยั่งสัญญาณโดยการรวบรวมข้อมูลเข้าจากอุปกรณ์ต่าง ๆ แล้วเก็บไว้ในคิวข้อความของระบบ (system message queue) ก่อนที่จะวิเคราะห์เพื่อแยกแยะรายการในคิวข้อความของระบบ แล้วแจกจ่ายไปยังคิวข้อความของแต่ละโปรแกรม (application message queue) ที่กำลังดำเนินการ (run) ในระบบจัดการวินโดว์ในขณะนั้น

โปรแกรมที่สร้างขึ้นในสภาพแวดล้อมของระบบจัดการวินโดว์ จะต้องมีส่วนของโปรแกรมซึ่งทำหน้าที่ในการวนรอบ (loop) เพื่อดึงเอารายการในคิวซึ่งเป็นตัวบอกเหตุการต่าง ๆ ที่เกิดขึ้น จากนั้นก็จะส่งไปทำการตีความหมาย แล้วจึงส่งไปยังกระบวนการ (procedure) ซึ่งมีหน้าที่รับผิดชอบที่เตรียมไว้สำหรับการประมวลผลเพื่อตอบสนองต่อเหตุการณ์แต่ละเหตุการณ์อย่างเหมาะสมต่อไป เรียกส่วนของโปรแกรมนี้ว่า เมสเสจลูป (Message Loop)

โปรแกรมที่สร้างขึ้นจึงประกอบด้วยส่วนประกอบสำคัญ 2 ส่วนคือ ส่วนโปรแกรมหลัก ซึ่งทำหน้าที่วนรอบเพื่อดึงรายการจากคิวข้อความ และฟังก์ชันประจำวินโดว์ซึ่งทำหน้าที่ประมวลผลตามเหตุการณ์แต่ละเหตุการณ์ที่เกิดขึ้นภายในวินโดว์นั้น โดยฟังก์ชันประจำวินโดว์จะถูกกำหนดไว้ก่อนหน้าในขั้นตอนการสร้างคลาสของวินโดว์นั้น

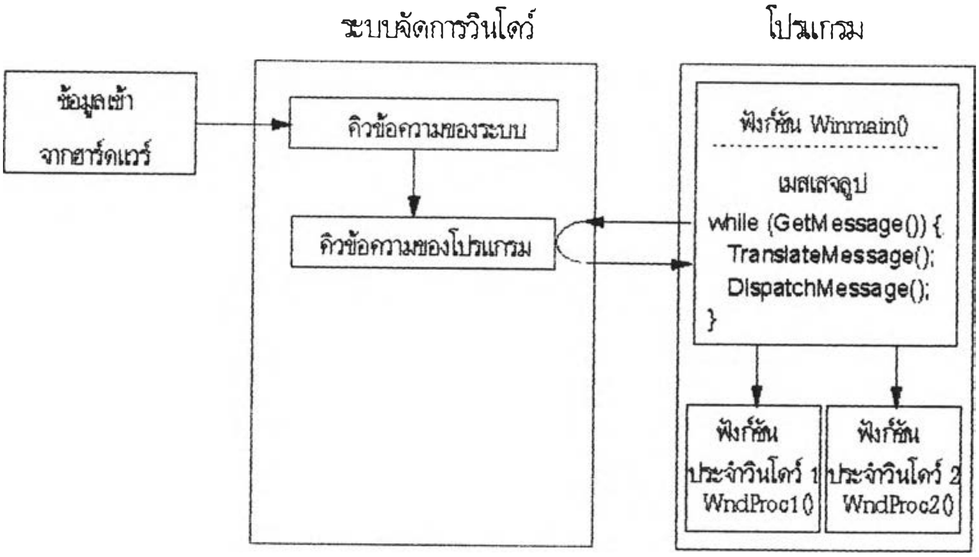
ส่วนของคำสั่งภาษาซี ในโปรแกรมหลักและในส่วนประมวลผลเหตุการณ์ แสดงได้ดังตัวอย่างต่อไปนี้

```
int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpCmdLine, int nCmdShow)
  MSG  msg;
        .
        .
        .
  /* Message Loop */
  while ( GetMessage( &msg, NULL, NULL, NULL ) ) {
     TranslateMessage( &msg );
     DispatchMessage( &msg );
  }
  return ( msg.wParam ) ;
}

LONG FAR PASCAL  WndProc(HWND hWndMain, unsigned message,
     WORD wParam, LONG lParam)

  switch (message) {
     case WM_XXX :
        .
        .
        .
     default:
           return (DefWindowProc( hWndMain, message, wParam, lParam) );
  }
  return (NULL);
}
```

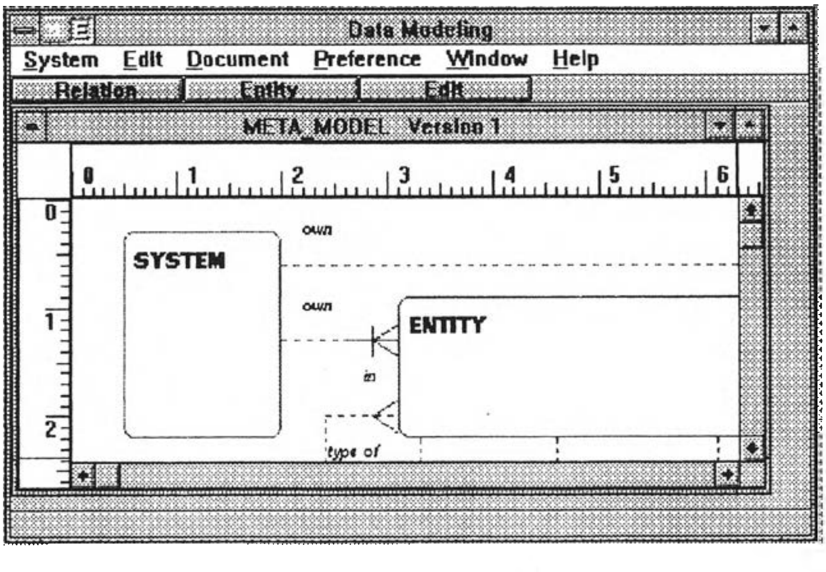
โดยที่การทำงานของโปรแกรมทั้ง 2 ส่วน จะแสดงได้ดังแผนภาพในรูปที่ 6.2



รูปที่ 6.2 แสดงการจัดการกับข้อมูลเข้าระหว่างระบบจัดการวินโดว์กับโปรแกรมโดยผ่านคิวข้อความ

<u>ตัวประสานกับผู้ใช้ (user interface)</u>

ตัวประสานกับผู้ใช้ คือส่วนประกอบต่าง ๆ ที่แสดงอยู่บนจอภาพซึ่งใช้เป็นช่องทางในการสื่อสารระหว่างโปรแกรมและผู้ใช้งานโปรแกรม โดยมีระบบจัดการวินโดว์ทำหน้าที่ในการรับข้อมูลเข้าจากอุปกรณ์ต่าง ๆ และทำหน้าที่ในการตีความหมาย แล้วส่งต่อให้โปรแกรม ทำการประมวลผลตามเหตุการณ์ที่เกิดขึ้น



รูปที่ 6.3 แสดงจอภาพของโปรแกรมหลักซึ่งประกอบด้วย ตัวประสานกับผู้ใช้แบบต่างๆ

ตัวประสานกับผู้ใช้ของระบบจัดการวินโดว์ซึ่งนำมาใช้ในการสร้างโปรแกรมในการวิจัยนี้ มีดังนี้

1) วินโดว์

วินโดว์ คือ กรอบสี่เหลี่ยมบนจอภาพที่โปรแกรมซึ่งสร้างขึ้นในสภาพแวดล้อมของระบบจัดการวินโดว์ใช้เป็นอุปกรณ์สำหรับแสดงผลของโปรแกรม และยังเป็นช่องทางที่ผู้ใช้สามารถติดต่อกับโปรแกรม โดยผ่านทางส่วนประกอบต่าง ๆ ของวินโดว์ เช่น ระบบรายการเลือก ตัวควบคุม และพื้นที่ทำงาน (client area) ภายในกรอบวินโดว์อีกด้วย ระบบจัดการวินโดว์ จะเป็นผู้บริหารข้อมูลเข้าภายในวินโดว์ และยังทำหน้าที่ในการตรวจสอบข้อมูลเข้าดังกล่าวเพื่อทำการแปลงข้อมูลเข้าที่เป็นการเลือกรายการเลือกหรือตัวควบคุมของวินโดว์ไปเป็นข่าวสารเพื่อให้โปรแกรม ที่เป็นเจ้าของวินโดว์สามารถประมวลผลได้ทันทีโดยไม่ต้องตรวจสอบด้วยตนเองแต่โปรแกรมซึ่งสร้างขึ้นในสภาพแวดล้อมของระบบจัดการวินโดว์จะต้องทำการสร้างวินโดว์ขึ้นเอง โดยใช้คำสั่ง Create Window() และจะต้องสร้างกระบวนการ (procedure) ที่คู่กับวินโดว์เพื่อประมวลข้อมูลเข้าที่เกิดขึ้นภายในวินโดว์นั้น

โปรแกรมที่สร้างขึ้นในการวิจัยนี้สร้างขึ้นโดยมีลักษณะการประสานงานแบบหลายเอกสาร หรือ MDI (Multiple Document Interface) ซึ่งเป็นสัญนิยม (convention) แบบหนึ่งในการสร้างโปรแกรมภายใต้สภาพแวดล้อมของระบบจัดการวินโดว์ ทำให้โปรแกรมสามารถสร้างวินโดว์ย่อยสำหรับใช้ในการสร้างโมเดลข้อมูลได้หลายวินโดว์ย่อย ภายใต้วินโดว์หลักเดียวกัน

2) ระบบรายการเลือก

ระบบรายการเลือกของโปรแกรมในการวิจัยนี้ มีลักษณะเป็นระบบรายการเลือกแบบดึงลงซึ่งกำหนดให้เป็นส่วนหนึ่งของวินโดว์หลัก เมื่อผู้ใช้เลือกรายการใดรายการหนึ่งจากรายการเลือก ระบบจัดการวินโดว์จะเป็นผู้ส่งข่าวสารการเลือกดังกล่าวไปยังคิวข้อความของโปรแกรม ดังนั้นในการสร้างโปรแกรมจะต้องเตรียมระบวนการต่าง ๆ ที่ทำงานสอดคล้องกับรายการคำสั่งในรายการเลือกไว้ด้วย

3) ตัวควบคุม

ระบบจัดการวินโดว์ได้เตรียมตัวควบคุมต่างๆ ไว้สำหรับใช้ในกรรับคำสั่งง่ายๆ ภายในวินโดว์ของโปรแกรมที่พัฒนาขึ้นภายใต้ระบบจัดการวินโดว์ และสำหรับในงานวิจัยนี้ ได้มีการใช้ตัวควบคุมใน 3 ลักษณะ คือ

ก. ตัวควบคุมในวินโดว์หลัก ใช้ตัวควบคุมประเภทปุ่มกด (button) สำหรับให้ผู้ใช้เลือกในการสร้างโมเดลข้อมูล ได้แก่

- ปุ่มกดเพื่อเปลี่ยนสถานภาพการทำงานไปเป็นการสร้างเอนติตี
- ปุ่มกดเพื่อเปลี่ยนสถานภาพการทำงานไปเป็นการสร้างรีเลชันชิป
- ปุ่มกดเพื่อเปลี่ยนสถานภาพการทำงานไปเป็นการแก้ไขแผนภาพ

ข. ตัวควบคุมประเภทสโกรลบาร์ ซึ่งใช้ในการควบคุมการแสดงภาพโมเดลข้อมูลในวินโดว์ย่อย

ค. ตัวควบคุมในกล่องคำโต้ตอบ

เมื่อผู้ใช้เลือกตัวควบคุมตัวใดตัวหนึ่ง ระบบจัดการวินโดว์จะเป็นผู้ส่งข่าวสารการเลือกดังกล่าวไปยังคิวข้อความของโปรแกรม ดังนั้นการสร้างโปรแกรมจะต้องเตรียมระบวนการต่าง ๆ ที่ทำงานสอดคล้องกับหน้าที่ของปุ่มควบคุมไว้ด้วย

4) กล่องคำโต้ตอบ

โปรแกรมที่สร้างในการวิจัยนี้ มีการใช้กล่องคำโต้ตอบใน 2 ลักษณะ ดังนี้

ก. กล่องคำโต้ตอบที่ใช้ในการป้อนข้อมูลพจนานุกรม ซึ่งประกอบด้วยตัวควบคุมต่าง ๆ ประกอบกันดังที่ได้แสดงไว้ในขั้นตอนการออกแบบโปรแกรมส่วนหน้า ในบทที่ 3

ข. กล่องคำโต้ตอบสำหรับแสดงข่าวสาร หรือแจ้งข้อผิดพลาดแก่ผู้ใช้

โปรแกรมบรรณาธิกรณ์แผนภาพ

1) ระบบพิกัดที่ใช้

เพื่อให้ภาพที่สร้างขึ้นเป็นอิสระไม่ขึ้นกับชนิดอุปกรณ์แสดงผล ได้กำหนดให้ใช้ ระบบพิกัด 2 ระบบ โดยใช้ระบบพิกัดโลกในการจัดเก็บขนาดและตำแหน่งของโมเดล วัตถุซึ่งเป็นตัวแทนของรูปภาพเอนติตีหรือรีเลชันชิป และใช้ระบบพิกัดบนอุปกรณ์แสดงผล เมื่อต้องการแสดงภาพวัตถุซึ่งก็คือเอนติตีและรีเลชันชิป ดังนี้

ก. ระบบพิกัดโลก กำหนดให้มีหน่วยวัดเป็นนิ้ว โดยพิกัดตามแนวนอนจะเพิ่ม จากซ้ายไปขวา และพิกัดตามแนวตั้งจะเพิ่มจากบนลงล่าง โมเดลวัตถุ จะสร้างขึ้นในพิกัดที่เป็นบวกเสมอ

ข. ระบบพิกัดที่ใช้ในการแสดงผลบนจอภาพและเครื่องพิมพ์ มีหน่วยวัดเป็นจุดภาพ (pixel) โดยพิกัดตามแนวนอนจะเพิ่มจากซ้ายไปขวา และพิกัดตามแนวตั้งจะเพิ่มจากบนลงล่างและมีจุดพิกัดเริ่มต้น (0,0) อยู่ที่มุมบนซ้ายของช่องแสดงภาพ

2) การแปลงส่ง (mapping) ระหว่างระบบพิกัด

การทำงานของโปรแกรมบรรณธิกรณ์แผนภาพจะประกอบด้วยกระบวนการต่างๆซึ่งอาศัยการแปลงส่ง (mapping) ไปมาระหว่างระบบพิกัดทั้งสองระบบ ดังนี้

ก. การสร้างภาพโดยกำหนดจุดซึ่งเป็นตัวแทนของโมเดลวัตถุ โดยใช้อุปกรณ์ ประเภทตัวชี้ในระบบพิกัดของอุปกรณ์แสดงภาพ แล้วจึงทำการแปลงส่งไป เป็นจุดในระบบพิกัดโลกเพื่อจัดเก็บต่อไป

ข. การแสดงรูปภาพ ทำได้โดยการแปลงส่งค่าจุด ซึ่งเป็นตัวแทนโมเดลวัตถุ ในระบบพิกัดโลกไปเป็นจุดในระบบพิกัดบนอุปกรณ์แสดงภาพ เพื่อทำการ วาดภาพวัตถุบนอุปกรณ์นั้น

ค. การแก้ไขภาพ ทำได้โดยการอ่านค่าจุดซึ่งเป็นตำแหน่งใหม่ หรือขนาดใหม่ ของตัวแทนโมเดลวัตถุในระบบพิกัดบนอุปกรณ์แสดงผล จากอุปกรณ์นำเข้า ประเภทตัวชี้ จากนั้นจึงทำการแปลงส่งกลับไปเป็นจุดในระบบพิกัดโลก

3) โครงสร้างข้อมูลในหน่วยความจำ

เพื่อให้โปรแกรมบรรณธิกรณ์แสดงภาพโมเดลวัตถุซึ่งแทนเอนติตีและรีเลชันชิปได้อย่างรวดเร็ว และสามารถแก้ไขโมเดลวัตถุนั้นได้ การจัดเก็บโมเดลวัตถุไว้ในหน่วยความจำจะทำให้การจัดการกับโมเดลวัตถุเป็นไปอย่าง มี ประสิทธิภาพ ทำให้สามารถสร้าง แสดงภาพ แก้ไข และลบโมเดลวัตถุนั้น ซึ่งจัดเก็บในหน่วยความจำได้ในทันที

โครงสร้างข้อมูลในหน่วยความจำซึ่งใช้จัดเก็บโมเดลวัตถุ มีโครงสร้างเป็นแบบตัวแปรแถวลำดับ (array) โดยที่โครงสร้างของแต่ละหน่วยในแถวลำดับมีเขต ข้อมูลที่จำเป็นต้องใช้แทนโมเดลวัตถุ ดังนี้

ชื่อเขตข้อมูล	ชนิด	คำอธิบาย
theShape	int	บอกชนิดโมเดลวัตถุที่จัดเก็บ มีค่าเป็น 1 เมื่อเป็นโมเดลของเอนติตี มีค่าเป็น 0 เมื่อเป็นโมเดลของรีเลชันชิป
PointNum	int	บอกจำนวนจุดที่เป็นตัวแทนของโมเดลวัตถุ
Points	POINT	เป็นเขตข้อมูลที่มีโครงสร้างเป็นแถวลำดับของโครงสร้างข้อมูลชนิด POINT จำนวน 5 หน่วย ซึ่งมีโครงสร้างดังนี้ int x; int y;

ใช้เพื่อเก็บตำแหน่งพิกัดของจุดต่างๆที่เป็นตัวแทนของโมเดลวัตถุ

<u>การจัดการ (operate) ในโปรแกรมสร้างแผนภาพ</u>

1) ส่วนโปรแกรมที่จัดการกับการแสดงผลบนจอภาพ

การนำข้อมูลออกทางอุปกรณ์แสดงผลทั้งหมดของโปรแกรมซึ่งสร้างขึ้นภายใต้ระบบจัดการวินโดว์ จะถูกควบคุมโดยส่วนของโปรแกรมระบบที่เรียกว่า ระบบติดต่ออุปกรณ์กราฟิก หรือ GDI (Graphics Device Interface) โดยผ่านทางคอนเท็กซ์ (context) ของอุปกรณ์ที่ใช้แสดงผลคอนเท็กซ์ของอุปกรณ์จะประกอบด้วย ข้อมูลต่าง ๆ ซึ่งบอกถึงลักษณะและความสามารถของอุปกรณ์แสดงผลนั้น และข้อมูลของสิ่งที่ต้องการแสดงผล โดยโปรแกรมจะต้องทำการยืมคอนเท็กซ์ของอุปกรณ์แสดงผลจากระบบติดต่ออุปกรณ์กราฟิกเมื่อต้องการแสดงผล โดยใช้ฟังก์ชัน GetDC หรือ BeginPaint ก่อนที่จะเรียกฟังก์ชันสำหรับวาดภาพ และเมื่อมีจบการแสดงผลภาพ ก็จะต้องทำการคืนคอนเท็กซ์ของอุปกรณ์แสดงผลแก่ ระบบติดต่ออุปกรณ์กราฟิก ดังนี้

```
hDC = GetDC( hWnd );

วาดภาพด้วยฟังก์ชันวาดภาพ ออกทางวินโดว์ hWnd
โดยการผ่านค่าคอนเท็กซ์ hDC ให้แก่ฟังก์ชันวาดภาพ  เช่น
TextOut( hDC, 10, 10, "Hello !", 7 );

ReleaseDC(hDC);
```

หรือ

```
PAINTSTRUCT ps;

hDC = BeginPaint( hWnd, &ps );

วาดภาพด้วยฟังก์ชันวาดภาพ ออกทางวินโดว์ hWnd โดยการผ่านค่าคอนเท็กซ์ hDC เช่น
TextOut( hDC, 10, 10, "Hello !", 7 );

EndPaint( hWnd, &ps );
```

คอนเท็กซ์ของอุปกรณ์ทำหน้าที่เป็นสะพานเชื่อมระหว่างโปรแกรมประยุกต์กับโปรแกรมขับอุปกรณ์ เมื่อโปรแกรมประยุกต์ต้องการแสดงผลออกทางอุปกรณ์ต่าง ๆ ก็สามารถเรียกใช้งานฟังก์ชันของระบบติดต่อ อุปกรณ์กราฟิก (GDI) ผ่านทางคอนเท็กซ์ของอุปกรณ์ โดยฟังก์ชันเหล่านี้จะผ่านข้อมูลไปให้กับโปรแกรม ขับอุปกรณ์ซึ่งทำหน้าที่ในการควบคุมอุปกรณ์โดยตรง

โปรแกรมประยุกต์ภายใต้ระบบจัดการวินโดว์สามารถเรียกใช้ฟังก์ชันใน GDI สำหรับแสดงผลออกทางอุปกรณ์แสดงผล โดยฟังก์ชันของ GDI มีลักษณะการทำงานไม่ขึ้นกับอุปกรณ์ แต่จะผ่านค่าต่าง ๆ ที่ได้จากโปรแกรมประยุกต์ ไปยังโปรแกรมขับอุปกรณ์ของอุปกรณ์แสดงผลนั้น

เพื่อช่วยในการทำงานแสดงผลในลักษณะไม่ขึ้นกับอุปกรณ์ ระบบติดต่ออุปกรณ์ กราฟิก (GDI) มีภาวะ (mode) ในการแปลงส่งซึ่งเตรียมไว้ให้ผู้พัฒนาโปรแกรมได้ใช้งาน 8 ภาวะ ซึ่งแต่ละภาวะมีลักษณะในการใช้งานเฉพาะดังรายละเอียดในตารางต่อไปนี้

ภาวะในการแปลงส่ง	หน่วยวัดของการแปลงส่ง	ทิศทางการเพิ่มของพิกัดในแกนตั้ง	ทิศทางการเพิ่มของพิกัดในแกนนอน
MM_ANISOTROPIC	ปรับเปลี่ยนได้โดยพิกัดแกนตั้งเท่ากับพิกัดแกนนอนหรือไม่ก็ได้	ปรับเปลี่ยนได้	ปรับเปลี่ยนได้
MM_ISOTROPIC	ปรับเปลี่ยนได้โดยพิกัดแกนตั้งต้องเท่ากับพิกัดแกนนอน	ปรับเปลี่ยนได้	ปรับเปลี่ยนได้
MM_HIENGLISH	0.001 นิ้ว	จากซ้ายไปขวา	จากล่างขึ้นบน
MM_HIMETRIC	0.01 มิลลิเมตร	จากซ้ายไปขวา	จากล่างขึ้นบน
MM_LOENGLISH	0.01 นิ้ว	จากซ้ายไปขวา	จากล่างขึ้นบน
MM_LOMETRIC	0.1 มิลลิเมตร	จากซ้ายไปขวา	จากล่างขึ้นบน
MM_TEXT	1 พิกเซล	จากซ้ายไปขวา	จากบนลงล่าง
MM_TWIPS	1/1440 นิ้ว	จากซ้ายไปขวา	จากล่างขึ้นบน

ตารางที่ 6.1 แสดงภาวะในการแปลงส่งของ GDI 8 ภาวะ

สำหรับโปรแกรมในการวิจัยนี้ มีการเรียกใช้ฟังก์ชัน GDI ในลักษณะต่าง ๆ เพื่อการจัดการแสดงผลบนจอภาพ ดังนี้

ก. การกำหนดค่าเริ่มต้น ในการแสดงผลภาพทางจอภาพ

1. กำหนดภาวะการแปลงส่งระหว่างระบบพิกัดโลกและระบบพิกัดจอภาพ

```
SetMapMode( hDC, MM_ENLISH );
SetMapMode( hDC, MM_ANISOTROPIC );
dw = GetViewportExt(hDC);
SetViewportExt(hDC, LOWORD(dw), -HIWORD(dw));
```

2. กำหนดตำแหน่งพิกัดเริ่มต้นให้กับวินโดว์ในพิกัดโลก

```
SetWindowOrg( hDC, 0, 0 );
```

ข. การสโกรลซึ่งทำได้โดยการกำหนดตำแหน่งพิกัดตำแหน่งใหม่ให้กับวินโดว์ในพิกัดโลก แล้วทำการแสดงภาพซึ่งอยู่ในวินโดว์ ออกทางกรอบมองภาพบนจอภาพ ดังนี้

```
SetWindowOrg( hDC, x, y );
InvalidateRect( hWnd, NULL, TRUE );
UpdateWindow( hWnd );
```

ค. การแปลงพิกัด ระหว่างระบบพิกัดโลกและระบบพิกัดของอุปกรณ์ ที่กำหนดโดยคอนเท็กซ์อุปกรณ์ hDC หรือทำการแปลงกลับคืนโดยฟังก์ชัน DPtoLP() และ LPtoDP() ดังนี้

```
POINT ps;
LPtoDP( hDC, &ps, 1 );
```

ใช้ในการแปลงตำแหน่งพิกัดตรรกะในระบบพิกัดโลก ไปเป็นตำแหน่งพิกัดในช่องแสดงภาพบนอุปกรณ์แสดงผล

```
DPtoLP( hDC, &ps, 1 );
```

ใช้ในการแปลงตำแหน่งพิกัดในช่องแสดงภาพบนอุปกรณ์แสดงผลไปเป็นตำแหน่งพิกัดตรรกะใน ระบบพิกัดโลก

ง. การวาดเส้นตรง และเส้นตรงต่อเนื่องซึ่งใช้แทนภาพรีเลชันชิป เป็นการวาดเชื่อมโยงระหว่างจุด

โดยโปรแกรมในการวิจัยนี้จะรับตำแหน่งพิกัดของจุด ในระบบพิกัดอุปกรณ์ แล้วจึงแปลงไปเป็นจุดในระบบพิกัดโลกซึ่งจะถูกใช้ในฟังก์ชันวาดเส้นตรง ดังนี้

POINT Points[5];	เป็นตัวแปรแถวลำดับที่ใช้ในการเก็บตำแหน่งพิกัดของจุดต่าง ๆ ที่อยู่บนเส้นตรง
int NumPoint;	เป็นตัวแปรที่ใช้เก็บจำนวนจุดบนเส้นตรง

```
PolyLine( hDC, Points, NumPoint );
```

จ. การวาดรูปสี่เหลี่ยมซึ่งใช้แทนภาพเอนติตี เป็นการวาดรูปสี่เหลี่ยมระหว่างจุดบนซ้าย และจุดล่างขวา โดยโปรแกรมในการวิจัยนี้จะรับตำแหน่งพิกัดของจุดทั้งสองในระบบพิกัดอุปกรณ์ แล้วจึงแปลงไปเป็นจุดในระบบพิกัดโลกซึ่งจะถูกใช้ในฟังก์ชันวาดรูปสี่เหลี่ยม ดังนี้

POINT Points[2];	เป็นตัวแปรแถวลำดับที่ใช้ในการเก็บตำแหน่งพิกัดของจุดบนซ้าย และจุดล่างขวาของรูปสี่เหลี่ยม
int NumPoint;	เป็นตัวแปรที่ใช้เก็บจำนวนจุดบนเส้นตรง

```
Rectangle( hDC, Points[0].x, Points[0].y, Points[1].x, Points[1].y );
```

ฉ. การแสดงข้อความ ซึ่งใช้แสดงชื่อรีเลชันชิปชื่อเอนติตี และชื่อแอตตริบิวที่ตำแหน่งพิกัดตรรกะ (x,y) ในระบบพิกัดโลกโดยใช้ฟังก์ชัน TextOut() ดังนี้

```
TextOut( hDC, x, y, output_string, lstrlen(output_string) );
```

2) ส่วนของโปรแกรมที่จัดการกับการวาดแผนภาพ

การสร้างภาพเส้นตรงและสี่เหลี่ยมในโปรแกรมในการวิจัยนี้ ใช้เทคนิคการสร้างภาพเชิงโต้ตอบดังมีวิธีการดังต่อไปนี้

ก. การวาดเส้นตรง ทำได้โดยกำหนดจุดเริ่มต้นแล้วขณะที่เลื่อนตัวชี้ตำแหน่งไปยังจุดปลายอีกจุดหนึ่ง ก็จะแสดงเส้นไปยังจุดของตัวชี้ตำแหน่งตลอดเวลา จนเมื่อกำหนดจุดปลายแล้วจึงจะแสดงเส้นอย่างถาวรเชื่อมจุดปลายทั้งสอง

ข. การวาดเส้นที่มีลักษณะต่อเนื่อง จะทำโดยที่เมื่อวาดเส้นเชื่อมจุดเริ่มต้นกับจุดปลายแล้ว ก็จะใช้จุดปลายของเส้นตรงนั้นเป็นจุดเริ่มต้นแล้วลากไปยังจุดปลายอีกจุดที่กำหนดขึ้นใหม่

ค. การวาดสี่เหลี่ยมแทนเอนติตี ทำได้โดยกำหนดจุดที่ต้องการให้เป็นจุดกึ่งกลางของรูปสี่เหลี่ยมที่แทนเอนติตี เมื่อกำหนดจุดที่ต้องการแล้ว จะแสดงภาพสี่เหลี่ยมขนาดที่เหมาะสม

ง. การวาดสี่เหลี่ยมแสดงขอบเขตการเลือกรูปภาพ ทำได้โดยกำหนดจุดเพียงสองจุดคือจุดที่อยู่ตรงข้ามกันในแนวทะแยง การสร้างภาพเริ่มจากกำหนดจุดที่มุมใดมุมหนึ่ง แล้วเลื่อนตัวชี้ตำแหน่งไปยังจุดตรงข้าม โดยในขณะที่เลื่อนตัวชี้ตำแหน่ง จะต้องแสดงภาพสี่เหลี่ยมจากจุดเริ่มต้นไปยังจุดที่ตัวชี้ตำแหน่ง

<u>การพิมพ์</u>

โปรแกรมที่สร้างขึ้นในสภาพแวดล้อมของระบบจัดการวินโดว์ มีคุณสมบัติของความไม่ขึ้นกับอุปกรณ์ (device independent) สูงทำให้สามารถแสดงผลบนอุปกรณ์แสดงผลต่าง ๆ ได้หลายชนิดทั้งจอภาพและเครื่องพิมพ์ โดยผู้พัฒนาโปรแกรมไม่จำเป็นต้องรู้จักว่ามีอุปกรณ์ชนิดใดต่ออยู่และไม่จำเป็นต้องรู้ว่ารหัสควบคุมของ อุปกรณ์นั้นเป็นอย่างไรด้วย เพราะนอกจากระบบจัดการวินโดว์จะเตรียมระบบพิกัด และการติดต่อผ่านคอนเท็กซ์ของอุปกรณ์ประเภทจอภาพซึ่งทำให้ผู้พัฒนาโปรแกรมไม่จำเป็นต้องทำการควบคุมการทำงานของอุปกรณ์โดยตรงแล้ว ระบบจัดการวินโดว์ยังได้เตรียมคอนเท็กซ์สำหรับอุปกรณ์ประเภทเครื่องพิมพ์อีกด้วย ดังนั้นการพิมพ์ภาพออกทางเครื่องพิมพ์จึงเปรียบเสมือนกับการสร้างภาพผ่านคอนเท็กซ์ของเครื่องพิมพ์ เช่นเดียวกับการสร้างภาพผ่าน คอนเท็กซ์ของอุปกรณ์ประเภทจอภาพ โดยเรียกใช้ฟังก์ชันการวาดภาพเดียวกัน ดังนั้นผู้พัฒนาโปรแกรมจึงสามารถสร้างภาพต่าง ๆ ออกทางเครื่องพิมพ์ได้ โดยการเปลี่ยนชนิดจากคอนเท็กซ์ของจอภาพ ไปเป็นคอนเท็กซ์ของเครื่องพิมพ์เท่านั้น โดยไม่ต้องทำการแก้ไขโปรแกรมส่วนแสดงผลภาพแต่อย่างใด

การพิมพ์ภาพออกทางเครื่องพิมพ์มีขั้นตอนดังนี้

1) สร้างคอนเท็กซ์อุปกรณ์ของเครื่องพิมพ์ โดยการเรียกฟังก์ชัน CreateDC ดังนี้

```
hPrinterDC = CreateDC(DriverName,DeviceName,PortName,0);
```

2) กำหนดระบบการแปลงส่งระหว่างระบบพิกัดโลกกับระบบพิกัดของเครื่องพิมพ์ และกำหนดตำแหน่งพิกัดเริ่มต้นให้กับวินโดว์ในพิกัดโลก เช่นเดียวกันกับที่ทำในการแสดงผลทางจอภาพ

3) เริ่มการร้องขอการพิมพ์โดยใช้รหัส STARTDOC ของฟังก์ชัน Escape ดังนี้

```
Escape(hPrinterDC,STARTDOC,0,PrintName,0);
```

3) แสดงผลภาพด้วยการเรียกฟังก์ชันการวาดภาพผ่านทางคอนเท็กซ์ของเครื่องพิมพ์

4) ขึ้นหน้าใหม่โดยใช้รหัส NEWFRAME ของฟังก์ชัน Escape ดังนี้

```
Escape( hPrinterDC, NEWFRAME, 0, 0, 0);
```

5) จบการร้องขอการพิมพ์โดยใช้รหัส ENDDOC ของฟังก์ชัน Escape ดังนี้

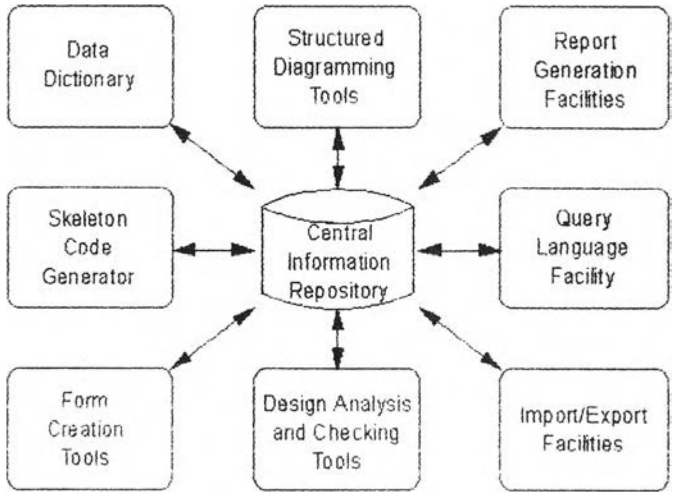
```
Escape( hPrinterDC, NEWFRAME, 0, 0, 0);
```

6) ลบคอนเท็กซ์ของเครื่องพิมพ์ โดยการเรียกฟังก์ชัน DeleteDC ดังนี้

```
DeleteDC( hPrinterDC );
```

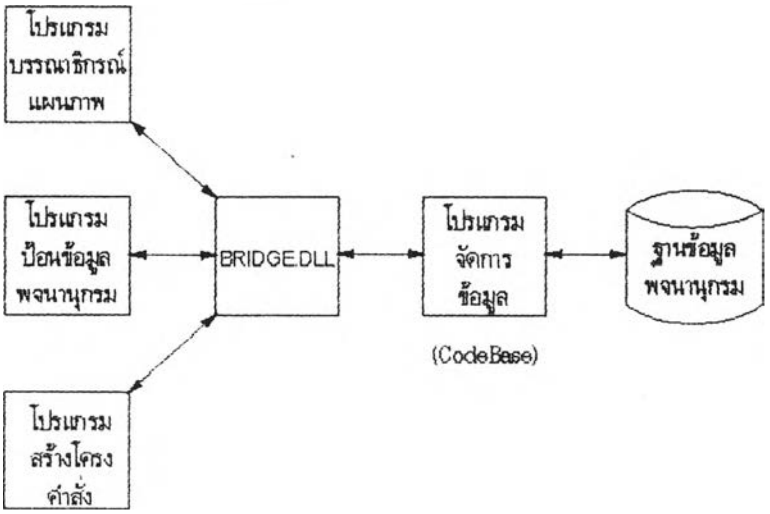
<u>การพัฒนาส่วนเชื่อมต่อระหว่างโปรแกรมส่วนหน้ากับโปรแกรมจัดการพจนานุกรมข้อมูล</u>

จากแนววิธีการของเคส ซึ่งกำหนดรูปแบบของการจัดเก็บข้อมูล ให้มีพจนานุกรมข้อมูลเป็นศูนย์กลางในการจัดเก็บรวบรวมข้อมูลจากกระบวนการต่างๆ ในการพัฒนาระบบงานและทำหน้าที่เป็นตัวกลางเชื่อมต่อในการส่งผ่านข้อมูลระหว่างกระบวนการต่างๆ ในการพัฒนาระบบงานอีกด้วย การวิจัยนี้จึงได้ออกแบบพจนานุกรมข้อมูลในลักษณะริเลชันนัลโดยมีคลังโปรแกรมจัดการฐานข้อมูล Codebase เวอร์ชัน 4.5 เป็นตัวควบคุมการจัดการข้อมูลในพจนานุกรม



รูปที่ 6.4 แผนภาพแสดงการเชื่อมโยงเครื่องมือต่างๆ ในการพัฒนาระบบงาน ผ่านตัวกลางซึ่งเป็นฐานข้อมูลพจนานุกรม

เพื่อให้โปรแกรมส่วนหน้า มีความเป็นอิสระจากโปรแกรมจัดการฐานข้อมูล ผู้วิจัยจึงได้พัฒนาส่วนเชื่อมโยงระหว่างโปรแกรมจัดการข้อมูลในพจนานุกรมข้อมูลกับโปรแกรมส่วนหน้า ซึ่งในการวิจัยนี้ประกอบด้วยส่วนโปรแกรมบรรณาธิกรณ์แผนภาพ ส่วนโปรแกรมแบบฟอร์มสำหรับป้อนข้อมูลพจนานุกรมทางจอภาพ และส่วนโปรแกรมสำหรับสร้างโครงคำสั่งภาษาเอสคิวเอล โดยจัดสร้างไว้เป็นคลังโปรแกรมในลักษณะเชื่อมโยงแบบพลวัต (Dynamic Link Library หรือ DLL) โดยกำหนดชื่อส่วนโปรแกรมนี้ว่า BRIDGE.DLL



รูปที่ 6.5 แสดงแผนภาพการติดต่อระหว่างโปรแกรมส่วนหน้ากับโปรแกรมจัดการข้อมูล สำหรับโปรแกรมในการวิจัยนี้ ผ่านฟังก์ชันในคลังโปรแกรม BRIDGE.DLL