บทที่ 3

การกำหนดขอบเขตของข้อมูลด้วยรูป

แนวความคิดของการกำหนดขอบเขตของข้อมูลด้วยรูป

การสร้างข้อมูลที่มีจำนวนมากและมีคุณลักษณะที่เด่นชัดนี้สามารถสร้างขึ้นมาได้โดยง่าย ถ้าใช้วิธีการมองข้อมูลในลักษณะของภาพรวมเป็นหลัก โดยใช้การสร้างข้อมูลโดยผ่านระบบการติดต่อกับผู้ใช้ด้วยกราฟฟิกส์ (Graphics User Interface-GUI) ภายใต้ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ ซึ่งจะเป็นการออกแบบเชิงทัศน์โดยใช้ภาพเป็นสื่อถึงคุณลักษณะต่างๆ ของข้อมูลที่ต้องการสร้างขึ้นมา และให้คอมพิวเตอร์สร้างข้อมูลที่มีคุณลักษณะตามที่กำหนดขึ้นมาให้

การสร้างข้อมูลทำได้โดยการปรับตั้งค่าตัวแปรบางตัวเพื่อกำหนดคุณลักษณะของข้อมูลที่จะสร้างขึ้นรวมทั้งการปรับเปลี่ยนลักษณะของข้อมูลโดยผ่านการออกแบบเชิงทัศน์โดยจะมีลักษณะแตกต่างกันไปตามความเหมาะสมของข้อมูลในแต่ละแบบดังจะกล่าวในหัวข้อต่อไปแยกตามชนิดของข้อมูล

นอกจากนี้ผู้ใช้งานยังสามารถปรับแต่งคุณลักษณะของข้อมูลต่อได้อีกตามต้องการโดยการวาดลักษณะข้อมูลที่ต้องการลงไปได้โดยให้ความรู้สึกเหมือนกับการวาดรูปเท่านั้น โดยการใช้เครื่องมือต่างๆ เช่น ปากกา สเปรย์พ่นสี หรือการลากเส้นตรง เป็นต้น

หลังการสร้างและปรับแต่งคุณลักษณะของข้อมูลเสร็จสิ้นระบบจะเริ่มทำการสร้างตัวข้อมูลซึ่งมีคุณลักษณะตามที่ได้ออกแบบไว้แล้วออกมา

การแทนข้อมูลด้วยภาพยังคงมีปัญหาในเรื่องของความละเอียดของกราฟฟิกส์ที่ใช้แสดงผล (Graphic Resolution) เนื่องจากความละเอียดของภาพที่ใช้แสดงแทนตัวข้อมูลจะต้องไม่มีผลกระทบต่อข้อมูลที่จะสร้างขึ้นจริง ดังนั้นจึงต้องเลือกใช้หน่วยแสดงผลที่ไม่ขึ้นกับความละเอียดของจอภาพและการ์ดแสดงผล ซึ่งในระบบปฏิบัติการไมโครซอฟต์วินโดวส์ ได้มีการใช้หน่วย

Twip เพื่ออ้างอิงตำแหน่งบนจอแสดงผลโดยไม่ขึ้นกับความละเอียดที่ใช้อยู่แล้ว จึงได้นำหน่วยนี้มาใช้เพื่ออ้างอิงตำแหน่งบนจอแสดงผลในระบบสร้างข้อมูลเชิงทัศน์ที่ได้พัฒนาขึ้นมานี้ด้วย

การแทนข้อมูลแบบรายการ

ในการออกแบบเชิงทัศน์ของข้อมูลแบบรายการ ข้อมูลแบบรายการจะแสดงโดยใช้กราฟแบบเอกซ์-วาย (X-Y Graph) ซึ่งเป็นกราฟ 2 มิติ โดยทางแกนนอนจากซ้ายไปขวาจะแทนตำแหน่งของข้อมูลในรายการจากตำแหน่งแรกจนถึงตำแหน่งสุดท้ายในรายการ และทางแกนตั้งจากล่างขึ้นบนจะแทนค่าของข้อมูลจากน้อยไปหามาก และแต่ละจุดก็จะแทนสมาชิกภายในรายการ

อย่างไรก็ตามคุณสมบัติของข้อมูลแบบรายการทั้งหมดไม่สามารถกำหนดได้โดยเพียงการใช้กราฟ เอกซ์-วาย ยังต้องมีการกำหนดค่าอื่นๆ ร่วมดังนี้คือ

1) จำนวนของสมาชิกในรายการ ซึ่งตำแหน่งในแกนนอน (X) ของกราฟ เอกซ์-วาย ทางซ้ายสุดจะแทนสมาชิกตัวแรกของรายการและตำแหน่งทางขวาสุดจะแทนสมาชิกลำดับสุดท้ายในรายการ

2) กำหนดช่วงของข้อมูลบนกราฟ โดยการกำหนดค่าต่ำสุดและค่าสูงสุดที่เป็นไปได้ของสมาชิกภายในรายการ ซึ่งช่วงของข้อมูลนี้จะแทนกราฟในแนวแกนตั้ง (Y) นั่นเอง

เมื่อทำการกำหนดโครงสร้างข้อมูลแบบรายการโดยใช้ตัวแปลภาษาไมโครซอฟต์ วิชวล เบสิก (Microsoft Visual Basic) จึงมีการแยกเก็บระหว่างคุณสมบัติของรายการและตัวรายการ โดยส่วนของคุณสมบัติของรายการสามารถเขียนเป็นโครงสร้างได้ดังนี้ สาเหตุที่ทำให้ต้องมีการแยกเก็บระหว่างคุณสมบัติและตัวรายการเนื่องจากข้อจำกัดของตัวแปลภาษาดังที่ได้อธิบายไว้ในบทที่ 6

```
Type ListProperties
    Node As Long            'No of node in list
    MinValue As Long        'Minimum value in list
    MaxValue As Long        'Maximum value in list
End Type
```

Dim *ListProp* as ***ListProperties*** 'List header, keep all properties

โดยโครงสร้างข้อมูล ***ListProperties*** จะประกอบด้วย *Node* ซึ่งบอกถึงจำนวนสมาชิกที่มีในรายการ ส่วน *MinValue* และ *MaxValue* บอกถึงค่าต่ำสุดและค่าสูงสุดที่ข้อมูลในรายการสามารถมีได้

ส่วนตัวรายการ *ListValue* ซึ่งข้อมูลในรายการจะถูกแทนด้วยอาเรย์ของข้อมูลแบบจำนวนเต็มแบบยาว (Long Integer) ซึ่งใช้หน่วยความจำ 32 บิตแทนจำนวนตัวเลขที่เป็นไปได้ทั้งหมดคือ -2,147,483,648 ถึง 2,147,483,647 โดยจำนวนสมาชิกในอาเรย์จะได้มาจากจำนวนสมาชิกของรายการที่ได้กำหนดค่าไว้ก่อนแล้วใน *Listprop.Node*

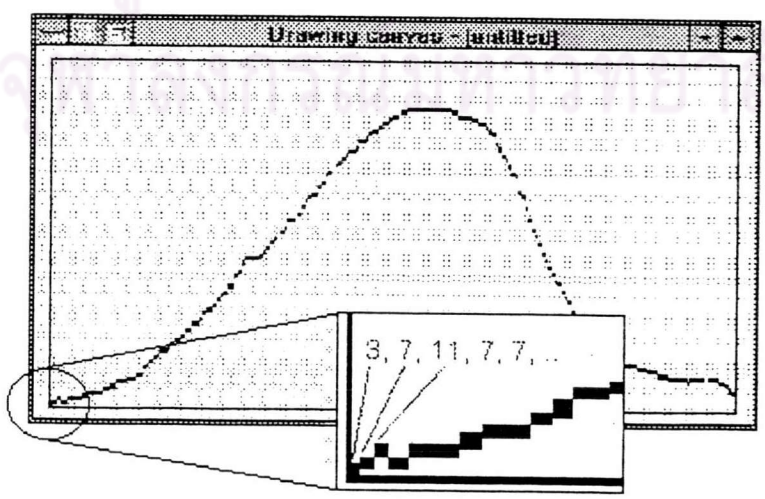
ReDim *ListValue*(*ListProp.Node*) As Long 'Keep the value for each node in list

จากตัวอย่างของรายการดังรูปที่ 3.1 ซึ่งกำหนดจำนวนสมาชิกสูงสุดในรายการไว้ที่ 200 ตัวและกำหนดค่าต่ำสุดคือ 0 และค่าสูงสุดคือ 100 ดังนั้นจะได้ว่า *ListProp* ซึ่งเป็นตัวแปรของโครงสร้างข้อมูล ***ListProperties*** จะมีค่าดังต่อไปนี้

ListProp.Node = 200

ListProp.MinValue = 0

ListProp.MaxValue = 100



รูปที่ 3.1 ตัวอย่างการแทนข้อมูลรายการด้วยรูป

และจากตัวอย่างในรูปที่ 3.1 จะได้ว่าค่าของข้อมูลในอาเรย์ *ListValue* ซึ่งเก็บค่าของสมาชิกแต่ละตัวในรายการจะมีค่าเป็นดังต่อไปนี้

ListValue(0) = 3

ListValue(1) = 7

ListValue(2) = 11

ListValue(3) = 7

ListValue(4) = 7

...

ซึ่งค่าที่เก็บจะเก็บแบบตัวเลขจำนวนเต็มซึ่งมีการแปลงค่าจากตำแหน่งของจุดบนจอภาพให้อยู่ในอัตราส่วนของช่วงข้อมูลที่กำหนดไว้ในคุณสมบัติของรายการ โดยการประมวลผลแบบทันที (Real-Time Processing) โดยเปรียบเทียบช่วงของข้อมูลที่เป็นไปได้ (มาจากค่ามากสุด - ค่าน้อยสุด) กับตำแหน่งในแนวตั้งของจุดในกราฟ เอกซ์-วาย

การแทนข้อมูลแบบต้นไม้

การออกแบบเชิงทัศน์ของข้อมูลแบบต้นไม้ สำหรับข้อมูลแบบต้นไม้เราจะพิจารณารูปร่างโดยรวมภายนอกของต้นไม้เป็นหลักโดยจะใช้เส้นขอบของรูปหลายเหลี่ยม (Polygon) ที่มีจุดหนึ่งเป็นจุดรากของต้นไม้ โดยเราจะไม่พิจารณาต้นไม้ในรายละเอียดว่าสมาชิกตัวใดต้องเชื่อมต่อกับสมาชิกอื่นๆ อย่างไร แต่เราจะให้รูปหลายเหลี่ยมนี้แทนลักษณะโดยรวมคร่าวๆ ของรูปร่างข้อมูลต้นไม้ รูปหลายเหลี่ยมนี้จะมีจุดยอดจุดหนึ่งที่อยู่บนสุดแทนตำแหน่งของสมาชิกที่เป็นรากของต้นไม้ โดยเส้นขอบของรูปหลายเหลี่ยมนี้จะแทนขอบเขตของสมาชิกและการเชื่อมต่อในต้นไม้นี้ และที่แนวเส้นขอบด้านล่างจะแทนแนวของสมาชิกในต้นไม้ที่เป็นใบ

ข้อมูลเพิ่มเติมที่ต้องมีการกำหนดเพิ่มเพื่อใช้ร่วมกับรูปหลายเหลี่ยมเพื่อกำหนดคุณลักษณะของข้อมูลแบบต้นไม้นี้ให้ครบถ้วนมีดังนี้คือ

1) ชนิดของต้นไม้ ชนิดของต้นไม้สามารถกำหนดได้ดังนี้คือ ต้นไม้แบบทวิภาค ต้นไม้ค้นแบบทวิภาค ต้นไม้แบบเอวีแอล ต้นไม้ค้นแบบหลายทาง และต้นไม้แบบบี ซึ่งชนิดของต้นไม้แบบทวิภาค ต้นไม้ค้นแบบทวิภาค และต้นไม้แบบเอวีแอล จะมีจำนวนของลูกที่แต่ละโหนดสามารถมีได้สูงสุดคือ 2 แต่ต้นไม้ค้นแบบหลายทาง และต้นไม้แบบบี สามารถกำหนดจำนวนลูกที่สามารถมีได้ในแต่ละโหนดได้ นอกจากนี้ ชนิดของต้นไม้ยังบอกถึงว่าข้อมูลที่สร้างขึ้นมานั้นยังต้องเป็นไปตามข้อกำหนดของต้นไม้ชนิดนั้นๆ ด้วย

2) ความสูงของต้นไม้ บอกถึงจำนวนของระดับของสมาชิกที่สามารถมีได้ในต้นไม้นั้นๆ ลบด้วยหนึ่ง ดังนั้นต้นไม้ที่ถูกกำหนดให้มีความสูงเป็นศูนย์ จะมีสมาชิกได้หนึ่งระดับหรือมีได้เพียงสมาชิกที่เป็นรากนั่นเอง

3) จำนวนของลูกสูงสุดที่แต่ละโหนดในต้นไม้มีได้ (Branching Factor) ค่าของข้อมูลนี้บ่งบอกถึงจำนวนโหนดลูกที่มากที่สุดที่โหนดที่เป็นแม่สามารถมีได้ โดยค่านี้จะสามารถกำหนดได้ก็ต่อเมื่อชนิดของต้นไม้เป็นต้นไม้ค้นแบบหลายทาง และต้นไม้แบบบี เท่านั้น

เมื่อทำการกำหนดโครงสร้างข้อมูลแบบรายการโดยใช้ภาษาวิชวล เบสิก จึงมีการแยกเก็บระหว่างคุณสมบัติของต้นไม้และตัวข้อมูลของโหนดในต้นไม้ (สาเหตุที่ต้องแยกเก็บได้อธิบายไว้ในบทที่ 6) โดยส่วนของคุณสมบัติของต้นไม้สามารถเขียนเป็นโครงสร้างชื่อ ***TreeProperties*** ได้ดังนี้

```
Type TreeProperties
    TreeType As Integer     '0=binary, 1=binary search, 2=AVL, 3=m-way search, 4=B-tree
    Degree As Integer       'max degree(height) of the tree
    BranchFactor As Integer 'branching factor
End Type
Global TreeProp As TreeProperties
```

โดย *TreeType* จะเก็บชนิดของต้นไม้ที่ใช้อยู่ปัจจุบันเพื่อที่จะได้ตรวจสอบคุณสมบัติของต้นไม้ในขณะที่สร้างข้อมูลได้ถูกต้อง ส่วน *Degree* จะเก็บความสูงของต้นไม้ซึ่งจะนับจาก 0 คือมีแต่โหนดราก และ *BranchFactor* เก็บจำนวนโหนดลูกสูงสุดที่โหนดที่เป็นแม่สามารถมีได้

ส่วนที่เก็บรูปร่างของต้นไม้ที่ใช้รูปหลายเหลี่ยมแทนการเก็บโหนดของต้นไม้ซึ่งจะต้องมีการเก็บค่าของข้อมูล และตัวชี้เพื่อใช้บอกถึงตำแหน่งของโหนดลูกด้วยนั้น จะใช้โครงสร้างของข้อมูลที่เก็บเซตของจุดยอดของรูปหลายเหลี่ยมแทน โดยโครงสร้างชื่อ ***TreeData*** จะเก็บจุดยอดของรูปหลายเหลี่ยมซึ่ง *x* จะเก็บตำแหน่งในแนวแกนนอน และ *y* เก็บตำแหน่งในแนวแกนตั้ง

```
Type TreeData
    x As Long   'x and y position of each vertex in tree
    y As Long
End Type
```

การเก็บรูปหลายเหลี่ยม 1 รูป ในที่นี้จะเรียกเป็นวัตถุ (Object) 1 ก้อน หรือเป็นต้นไม้ต้นหนึ่งนั่นเอง โดยโครงสร้างข้อมูล ***TreeObjectType*** จะเก็บข้อมูลของรูปหลายเหลี่มได้ 1 รูป ซึ่งประกอบด้วย *Free* ซึ่งบอกถึงว่ารูปหลายเหลี่ยมนี้ได้มีการสร้างขึ้นมาหรือไม่ ส่วน *NoOfVertex* บอกถึงจำนวนจุดยอดของรูปหลายเหลี่ยมนี้ และ *Vertex* ซึ่งเป็นอาเรย์ของ ***TreeData*** จะเก็บจุดยอดทั้งหมดของรูปหลายเหลี่ยมนี้ไว้ ซึ่งในที่นี้สามารถเก็บได้ 100 จุดต่อรูปหลายเหลี่ยม 1 รูป ตามค่าที่กำหนดไว้ใน *MaxTreeVertex*

```
Global Const MaxTreeVertex = 100

Type TreeObjectType
    Free As Integer          'true=free(no object), false=object exist
    NoOfVertex As Integer    'number of vertex in tree
    Vertex(0 To MaxTreeVertex) As TreeData      'position of each vertex
End Type
```

ดังนั้นจะได้ว่า *TreeObject* ซึ่งเป็นอาเรย์ของ ***TreeObjectType*** จะเก็บรูปหลายเหลี่ยมที่สร้างขึ้นมาทั้งหมดไว้ เพื่อที่จะสามารถเลือกใช้งานได้สูงสุดคือ 20 รูปต่อแฟ้มข้อมูล 1 แฟ้ม นั่นหมายถึงผู้ใช้สามารถวาดรูปหลายเหลี่ยมได้สูงสุด 20 รูปเก็บไว้ที่เดียวกันได้ แต่จะสามารถเลือกมาได้ทีละ 1 รูป เพื่อทำการสร้างเป็นข้อมูลต้นไม้จริง ตามที่ได้กำหนดค่าไว้ใน *MaxTreeObject* ดังต่อไปนี้

Global Const *MaxTreeObject* = 20

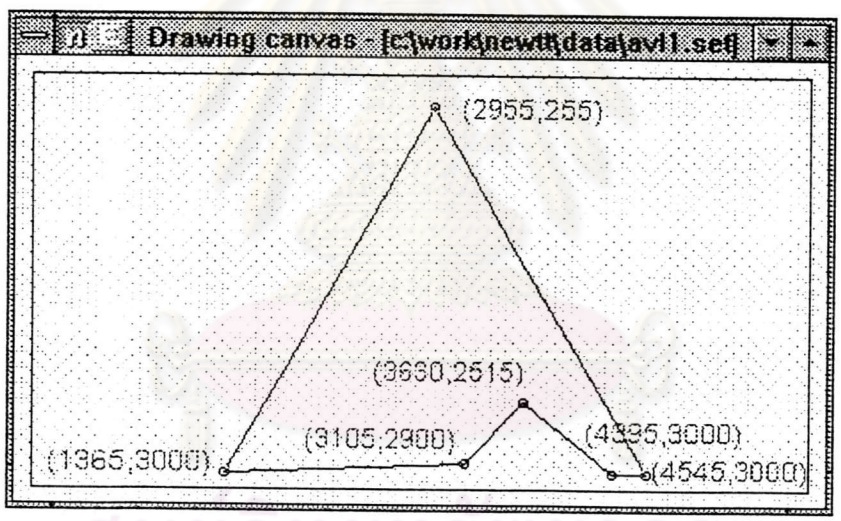
Global *TreeObject*(0 **To** *MaxTreeObject*) As ***TreeObjectType***

ตัวอย่างของการแทนข้อมูลแบบต้นไม้ด้วยภาพดังรูปที่ 3.2 ซึ่งเป็นต้นไม้แบบเอวีแอล ดังนั้นต้นไม้นี้จะมีลูกได้สูงสุดคือ 2 และมีความสูงสูงสุดที่เป็นไปได้คือ 5 จะได้ว่า **TreeProp** ซึ่งเก็บคุณสมบัติของต้นไม้จะมีค่าดังต่อไปนี้

TreeProp.TreeType = 2 ซึ่ง 2 ในที่นี้จะแทนต้นไม้ชนิดเอวีแอล

TreeProp.Degree = 5

TreeProp.BranchFactor = 2



รูปที่ 3.2 การแทนข้อมูลต้นไม้ด้วยรูป

ดังนั้นจะได้ว่า *TreeObject(0).Free* = False เพราะมีการสร้างรูปหลายเหลี่ยมขึ้นเพื่อแทนโครงสร้างข้อมูลแบบต้นไม้ และ *TreeObject(0).NoOfVertex* = 6 เนื่องจากรูปหลายเหลี่ยมนี้มีจุดยอดอยู่ 6 จุด และจะได้ลำดับของจุดยอดแต่ละจุดในรูปหลายเหลี่ยม วนตามเข็มนาฬิกาเป็นดังต่อไปนี้

TreeObject(0).Vertex(0).x = 2955

TreeObject(0).Vertex(0).y = 255

TreeObject(0).Vertex(1).x = 1365

TreeObject(0).Vertex(1).y = 3000

TreeObject(0).Vertex(2).x = 3105

TreeObject(0).Vertex(2).y = 2900

...

TreeObject(0).Vertex(5).x = 4545

TreeObject(0).Vertex(5).y = 3000

TreeObject(0).Vertex(6).x = 2955

TreeObject(0).Vertex(6).y = 255

จะเห็นว่ามีการเก็บจุดเพิ่มขึ้นมาเป็นพิเศษอีก 1 จุดเพื่อปิดรูปหลายเหลี่ยมกลับไปที่จุดเริ่มต้นของรูปหลายเหลี่ยมนี้เพื่อความสะดวกในการคำณวนจุดตัดของเส้นขอบของรูปหลายเหลี่ยมเพื่อหาตำแหน่งของโหนดและการเชื่อมต่อของโหนดภายในต้นไม้เมื่อมีการสั่งสร้างข้อมูลจริงขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย