หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

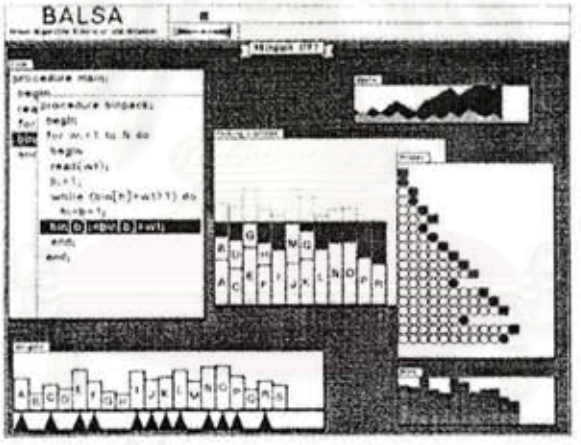
บทที่ 2

งานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงงานวิจัยสำหรับการจินตทัศน์อัลกอริทึมทั่วไป คือ BALSA-II[1], Tango[2] และ AVis[3] และ ระบบจินตทัศน์อัลกอริทึมสำหรับปัญหาด้านเรขาคณิต คือ GASP[4] และ XYZ Geobench[5] เพื่อให้ผู้ที่สนใจในด้านการจินตทัศน์อัลกอริทึมทราบถึงแนวคิดในการออกแบบระบบจินตทัศน์ และระบบได้รับการพัฒนาอยู่ในปัจจุบัน สำหรับเป็นแนวทางในการออกแบบระบบจินตทัศน์อัลกอริทึม หรือการสร้างการจินตทัศน์อัลกอริทึม

2.1 BALSA-II[6]

BALSA-II เป็นระบบจินตทัศน์อัลกอริทึม ทำงานบนเครื่องแมคอินทอช พัฒนาต่อจาก BALSA ซึ่งโดยเริ่มแรกมีจุดประสงค์เพื่อช่วยในการเรียนการสอนในวิชาทางด้านวิทยาศาสตร์คอมพิวเตอร์ เช่น โครงสร้างข้อมูล กราฟิกส์ วิชาการเขียนโปรแกรม ตัวอย่างกลุ่มของปัญหาใน BALSA-II เช่น การเรียงลำดับ การค้นหา กราฟ เรขาคณิตเชิงคำนวณ เป็นต้น



รูปที่ 2-1 แสดงหน้าต่างของส่วนแสดงผลหลาย ๆ รูปแบบใน BALSA-II[7]

---

[1] Brown, "Exploring algorithms using Balsa-II," IEEE Computer (1988) : 14-36.

[2] Stasko, "Tango: a framework and system for algorithm animation," IEEE Computer (1990) :27-39.

[3] ชัชวาล วงศ์ศิริประเสริฐ, "การออกแบบและพัฒนาแก่นกลางของระบบจินตทัศน์อัลกอริทึม," (วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, 2539).

[4] Tal and Dobbin, "Visualization of geometric algorithm," IEEE Transactions on Visualiztion and Computer Graphics 1(1995) : 194-204.

[5] Schorn, "The XYZ GeoBench : A programming environment for geometric algorithms," LNCS 553,(1991):187-202.

[6] Brown, "Exploring algorithms using Balsa-II," IEEE Computer (1988) : 14-36.

[7] Brown, Algorithm animation (MA:MIT Press, 1988), p. 16

ใน BALSA-II ผู้ใช้สามารถดูการประมวลผลอัลกอริทึมผ่านการแสดงผลในรูปแบบต่าง ๆ แสดงดังรูปที่ 2-1 โดยการแสดงผลแต่ละแบบจะแสดงในหน้าต่างของส่วนแสดงผลที่ผู้ใช้สามารถปรับขนาด เปลี่ยนตำแหน่ง หรือขยายขนาดของการแสดงผลเพื่อดูส่วนรายละเอียด นอกจากนี้ใน BALSA-II ยังมีการกำหนดการทำงานล่วงหน้าในรูปแบบของบทการจินตทัศน์ (script) ซึ่งสามารถจัดเก็บและเล่นซ้ำได้

2.1.1 แนวคิดต้นแบบ

ใน BALSA-II แบ่งผู้ใช้ที่เกี่ยวข้องกับระบบเป็น 2 แบบ คือผู้ใช้ปลายทาง และนักเขียนโปรแกรม จุดมุ่งหมายหลักของระบบในส่วนของผู้ใช้ปลายทาง คือ จัดเตรียมรูปแบบที่สอดคล้องกันในการจินตทัศน์ และไม่ขึ้นอยู่กับกลุ่มของปัญหา หรือผู้ที่เตรียมการจินตทัศน์ เมื่อผู้ใช้เคยทำการจินตทัศน์สำหรับอัลกอริทึมหนึ่ง ๆ แล้ว สามารถที่จะทราบวิธีทำการจินตทัศน์สำหรับอัลกอริทึมอื่น ๆ ได้ และจุดมุ่งหมายหลักในส่วนของนักเขียนโปรแกรม คือการจัดเตรียมส่วนที่จำเป็นในการสร้างการจินตทัศน์โดยที่นักเขียนโปรแกรมไม่จำเป็นต้องสร้างหรือเขียนโปรแกรมส่วนที่ใช้ร่วมกันได้ใหม่ นอกจากนี้โปรแกรมส่วนของอัลกอริทึม ส่วนสร้างข้อมูล และส่วนที่ใช้แสดงผลควรมีอิสระต่อกัน

2.1.1.1 ผู้ใช้ปลายทาง

ผู้ใช้ปลายทางจะใช้ระบบในงานสองส่วน คือ การจัดเตรียม และการเรียกใช้ ในขั้นตอนการจัดเตรียมผู้ใช้ปลายทางสามารถเลือกอัลกอริทึมที่จะเรียกใช้ ชนิดของข้อมูลเข้า และส่วนแสดงผลที่จะใช้ เลือกค่าพารามิเตอร์สำหรับแต่ละส่วนประกอบ ซึ่งแต่ละอัลกอริทึมจะมีค่าเริ่มต้นที่สามารถปรับเปลี่ยนได้โดยผู้ใช้ปลายทาง ในขั้นตอนการเรียกใช้ ผู้ใช้ปลายทางทำการเรียกใช้อัลกอริทึมและดูการทำงานในหน้าต่างส่วนแสดงผล ขณะที่อัลกอริทึมทำงาน ผู้ใช้ปลายทางสามารถหยุดการทำงานชั่วคราวแล้วเปลี่ยนเป็นส่วนแสดงผลอื่นในชุดเดียวกัน หรือปรับเปลี่ยนความเร็วได้

2.1.1.2 นักเขียนโปรแกรม

นักเขียนโปรแกรมเป็นผู้เขียนโปรแกรมในภาษาระดับสูง ทำการแก้ไขและแปลโปรแกรมภายนอก BALSA-II แล้วจึงรวมในตัวประมวลผลก่อนของ BALSA-II (BALSA-II preprocessor) จากนั้นโปรแกรมจะได้รับการเชื่อมโยงภายใน BALSA-II โดยโปรแกรมที่จะทำการจินตทัศน์จะแบ่งออกเป็นส่วนประกอบย่อย ๆ 3 ส่วนคือ ส่วนอัลกอริทึม ส่วนสร้างข้อมูล และส่วนแสดงผล ในแต่ละส่วนพัฒนาแยกจากกัน ความสัมพันธ์ระหว่างองค์ประกอบต่าง ๆ แสดงดังรูปที่ 2-3

1 *ส่วนประกอบอัลกอริทึม* ในส่วนประกอบอัลกอริทึมจะมีงานที่ต้องจัดการสองส่วน คือการหาเหตุการณ์ที่น่าสนใจซึ่งเป็นจุดที่ทำการติดต่อกับส่วนประกอบอื่น และการจัดโครงสร้างโปรแกรมซึ่งเป็นส่วนที่ได้รับการเรียกจาก BALSA-II ในการตอบสนองกับผู้ใช้

1.1 เหตุการณ์ที่น่าสนใจแบ่งออกเป็น เหตุการณ์ขาเข้า (Input event) และเหตุการณ์ขาออก (Output event) โดยเหตุการณ์ขาเข้าจะเปรียบเสมือนกับเป็นคำสั่งรับข้อมูลในการเขียนโปรแกรมปกติ ดังนั้นเหตุการณ์ขาเข้าจึงเกี่ยวกับการขอข้อมูล และเหตุการณ์ขาออกจะ

เสมือนกับคำสั่งในการแสดงผลลัพธ์ ดังนั้นเหตุการณ์ขาออกจึงเกี่ยวข้องกับการแสดงสถานะของโปรแกรม หรือแสดงผลลัพธ์ในโปรแกรม โดยในระบบเหตุการณ์ขาออกจะทำให้ส่วนแสดงผลได้รับการแจ้งเตือนจากอัลกอริทึมทำให้เกิดการปรับปรุงภาพที่แสดงบนหน้าจอ ซึ่งส่วนของอัลกอริทึมจะไม่มีส่วนเกี่ยวข้องกับการจัดการด้านการแสดงผลกราฟิกส์ ส่วนเหตุการณ์ขาเข้าทำให้ส่วนสร้างข้อมูลได้รับการแจ้งเตือนให้ส่งข้อมูลบางอย่างกลับไปให้แก่ส่วนประกอบอัลกอริทึม

1.2 การจัดโครงสร้างโปรแกรมของอัลกอริทึม จะแบ่งออกเป็นรูทีนย่อยสำหรับควบคุมรหัสของอัลกอริทึม พารามิเตอร์ของอัลกอริทึม รวมทั้งการเริ่มต้นและการจบการทำงาน รูทีนย่อยเหล่านี้จะถูกเรียกโดยระบบจินตทัศน์อัลกอริทึม ส่วนของการจัดโครงสร้างและเหตุการณ์ขาเข้าและเหตุการณ์ขาออกแสดงดังตัวอย่างในรูปที่ 2-2

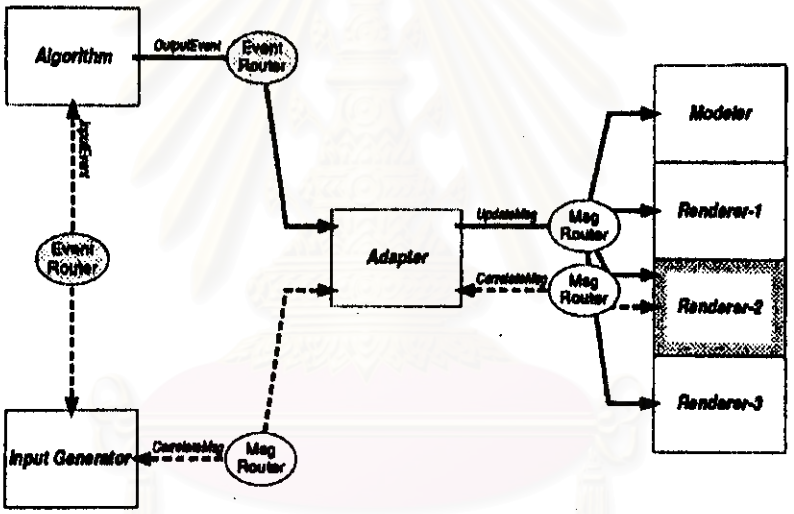
```
module Alg.Quicksort;
  var M : Integer;  { algorithm parameter }
  procedure Alg.Quicksort.Create;
    { This routine is called by BALSA-II once, before any of the other routines are called. }
    begin
    M := 1;  { default value of parameter }
    end;
  procedure Alg.Quicksort.Dispose;
    { This routine is called by BALSA-II once, after all of the other routines are called. }
    begin end;
  procedure Alg.Quicksort.StartRun;
    { This routine is called by BALSA-II each time the Quicksort algorithm begins running. }
    begin end;
  procedure Alg.Quicksort.EndRun;
    { This routine is called by BALSA-II each time the Quicksort algorithm finishes running. }
    begin end;
  procedure Alg.Quicksort.Parms;
    { This routine is called by BALSA-II each time the user wants to observe or change }
    { the "algorithm parameters" for the Quicksort algorithm. }
    var newM : Integer;
    begin
    Writeln('Current cutoff for small files is ', M);
    If BALSAModifyParmsFg then
      begin
      Writeln('Enter new cutoff:'); Readln(newM);
      if newM > 0 then M := newM
      end
    end;
  procedure Alg.Quicksort.Code;
    { Here is what one intuitively thinks of as "The Quicksort Algorithm." }
    { This routine is called by BALSA-II each time the user issues a run command, }
    { preceded by a call to ...StartRun and followed by call to ...EndRun routine. }
      See Fig. 22
end.
```

รูปที่ 2-2 แสดงการจัดโครงสร้างของโปรแกรมการเรียงลำดับแบบเร็ว[8]

[8] Brown, "Exploring algorithms using Balsa-II," IEEE Computer (1988) : 14-36.

2 *ส่วนสร้างข้อมูล* จะประกอบด้วยชุดของรูทีนย่อยคล้ายกับส่วนของอัลกอริทึม ซึ่งประกอบด้วยรูทีนซึ่งจัดการพารามิเตอร์ เริ่มต้นและจบการทำงานนอกจากนี้ยังมีกลุ่มของรูทีนที่สัมพันธ์กับเหตุการณ์ขาเข้า ผู้ใช้สามารถใส่ข้อมูลผ่านทางหน้าต่างส่วนแสดงผลให้กับตัวสร้างข้อมูล ในกรณีนี้จะมีการส่งข้อความคำสั่งจากส่วนแสดงผลให้กับตัวสร้างข้อมูล (correlate message)

3 *ส่วนแสดงผล* ส่วนทั่วไปของโปรแกรมในส่วนแสดงผลจะคล้ายกับทั้งสองส่วนที่กล่าวมาแล้ว แต่ในส่วนแสดงผลมีการแบ่งโครงสร้างออกเป็น ส่วนกำหนดรูปแบบ (modeler) ส่วนแสดงภาพ (renderer) โดยส่วนกำหนดรูปแบบจะสร้างและรักษาต้นแบบที่แทนข้อมูลที่ใช้ ส่วนแสดงภาพเป็นส่วนของการแสดงภาพบนหน้าจอ และตัวแปลงคำสั่ง (adapter) ซึ่งทำหน้าที่แปลงเหตุการณ์ขาออกของอัลกอริทึมเป็นข้อความคำสั่งปรับปรุงภาพ (update message) ที่ส่วนกำหนดรูปแบบและส่วนแสดงผลเข้าใจ ในโปรแกรมของส่วนกำหนดรูปแบบ ส่วนแสดงภาพ และตัวแปลงคำสั่งจะมีการแบ่งเป็นรูทีนย่อย ๆ เหมือนตัวสร้างข้อมูลและส่วนอัลกอริทึม



รูปที่ 2-3 ภาพรวมความสัมพันธ์ระหว่างองค์ประกอบภายในระบบ[9]

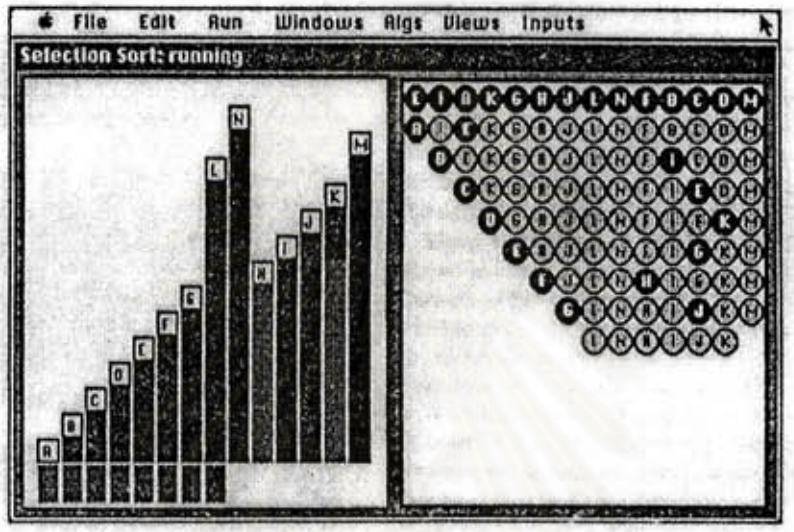
2.1.2 ส่วนต่อประสานกับผู้ใช้ใน BALSA-II

ใน BALSA-II ใช้การต่อประสานกับผู้ใช้ในรูปแบบของแมคอินทอช ซึ่งมีเมนูสำหรับแฟ้มบทการจินตทัศน์ เมนูสำหรับเลือกอัลกอริทึม ส่วนแสดงผลและส่วนสร้างข้อมูล เมนูสำหรับการจินตทัศน์ ผู้ใช้สามารถจัดเรียงหน้าต่างการแสดงผล เพื่อแสดงผลหลาย ๆ แบบได้ สำหรับเมนูต่าง ๆ ใน BALSA-II แสดงรูปที่ 2-4

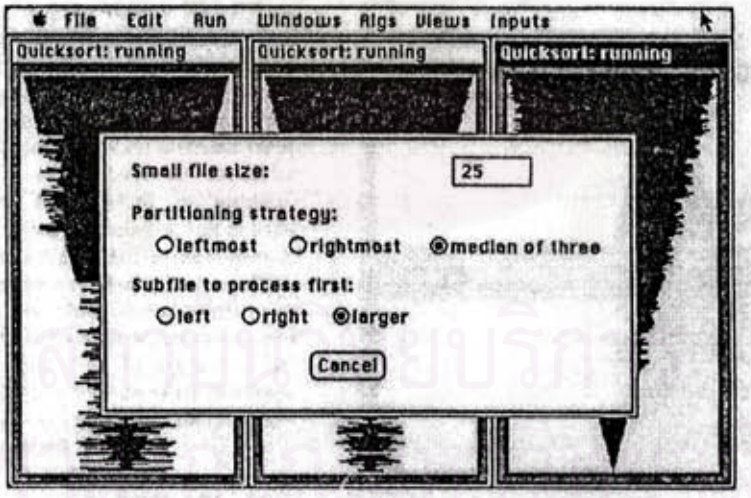
นอกจากการที่ผู้ใช้สามารถเลือกส่วนประกอบต่าง ๆ เพื่อทำการจินตทัศน์แล้ว ยังสามารถปรับเปลี่ยนพารามิเตอร์ของส่วนประกอบต่าง ๆ เพื่อดูผลการทำงานหรือคุณสมบัติบางอย่างของอัลกอริทึมที่เปลี่ยนไป ตัวอย่างพารามิเตอร์ของอัลกอริทึมที่มีผลต่อการทำงานของอัลกอริทึม เช่น การกำหนดขนาดตารางแฮช หรือ

[9] Brown, Algorithm animation (MA:MIT Press, 1988), p. 10.

ในส่วนสร้างข้อมูลจะกำหนดให้ข้อมูลเป็นแบบสุ่ม หรือมีการเรียงลำดับ จากรูปที่ 2-5 เป็นการกำหนดพารามิเตอร์สำหรับอัลกอริทึมการเรียงลำดับแบบเร็ว การกำหนดพารามิเตอร์ไม่สามารถทำได้ตลอดเวลา หากแต่ทำตอนเริ่มต้น หรือถ้าในขณะเรียกใช้จะเป็นการตอบสนองการร้องขอของอัลกอริทึม เช่น การกำหนดโหนดที่ต้องการลบในต้นไม้ทวิภาค



รูปที่ 2-4 เมนูและหน้าต่างส่วนแสดงผลของ BALSA-II[10]



รูปที่ 2-5 ภาพการกำหนดพารามิเตอร์สำหรับอัลกอริทึมการเรียงลำดับแบบเร็ว[11]

ผู้ใช้สามารถควบคุมการประมวลผลในระหว่างการจินตทัศน์ได้ โดยการกำหนดจุดหยุด (stop-point) หรือกำหนดการทำงานทีละขั้น และสามารถดูค่าเวลาใช้สำหรับแต่ละอัลกอริทึมโดยพิจารณาจากการดำเนินการพื้นฐานของอัลกอริทึม

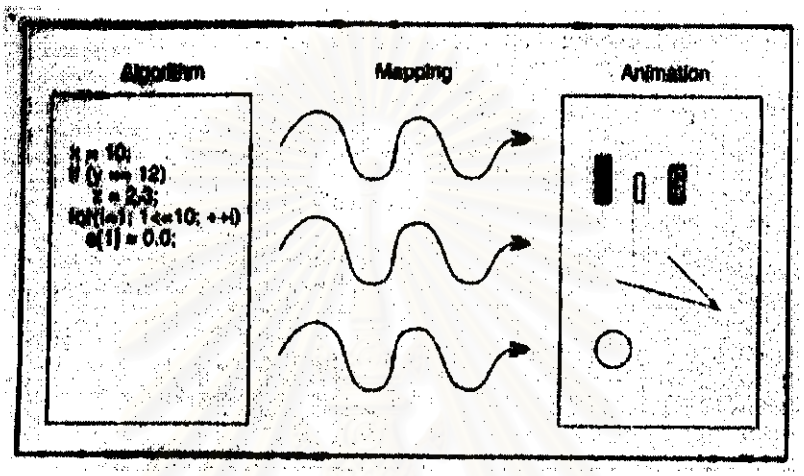
---

[10] Ibid., p. 48.

[11] Ibid., p. 58

2.2 *Tango*[12]

Tango เป็นระบบสำหรับการจินตทัศน์อัลกอริทึมที่นำเสนอรูปแบบการเปลี่ยนตามเส้นทาง (Path transition paradigm) ซึ่งง่ายแก่ผู้พัฒนาในการระบุการเปลี่ยนแปลงอย่างต่อเนื่องของเหตุการณ์ภายในโปรแกรม โดยสามารถกำหนดเส้นทางของวัตถุที่ทำการจินตทัศน์ให้มีการเคลื่อนที่เปลี่ยนตำแหน่งตามต้องการ และยังสามารถกำหนดให้มีการเปลี่ยนแปลงกับวัตถุในขณะที่เคลื่อนที่ได้



รูปที่ 2-6 เค้าโครงส่วนประกอบของระบบ[13]

2.2.1 ส่วนประกอบของระบบ

ใน Tango ได้แบ่งการจินตทัศน์อัลกอริทึมออกเป็นส่วนประกอบ 3 ส่วนดังรูปที่ 2-6 คือ ส่วนประกอบอัลกอริทึม ส่วนประกอบการจินตทัศน์ (animation component) และส่วนประกอบเชื่อมโยง (mapping component) ซึ่งจะทำการเชื่อมโยงส่วนของอัลกอริทึมเข้ากับภาพเคลื่อนไหวที่ใช้ในการจินตทัศน์

2.2.1.1 ส่วนประกอบอัลกอริทึม

ส่วนประกอบอัลกอริทึม เป็นส่วนการทำงานของอัลกอริทึม ในส่วนนี้จะมีการระบุตำแหน่งในโปรแกรมที่สัมพันธ์กับการดำเนินการที่สำคัญ ซึ่งแสดงความหมายหรือแนวคิดของโปรแกรม ตัวอย่างเช่น ในอัลกอริทึมของการเรียงลำดับ การดำเนินการที่สำคัญ คือ การเปรียบเทียบและการสลับที่ ณ ตำแหน่งที่ระบุนี้จะมีการกำหนดพารามิเตอร์ โดยเมื่อมีการประมวลผลมาถึงตำแหน่งนี้จะมีการส่งพารามิเตอร์นี้ให้กับส่วนประกอบเชื่อมโยงและส่งต่อไปยังส่วนประกอบการจินตทัศน์

2.2.1.2 ส่วนประกอบการจินตทัศน์

ส่วนนี้จะประกอบด้วยวัตถุกราฟิกส์ ซึ่งเปลี่ยนแปลงตำแหน่ง สี หรือขนาด ในขณะทำการจินตทัศน์และการดำเนินการสำหรับควบคุมการเปลี่ยนแปลงในการจำลองเหตุการณ์บางอย่าง โดยประกอบไปด้วยแบบ

---

[12] Stasko, "Tango: a framework and system for algorithm animation," IEEE Computer (1990) :27-39.

[13] Ibid.

ชนิดข้อมูลนามธรรม (abstract data type) 4 ชนิด คือ ภาพ (image) ตำแหน่ง (location) เส้นทาง (path) และ การเปลี่ยน (transition)

1. *ภาพ* คือ วัตถุกราฟิกส์ ซึ่งมีการเปลี่ยนแปลงตำแหน่ง ขนาดและสีในขณะทำการจินตทัศน์ ตัวอย่างของภาพ เช่น เส้นตรง รูปสี่เหลี่ยม วงกลม ข้อความ
2. *ตำแหน่ง* คือ ตำแหน่งที่กำหนดด้วยคู่ลำดับ (x,y) เพื่อให้สามารถเก็บและอ้างอิงตำแหน่งเฉพาะสำหรับการแสดงภาพ
3. *เส้นทาง* คือ ส่วนที่กำหนดการเปลี่ยนแปลงในแต่ละกรอบ (frame) โดยภาพจะได้รับปรับปรุงตลอดเส้นทาง เช่น ภาพอาจถูกย้ายตำแหน่งหรือเปลี่ยนสี ซึ่งภาพที่ปรากฏจะถูกเปลี่ยนแปลงไปตามเส้นทาง เส้นทางจะกำหนดโดยชุดลำดับของพิกัด (x,y) ซึ่งแต่ละคู่ลำดับจะแทนระยะจากตำแหน่งก่อนหน้า ความยาวของเส้นทาง คือจำนวนของคู่ลำดับที่รวมกันเป็นเส้นทาง
4. *การเปลี่ยน* จะใช้เส้นทางเป็นพารามิเตอร์ในการปรับปรุงตำแหน่ง หรือลักษณะที่ปรากฏของภาพ เพื่อสร้างภาพการเคลื่อนไหว เช่น การเคลื่อนย้าย การปรับขนาด การเติมสี ชนิดของการเปลี่ยนที่ต่างกันจะใช้พารามิเตอร์ของเส้นทางที่ต่างกัน เช่น ในการเปลี่ยนแบบเคลื่อนย้าย (move transition) ระยะห่างของ เส้นทางจะกำหนดว่าจะเคลื่อนย้ายภาพอย่างไรก่อนจะถึงกรอบถัดไป การเปลี่ยนแบบเติม (fill transition) จะใช้ค่า x ของระยะห่างของเส้นทางในการบวกเพิ่มเข้าค่าที่เติมสีของภาพซึ่งมีค่าตั้งแต่ 0.0 (เติมเค้าร่าง) จนถึง 1.0 (เติมสีทึบ)

   ตัวอย่างของการดำเนินการแสดงดังในตารางที่ 2-1

Image	Location	Path		Transition
Create	Create	Create	Rotate	Create
Locate	X	Load	Scale	Concatenate
	Y	Store	Example	Iterate
	Modify	Length	Motion	Compose
	Equal	Dx	Distance	Perform
		Dy	Concatenate	
		MakeType	Iterate	
		Null	Compose	
		Copy	AddHead	
		Color	AddTail	
		Extend	DeleteHead	
		Interpolate	DeleteTail	

ตารางที่ 2-1 แสดงการดำเนินการของส่วนประกอบการจินตทัศน์[14]

ผู้พัฒนาจะทำการสร้างส่วนการจินตทัศน์ด้วยการรวมภาพ ตำแหน่ง เส้นทาง การเปลี่ยน และการดำเนินการที่ทำให้เกิดภาพเคลื่อนไหวที่ต้องการ

2.2.1.3 ส่วนประกอบเชื่อมโยง (mapping component)

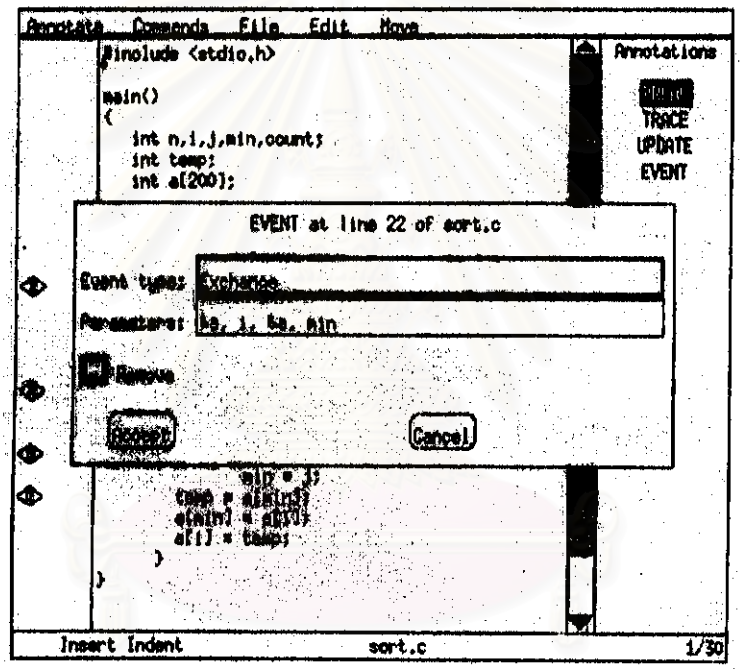
ส่วนประกอบเชื่อมโยง เป็นส่วนประกอบที่ทำการเชื่อมโยงโปรแกรม กับส่วนของการจินตทัศน์ ซึ่งประกอบด้วยส่วนของการสร้างความสัมพันธ์ (association) และการเชื่อมโยงจากอัลกอริทึมไปยังส่วนของภาพการเคลื่อนไหว

---

[14] Ibid.

ส่วนของการสร้างความสัมพันธ์ จะเป็นส่วนที่ให้ผู้พัฒนาเชื่อมต่อวัตถุ เช่น ภาพ ค่าของข้อมูล ตำแหน่ง กับชุดของพารามิเตอร์จากโปรแกรม ในการสร้างความสัมพันธ์จะต้องกำหนดชื่อที่ไม่ซ้ำ โดยจะมีพารามิเตอร์หรือไม่ก็ได้ ในระบบใช้วิธีแฮชชิงโดยคีย์ของวัตถุข้อมูล คือ ชื่อที่ใช้ในการสร้างความสัมพันธ์ และรายการของพารามิเตอร์ ค่าพารามิเตอร์นี้โดยทั่วไป คือ ค่าของข้อมูลที่ได้รับการเชื่อมโยงจากโปรแกรม ตัวอย่างของการสร้างความสัมพันธ์แสดงดังรูปที่ 2-7 เป็นการสร้างความสัมพันธ์ระหว่างการดำเนินการชื่อ Exchange มีพารามิเตอร์ 4 ตัว

ส่วนของการเชื่อมโยงการดำเนินการกับส่วนการจินตทัศน์ ในระบบนี้ใช้ ตัวแบบเครื่องเลียนแบบสิ่งมีชีวิตสถานะจำกัด (Finite state automaton model) ซึ่งสามารถจับคู่ความสัมพันธ์ได้หลายแบบ ผู้พัฒนาสามารถระบุการเชื่อมโยงได้อย่างอิสระจากการสร้างภาพเคลื่อนไหว



รูปที่ 2-7 รูปแสดงการสร้างความสัมพันธ์[15]

2.2.2 การทำงานใน Tango

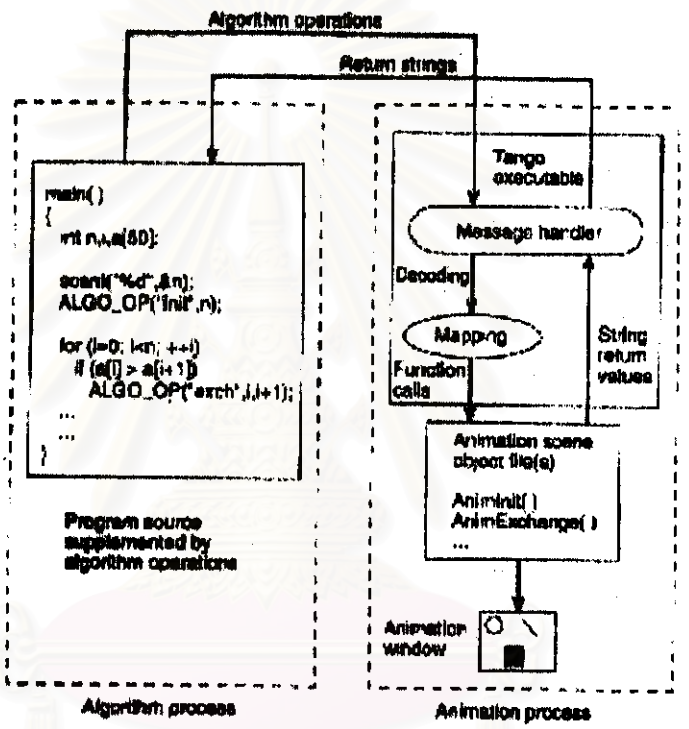
ใน Tango จะมีการจัดการในลักษณะของการทำงานร่วมกันของส่วนประกอบต่าง ๆ ส่วนประกอบใน Tango จะประกอบด้วย โปรแกรมที่จะทำการจินตทัศน์ รหัส (code) ควบคุมการจินตทัศน์ และโปรแกรม Tango โครงสร้างในลักษณะนี้ทำให้ไม่ต้องทำการแปลโปรแกรมใหม่ทั้งหมด เมื่อมีการเปลี่ยนแปลงส่วนใดส่วนหนึ่ง ส่วนประกอบในระบบจะติดต่อกันโดยวิธีการสื่อสารระหว่างการประมวลผล (IPC) และการเรียกฟังก์ชัน (function call) ในการประมวลผลโปรแกรมและสร้างการจินตทัศน์ ภายใต้ Tango จะใช้กระบวนการ (process) สองกระบวนการ ใน Unix มีการติดต่อดังรูปที่ 2-8

---

[15] Ibid.

ในกระบวนการการจินตทัศน์ เมื่อผู้ใช้เริ่มต้น Tango ระบบจะอ่านแฟ้มควบคุม (control file) ซึ่งระบุพารามิเตอร์สำหรับการจินตทัศน์ เช่น การดำเนินการของอัลกอริทึม และชื่อของภาพการเคลื่อนไหวแล้วทำการบรรจุ (load) แฟ้มรายละเอียดการจินตทัศน์ ซึ่งมีรหัสการจินตทัศน์ที่กำหนดโดยผู้ออกแบบ แล้วสร้างหน้าต่างบนหน้าจอสำหรับแสดงการจินตทัศน์จากนั้นจะรอการดำเนินการของอัลกอริทึมจากโปรแกรมที่ทำการจินตทัศน์

ในกระบวนการอัลกอริทึม เมื่อผู้ใช้เรียกใช้โปรแกรมที่จะทำการจินตทัศน์ โปรแกรมนี้จะต้องประกอบด้วยการดำเนินการของอัลกอริทึม ซึ่งจะถูกแจกจ่ายไปยังกระบวนการการจินตทัศน์ในขณะที่เกิดการดำเนินการนั้นขึ้น



รูปที่ 2-8 ภาพรวมการติดต่อระหว่าง 2 กระบวนงานใน Tango[16]

นอกจากนี้ Tango ได้จัดเตรียมวิธีการรับข้อมูลเข้า สำหรับการส่งข้อมูลจากกระบวนการการจินตทัศน์กลับไปยังกระบวนการอัลกอริทึม โดยระบบมีกระบวนงานซึ่งให้ผู้ใช้ เลือกพิกัดในหน้าต่าง และวัตถุรูปภาพโดยใช้เมาส์ ข้อมูลที่ถูกเลือกในภาพการเคลื่อนไหวจะถูกจัดรูปแบบเป็นสายอักขระ และส่งผ่านกลับไปยังการดำเนินการของอัลกอริทึม

2.2.3 การสร้างการจินตทัศน์ใน Tango

ในการสร้างภาพเคลื่อนไหวโดยใช้ Tango ประกอบด้วยขั้นตอนดังนี้

1. การระบุการดำเนินการของอัลกอริทึม โดยสร้างหมายเหตุประกอบ

---

[16] Ibid.

2. ออกแบบภาพการเคลื่อนไหว สำหรับการเขียนโปรแกรมในส่วนของการสร้างการเคลื่อนไหวของภาพ
3. สร้างแฟ้มควบคุม เพื่อระบุการเชื่อมโยงจากการดำเนินการของอัลกอริทึมไปยังภาพการเคลื่อนไหว

การกำหนดการดำเนินการของอัลกอริทึมใน Tango สามารถสร้างหมายเหตุประกอบได้สองวิธี คือใช้ตัวแก้ไขข้อความ (text editor) หรือตัวแก้ไขเขตข้อมูลหมายเหตุประกอบ (Field's annotation editor) หลังจากเพิ่มการดำเนินการของอัลกอริทึมที่ต้องการแล้ว ต้องทำการแปลโปรแกรมใหม่

การออกแบบภาพการเคลื่อนไหว ใน Tango ใช้ภาษาซีสำหรับชนิดข้อมูลนามธรรมทั้งสี่แบบข้างต้น การดำเนินการของชนิดข้อมูล คือฟังก์ชันในภาษาซี ซึ่งการสร้างภาพเคลื่อนไหวก็คือการเรียกใช้ฟังก์ชันภาษาซีด้วยพารามิเตอร์ที่ได้รับจากการดำเนินการของอัลกอริทึม ตัวอย่างภาพการเคลื่อนไหวซึ่งค้นคืนภาพและตำแหน่งของวัตถุ แล้วเคลื่อนย้ายภาพเป็นเส้นทางครึ่งวงกลม แสดงดังรูปที่ 2-9

```
MoveTo(locid,locnum,imid,imnum)
int locid,locnum,imid,immum
{
      Location       formloc,toloc;
      Image  image;
      Path   path1,path2;
      Transition    mover;
      toloc = AssoRetrieve("ID",locid,locnum);
      image = AssoRetrieve("ID",imid,imnum);
      fromloc = ImageLoc(image,Center);
      path1 = PathMakeType(Clockwise);
      path2 = PathExample(fromloc,toloc,path1);
      mover = TransCreate(Move,image,path2);
      TransPerform(mover);
}
```

รูปที่ 2-9 โปรแกรมภาพการเคลื่อนไหว[17]

การเชื่อมโยงระหว่างการดำเนินการของอัลกอริทึมและภาพการเคลื่อนไหว ผู้ใช้จะระบุการดำเนินการและเชื่อมโยงกับภาพการเคลื่อนไหว โดยใช้แฟ้มควบคุมซึ่งมีการจับคู่ระหว่างการดำเนินการและภาพการเคลื่อนไหว ในแฟ้มควบคุมจะประกอบด้วยข้อมูล 5 ส่วนแบ่งด้วยเครื่องหมาย %% โดยส่วนแรกคือ ขนาดหน้าต่างที่แสดงการจินตทัศน์ ส่วนที่สอง คือแฟ้มจุดหมาย (object file) ที่มีการกำหนดภาพการเคลื่อนไหวซึ่งจะบรรจุแบบพลวัตสำหรับการจินตทัศน์ ส่วนที่สาม คือชื่อของการดำเนินการในโปรแกรมและชนิดข้อมูลของพารามิเตอร์ ส่วนที่สี่ คือรายการของภาพการเคลื่อนไหวสำหรับการจินตทัศน์ ซึ่งภาพการเคลื่อนไหวนี้จะกำหนดเป็นฟังก์ชันภาษาซี ดังแสดงในรูปที่ 2-9 อยู่ในแฟ้มในส่วนที่สอง สำหรับส่วนสุดท้าย คือการกำหนดการจับคู่ของการดำเนินการกับภาพการเคลื่อนไหว ตัวอย่างส่วนต่าง ๆ ของแฟ้มควบคุมแสดงดังรูปที่ 2-10

การที่สามารถกำหนดการเชื่อมโยงในแฟ้มแยกต่างหาก ทำให้เกิดความยืดหยุ่นในการเปลี่ยนแปลงการเชื่อมโยง เช่น เมื่อต้องการเชื่อมโยงอัลกอริทึมเข้ากับภาพการเคลื่อนไหวใหม่ สามารถทำได้โดยการแก้ไขแฟ้มควบคุมเท่านั้นไม่ต้องทำการแปลโปรแกรมใหม่

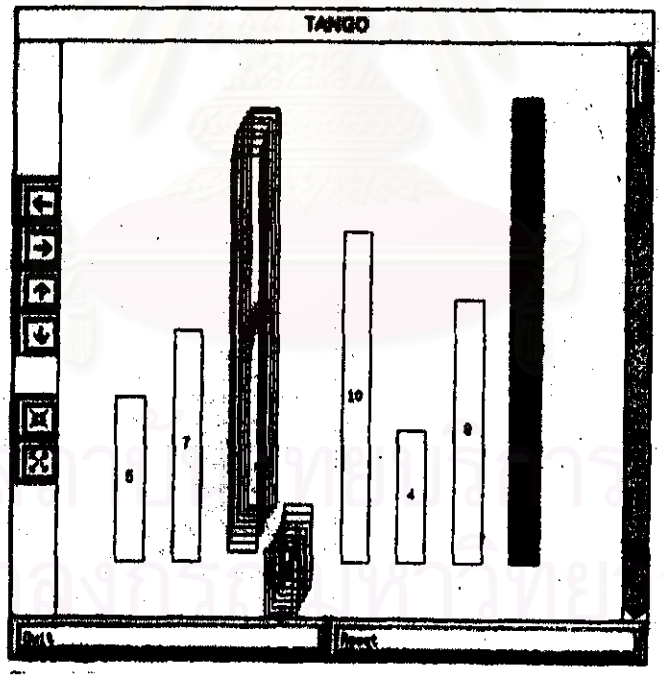
---

[17] Ibid.

```
0.0 1.0 1.0
%%
myscene.o animlib.o
%%
Init  %d  %f
Check  %d  %d
Update  %d
Complete  %d  %s
%%
ANIMInit
Display
ANIMComplete
Finish
%%
Init → ANIMInit
Check Check Check → Display
Update → Update
Complete → ANIMComplete Finish
```

รูปที่ 2-10 ตัวอย่างแฟ้มควบคุม[18]

2.2.4 ตัวอย่างการจินตทัศน์ใน Tango



รูปที่ 2-11 ภาพการจินตทัศน์อัลกอริทึมการเรียงลำดับแบบฟอง[19]

---

[18] Ibid.

[19] Ibid.

ภาพการจินตทัศน์ใน Tango แสดงดังรูปที่ 2-11 สำหรับอัลกอริทึมการเรียงลำดับแบบฟอง (bubble sort) ในหน้าต่างการจินตทัศน์ของ Tango มีคำสั่งแสดงในลักษณะของสัญรูป (icon) ซึ่งผู้ใช้สามารถปรับความเร็วในการจินตทัศน์ หยุดการจินตทัศน์ชั่วคราว ขยายภาพที่ทำการจินตทัศน์ได้

2.3 *Geometric Animation System, Princeton (GASP)*[20]

GASP เป็นระบบจินตทัศน์อัลกอริทึมสำหรับปัญหาทางด้านเรขาคณิตเชิงคำนวณโดยเฉพาะ ทำงานภายใต้ระบบ Unix บน Silicon graphics Iris เป็นระบบซึ่งมุ่งเน้นในการสร้างการจินตทัศน์อัลกอริทึมด้านเรขาคณิตในสามมิติ โดยลดภาระของนักเขียนโปรแกรมในการจัดการการแสดงผลด้านกราฟิกส์ โดย GASP เลือกใช้กลุ่มของปัญหาเล็ก ๆ เพื่อให้สามารถสร้างระบบจินตทัศน์อัลกอริทึมซึ่งผู้ใช้สามารถใช้ได้ง่าย ในกลุ่มของปัญหาที่ได้รับการนิยามอย่างดี จะทำให้สามารถใช้ความรู้เกี่ยวกับประเภทของวัตถุและการดำเนินการซึ่งต้องการทำการจินตทัศน์ ทำให้การสร้างภาพเคลื่อนไหวเป็นไปอย่างอัตโนมัติ ผู้ใช้เพียงแต่กำหนดการดำเนินการที่ต้องการทำการจินตทัศน์ โดยไม่จำเป็นต้องทราบวิธีสร้างภาพเคลื่อนไหว

2.3.1 แนวคิดต้นแบบ

ในระบบจินตทัศน์อัลกอริทึมเฉพาะด้านที่มีการจำกัดกลุ่มของปัญหา ทำให้สามารถแยกระหว่างการจินตทัศน์และอัลกอริทึมได้ง่ายขึ้น เนื่องจากรายการและการดำเนินการของรายการต่าง ๆ ค่อนข้างแน่นอน สามารถเลือกการแสดงผลที่เหมาะสมได้ง่าย ดังนั้นงานในส่วนการแสดงผลสามารถทำให้เป็นแบบอัตโนมัติได้ แนวคิดในการออกแบบต้องการให้ผู้พัฒนามีอิสระจากการต้องออกแบบและเขียนโปรแกรมในส่วนแสดงผล เนื่องจากงานเขียนโปรแกรมในส่วนการจินตทัศน์ค่อนข้างใช้เวลามาก และมีความเกี่ยวข้องกับการออกแบบทางด้านกราฟิกส์ ซึ่งผู้พัฒนาอาจไม่คุ้นเคย

ในบางกรณีการสร้างภาพเคลื่อนไหวอย่างอัตโนมัติทำให้เกิดข้อจำกัดมากเกินไป และระบบควรมีความสามารถปรับเปลี่ยนได้ ดังนั้นจึงมีการแบ่งประเภทของผู้ใช้ระบบออกเป็น 3 กลุ่มดังนี้

1. *ผู้ใช้ปลายทาง* เป็นผู้ที่ใช้ระบบในการทดลอง เพื่อทำความเข้าใจการทำงานของอัลกอริทึม ดังนั้นผู้ใช้ปลายทางจึงควรสามารถเรียกใช้โปรแกรมในลักษณะเชิงโต้ตอบได้ นั่นคือสามารถให้แสดงการจินตทัศน์เร็วหรือช้า แสดงภาพย้อนหลัง หยุดชั่วคราว หรือเปลี่ยนแปลงวัตถุที่จะพิจารณาได้
2. *นักเขียนโปรแกรมประยุกต์* จะสร้างภาพเคลื่อนไหวอย่างง่าย ๆ โดยทั่วไปนักเขียนโปรแกรมประยุกต์ต้องการใช้ภาพเคลื่อนไหวสำหรับ การหาจุดบกพร่อง ช่วยในการหาแนวคิดในการวิจัย และเป็นต้นแบบ (prototype)
3. *นักเขียนโปรแกรมชั้นสูง* ควรจะสามารถแก้ไข ปรับปรุงส่วนของการจินตทัศน์ได้ เนื่องจากการสร้างภาพเคลื่อนไหวแบบอัตโนมัติของระบบอาจไม่ตรงตามความต้องการของนักเขียนโปรแกรม ดังนั้นจึงต้องจัดเตรียมวิธีในการแก้ไขลักษณะการสร้างภาพเคลื่อนไหวโดยไม่ต้องเขียนโปรแกรมเพิ่ม

---

[20] Tal and Dobbin, "Visualization of geometric algorithm," IEEE Transactions on Visualiztion and Computer Graphics 1(1995) : 194-204.

ในระบบได้กำหนดการต่อประสานสำหรับการจินตทัศน์อัลกอริทึมคล้าย ๆ กัน ซึ่งประกอบด้วยการเรียกโปรแกรมในคลัง สำหรับนักเขียนโปรแกรมประยุกต์ และแฟ้มลักษณะ (style file) สำหรับนักเขียนโปรแกรมขั้นสูง ซึ่งนักเขียนโปรแกรมประยุกต์จะเขียนรหัสภาษาซีสั้น ๆ ในการกำหนดโครงสร้างของภาพเคลื่อนไหว ซึ่งระบบการจินตทัศน์สามารถสร้างภาพเคลื่อนไหวที่เหมาะสมจากรหัสภาษาซีนี้ได้ สำหรับนักเขียนโปรแกรมระดับสูงสามารถแก้ไขแฟ้มลักษณะ เพื่อเปลี่ยนแปลงการแสดงภาพเคลื่อนไหว ซึ่งภาพเคลื่อนไหวนี้ยังคงสร้างโดยระบบจินตทัศน์ ดังนั้นในการจินตทัศน์ใด ๆ จะประกอบด้วย 4 ส่วน

1. ระบบจินตทัศน์อัลกอริทึม
2. โปรแกรมอัลกอริทึม
3. การเกี่ยว (hook) การจินตทัศน์ภายในโปรแกรมอัลกอริทึม เป็นงานหลักของผู้สร้างภาพเคลื่อนไหว (animator)
4. แฟ้มลักษณะ

2.3.2 ส่วนต่อประสาน

เนื่องจากผู้ใช้มีผลต่อแนวคิดในการออกแบบระบบดังกล่าว ในส่วนนี้จึงกล่าวถึงส่วนต่อประสานกับผู้ใช้ซึ่ง GASP จัดเตรียมสำหรับผู้ใช้ประเภทต่าง ๆ

2.3.2.1 ส่วนต่อประสานกับนักเขียนโปรแกรมประยุกต์

ใน GASP นักเขียนโปรแกรมเพียงแต่เขียนรหัสภาษาซีสั้น ๆ โดยไม่จำเป็นต้องมีความรู้ด้านคอมพิวเตอร์กราฟิกส์ ผู้ใช้เพียงแต่กำหนดสิ่งที่ต้องการให้เคลื่อนไหวเท่านั้น ภาพการเคลื่อนไหวจะสร้างจากวัตถุทางเรขาคณิตและการแสดงโครงสร้างข้อมูล ตัวอย่างของวัตถุเรขาคณิต เช่น เส้น จุด รูปหลายเหลี่ยม ทรงกระบอก ทรงกลม ตัวอย่างของโครงสร้างข้อมูล เช่น ต้นไม้ และรายการซึ่งมีหลายรูปแบบ การดำเนินการที่ใช้กับวัตถุขึ้นอยู่กับชนิดของวัตถุนั้น จะมีการจินตทัศน์มาตรฐานในกลุ่มด้านเรขาคณิตเชิงคำนวณซึ่งสร้างมาจากคลังนี้ พารามิเตอร์จะเป็นชนิดข้อมูลพื้นฐาน เช่น จำนวนเต็ม จำนวนจริง อักขระ แถวลำดับของจำนวนเต็ม หรือสายอักขระ ในคลังของ GASP ประกอบไปด้วย การดำเนินการ 4 กลุ่ม ดังนี้

1. *การดำเนินการกับวัตถุ* วัตถุของ GASP ประกอบด้วย วัตถุเรขาคณิตในสองมิติ วัตถุเรขาคณิตในสามมิติ วัตถุเชิงการจัด (combinatorial object) มุมมอง (view) ข้อความและหัวเรื่อง วัตถุสามารถถูกสร้าง ลบ และแก้ไข รวมทั้ง คัดลอก จัดกลุ่ม และแยกกลุ่มได้
2. *หน่วยอะตอม (Atomic Units)* ในแต่ละช่วงของอัลกอริทึม จะเกี่ยวข้องกับหลายการดำเนินการ ซึ่งควรจะทำการจินตทัศน์พร้อม ๆ กัน ผู้ใช้สามารถแยกแต่ละช่วงของอัลกอริทึมโดยการรวมกลุ่มการดำเนินการพื้นฐานเป็นหน่วยอะตอมโดยใช้ Begin_atomic_End_atomic ในการรวมการดำเนินการ GASP จะประมวลผลการจินตทัศน์เสมือนเป็นหน่วยเดียวกัน ตัวอย่างเช่น ต้องการสร้างรูปทรงหลายหน้า สร้างระนาบใหม่ และหมุนวัตถุชิ้นที่สาม ทั้งสามการดำเนินการนี้ถือเป็นหนึ่งหน่วย ซึ่งจะทำการจินตทัศน์พร้อม ๆ กัน ตัวอย่างของรหัสของการกระทำนี้แสดงดังรูปที่ 2-12 นอกจากนี้ หน่วยอะตอม

สามารถสร้างแบบมีการซ้อนใน (nest) และสามารถรวมข้อความ และเสียงอธิบายเหตุการณ์ที่เกิดขึ้นภายในหน่วยนี้ได้

```
Begin_atomic("Example");
    Add_faces("Poly,"face_no,faces);
    Create_plane("Plane,"point1,point2,point3,point4);
    Rotate_object("ThirdObj");
End_atomic();
```

รูปที่ 2-12 ตัวอย่างการสร้างหน่วยอะตอม[21]

3. *การเคลื่อนที่ (Motion)* สามารถใช้ได้กับวัตถุเดี่ยว ๆ หรือกลุ่มของวัตถุที่ได้รับการจัดกลุ่มแล้ว หรือกับกล้องถ่ายรูป (camera) เมื่อมีการเคลื่อนที่กล้องถ่ายรูป ภาพที่ปรากฏจะเปลี่ยนตามไปด้วย ใน GASP มีการเคลื่อนที่ 5 แบบ คือ Rotate_obj, Scale_world หรือ Scale_obj , Translate_world หรือ Translate_obj, LinearPath_world หรือ LinearPath_obj ,Path_world หรือ Path_obj การเคลื่อนที่ทุกแบบจะมีการเปลี่ยนแปลงอย่างช้า ๆ ตัวอย่างเช่น การหมุนจะทำอย่างต่อเนื่องช้า ๆ จนกระทั่งได้มุมตามต้องการ
4. *ยกเลิก (undo)* ใช้สำหรับการแสดงย้อนหลัง การดำเนินการนี้จะใช้พารามิเตอร์เป็นจำนวนของหน่วยอะตอม ที่ต้องการย้อนกลับ

2.3.2.2 ส่วนต่อประสานกับนักเขียนโปรแกรมขั้นสูง

การดำเนินการที่ GASP สนับสนุนจะสร้างภาพเคลื่อนไหวซึ่งสาธิตการดำเนินการนั้นในรูปแบบที่เหมาะสม หากนักเขียนโปรแกรมต้องการปรับเปลี่ยนพารามิเตอร์สำหรับสร้างภาพเคลื่อนไหวที่ต่างออกไป สามารถทำได้โดยการแก้ไขแฟ้มลักษณะ ซึ่งภาพเคลื่อนไหวยังคงถูกสร้างโดยระบบโดยมีผลต่อการจินตทัศน์ไม่ใช่การเขียนโปรแกรม มีพารามิเตอร์ซึ่งสามารถเปลี่ยนแปลงได้ในแฟ้มลักษณะดังนี้

1. *การจินตทัศน์พื้นฐาน (Visualizing primitives)* การจินตทัศน์พื้นฐานสามารถทำได้หลายแบบ ซึ่งมีค่าเริ่มต้นที่ใช้โดย GASP อย่างไรก็ตามสามารถปรับเปลี่ยนเป็นแบบอื่นได้ ตัวอย่างเช่น สามารถสร้างวัตถุได้ โดยการปรับขนาดให้ใหญ่ขึ้น เคลื่อนที่เข้าสู่ภาพที่แสดงอยู่ กระพริบวัตถุ
2. *การจินตทัศน์วัตถุ (Visualizing object)* สามารถแสดงวัตถุได้หลายแบบ เช่น แบบราบ แบบโครงลวด (wire-frame) แสดงเส้นขอบหรือจะไม่แสดงก็ได้ สีที่แสดงวัตถุจะมีการเลือกสำหรับวัตถุและคุณสมบัติที่สร้าง GASP จะเลือกสีที่เหมาะสมสำหรับอุปกรณ์ที่แสดงผล เช่นบนหน้าจอ หรือวีดิทัศน์
3. *การจินตทัศน์การเคลื่อนที่ (Visualizing motion)* พารามิเตอร์สำหรับการเคลื่อนที่ในแฟ้มลักษณะเช่น ค่าแกนของการหมุนและมุม ขนาดของการปรับอัตราส่วน
4. *พารามิเตอร์อื่น ๆ* เช่นการกำหนดว่าแสดงการจินตทัศน์บนหน้าจอ หรือวีดิทัศน์ เนื่องจากการกำหนดสีบนอุปกรณ์ทั้งสองมีความแตกต่างกัน หรืออาจเพิ่มบรรทัดซึ่งกำหนดให้หยุดการแสดงผลแต่ละกรอบ

---

[21] Tal and Dobbin, "Visualization of geometric algorithm," IEEE Transactions on Visualiztion and Computer Graphics 1(1995) : 194-204.

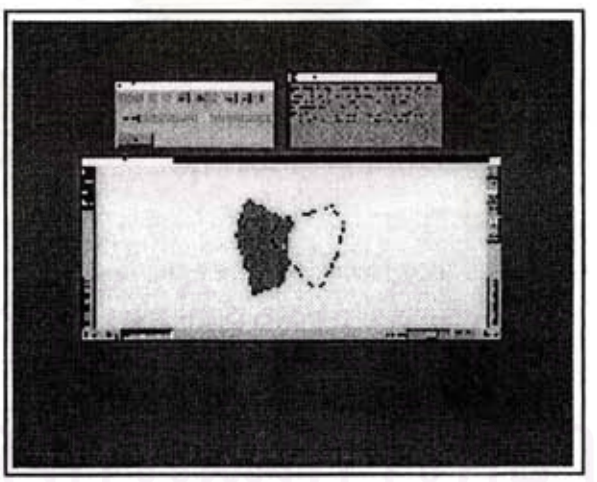
ตัวอย่างของแฟ้มลักษณะ ซึ่งกำหนดสีพื้นหลังเป็นเทา และเลือกการแสดงสีสำหรับวีดิทัศน์ แต่ละหน่วยอะตอมแบ่งออกเป็น 30 กรอบ การปรับเลี่ยนอัตราส่วนใช้ค่า 0.82 จากขนาดดั้งเดิม หมุน 20 องศารอบแกน y แฟ้มลักษณะซึ่งมีพารามิเตอร์ดังกล่าว แสดงดังรูปที่ 2-13

```
begin_global_style
   background = light_gray;
   color = VIDEO;
   frames = 30;
   scale_world = 0.82 0.82 0.82;
   rotation_world = Y 20.0;
end_global_style
```

รูปที่ 2-13 ตัวอย่างแฟ้มลักษณะ [22]

2.3.2.3 ส่วนต่อประสานกับผู้ใช้

ส่วนนี้เป็นสภาวะแวดล้อมเชิงโต้ตอบ ซึ่งประกอบด้วย แผงควบคุม (control panel) หน้าต่างข้อความ และหน้าต่างอัลกอริทึม ในแผงควบคุม ผู้ใช้ สามารถควบคุมการประมวลผลของอัลกอริทึม เช่น หยุด ทำงานทีละขั้น ย้อนกลับ หยุดชั่วคราวได้ หน้าต่างอัลกอริทึม ผู้ใช้สามารถหมุน ปรับขนาดของภาพได้ เลื่อนภาพเข้าออกได้ สามารถสั่งพิมพ์รายละเอียดของวัตถุที่เลือกได้ หน้าต่างข้อความ เป็นข้อความแสดงเกี่ยวกับ อัลกอริทึมและการจินตทัศน์ในหน้าต่าง เพื่ออธิบายเหตุการณ์ที่เกิดขึ้น ตัวอย่างหน้าจอแสดงดังรูปที่ 2-14



รูปที่ 2-14 รูปแสดงสภาวะแวดล้อมของ GASP[23]

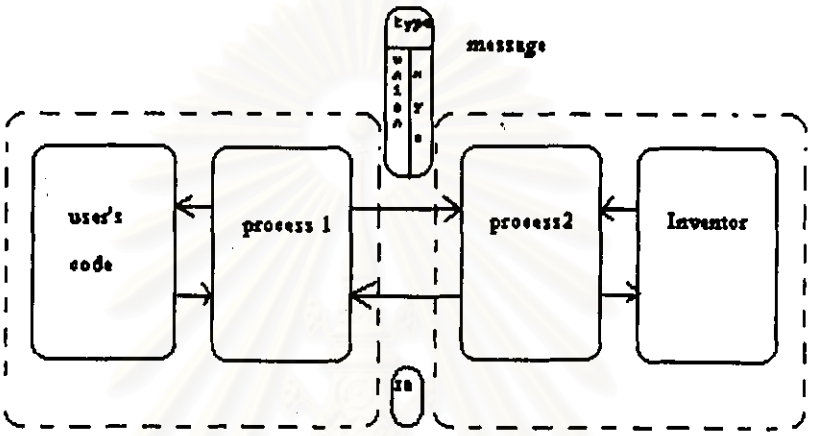
---

[22] Ibid.

[23] Ibid.

2.3.3 โครงสร้างของ GASP

GASP เขียนด้วยภาษาซี ภายใต้ระบบปฏิบัติการ Unix ประกอบด้วย 2 กระบวนการ ซึ่งติดต่อกันโดยใช้ข้อความคำสั่ง (message) โดยกระบวนการ 1 จะประกอบด้วยชุดของกระบวนงานซึ่งประกอบขึ้นเป็นส่วนต่อประสานกับนักเขียนโปรแกรม กระบวนการ 2 จะรับผิดชอบเกี่ยวกับการประมวลผลการจินตทัศน์และจัดการข้อมูลเข้าของผู้ใช้ดังรูปที่ 2-15



รูปที่ 2-15 โครงสร้างของ GASP[24]

เมื่อโปรแกรมประยุกต์เริ่มต้นการเรียกกระบวนงานในกระบวนการ 1 และกระบวนการ 1 จะเตรียมข้อความคำสั่งซึ่งมีชนิดของการดำเนินการ และข้อมูลที่เกี่ยวข้องสำหรับการดำเนินการนั้น แล้วส่งไปยังกระบวนการ 2 กระบวนการ 2 จะปรับปรุงโครงสร้างข้อมูลภายใน หรือประมวลผลการจินตทัศน์ขึ้นอยู่กับข้อความคำสั่งที่ได้รับ แล้วส่งการตอบรับกลับไปที่กระบวนการ 1 การตอบรับนี้จะส่ง รหัสประจำตัว (ID) ของวัตถุ กระบวนการ 1 จะปรับปรุงตารางแฮชของวัตถุ แล้วคืนกลับไปที่รหัสของโปรแกรมประยุกต์ กระบวนการ 2 ซึ่งรับผิดชอบด้านกราฟิกส์ ทำงานในวงวนหลักเหตุการณ์ (event main loop) ในที่นี้ใช้ Inventor'sTimer-Sensor ในการปรับปรุง กราฟิกส์ กระบวนการ 2 จะมีการตรวจสอบเป็นช่วงเวลาว่าการจินตทัศน์กำลังดำเนินไปอย่างไร ถ้าหากมีการประมวลผลไปข้างหน้าจะตรวจสอบว่าต้องทำการปรับปรุงกราฟิกส์หรือไม่ หรืออยู่ ณ จุดที่ต้องรับคำสั่งใหม่จากกระบวนงาน 1 ในกรณีนี้จะตรวจสอบว่ามีข้อความคำสั่งจากกระบวนงาน 1 หรือไม่ แล้วทำการเก็บข้อความคำสั่งที่รับมา ปรับปรุงโครงสร้างภายใน และยืนยันการตอบรับข้อความคำสั่งจนกระทั่งถึง ข้อความคำสั่ง END_ATOMIC หลังจากนั้น กระบวนงาน 2 จะเริ่มต้นประมวลผลคำสั่งทั้งหมดในกลุ่มหน่วยอะตอม และบอกกระบวนงาน 1 เกี่ยวกับการจบการทำงาน ถ้าหากการจินตทัศน์ทำงานย้อนหลัง จะทำการปรับปรุงการจินตทัศน์ตามขั้นตอนที่ทำอยู่

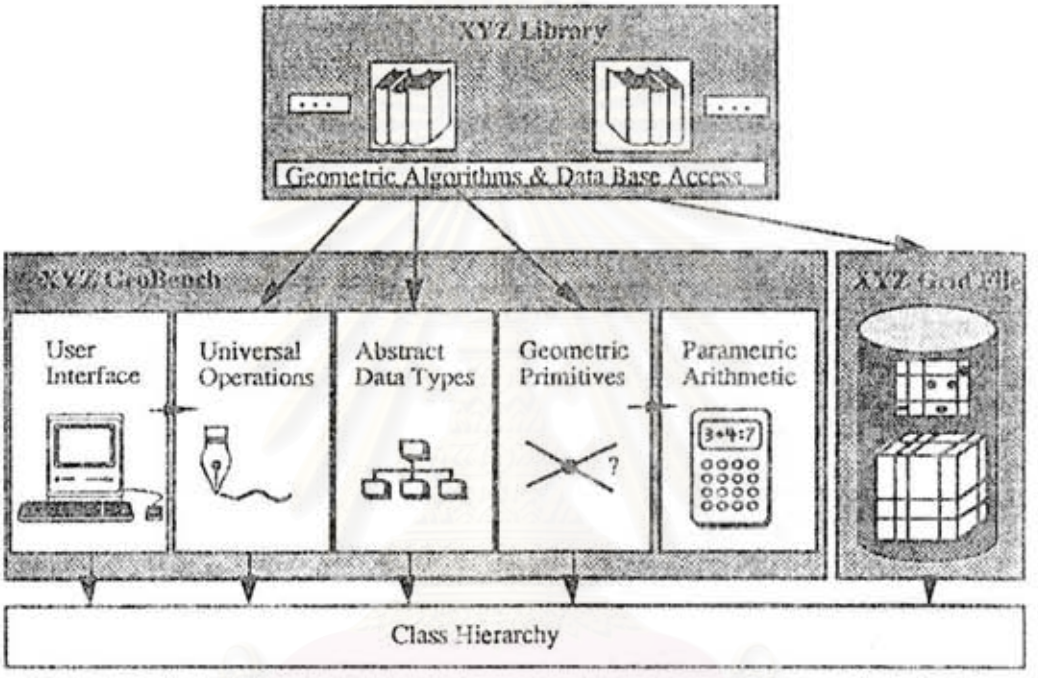
---

[24] Ibid.

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.4 XYZ GeoBench (eXperimental geometrY Zurich)[25]

XYZ Geobench เป็นสภาวะแวดล้อมสำหรับการเขียนโปรแกรม (programming environment) และคลังโปรแกรม สำหรับการคำนวณทางด้านเรขาคณิต เขียนโดยออบเจกต์ปาสคาล (Object Pascal) ทำงานบนเครื่องแมคอินทอช โดยในคลังโปรแกรมของระบบบรรจุอัลกอริทึมสำหรับปัญหาในสองมิติ บางปัญหาในสามมิติ และหนึ่งอัลกอริทึมใน d มิติ



รูปที่ 2-16 โครงสร้างของ XYZ Geobench และความสัมพันธ์กับโปรแกรมสำเร็จรูปในระบบ[26]

การพัฒนาระบบมีจุดประสงค์เพื่อพัฒนาซอฟแวร์ที่มีประโยชน์สำหรับการคำนวณทางด้านเรขาคณิต และเพื่อทดสอบในการประยุกต์ใช้งานที่หลากหลาย ในระบบมุ่งเน้นไปที่การพัฒนาโปรแกรมที่มีความเข้มแข็ง (robust) สามารถจัดการข้อมูลเข้าที่เป็นกรณีพิเศษได้ ส่วนประกอบของ XYZ Geobench จะรวมอยู่ด้วยกันโดยใช้ลำดับชั้นของกลุ่ม (class hierarchy) ของวัตถุเรขาคณิต สำหรับการอำนวยความสะดวกในการสร้างการจินตทัศน์ในระบบนี้ ผู้พัฒนาการจินตทัศน์จะต้องเขียนโปรแกรมสำหรับการวาดการเปลี่ยนแปลงลงไปโดยตรงสำหรับเหตุการณ์แต่ละเหตุการณ์

2.4.1 โครงสร้างและส่วนประกอบของ XYZ GeoBench

ในส่วนนี้จะกล่าวถึงส่วนประกอบของ Geobench จากรูปที่ 2-16 แสดงความสัมพันธ์ของส่วนประกอบใน GeoBench และความสัมพันธ์กับโปรแกรมสำเร็จรูปอื่นในระบบ

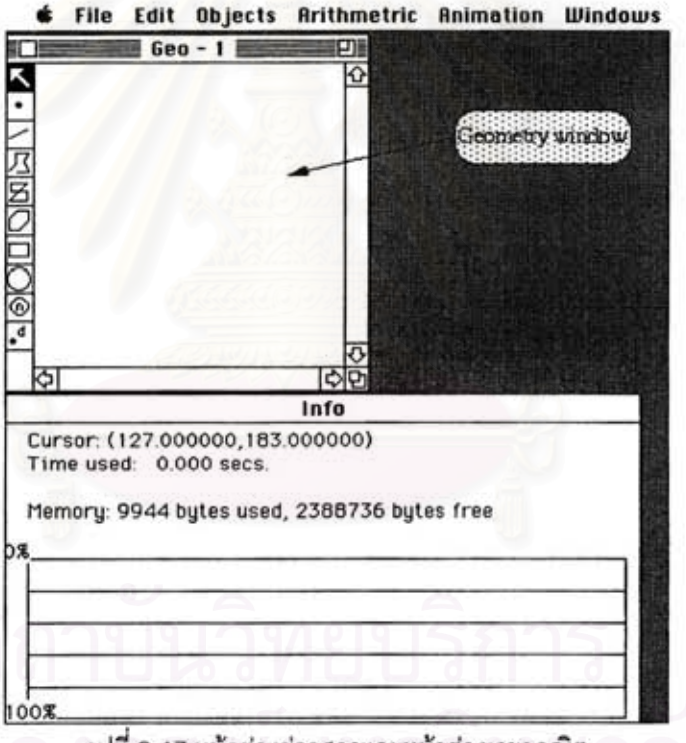
---

[25] Schorn, "The XYZ GeoBench : A programming environment for geometric algorithms," LNCS 553,(1991):187-202.

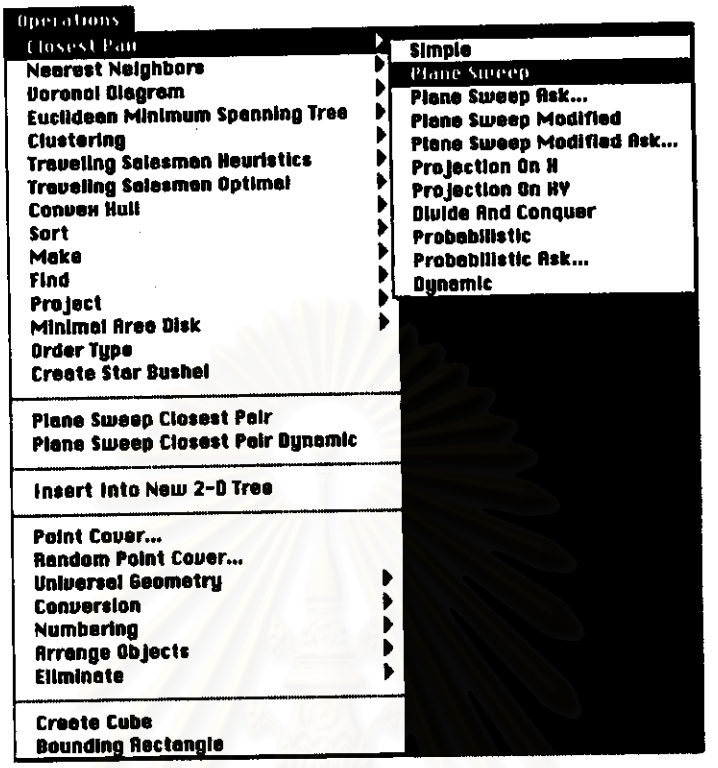
[26] Ibid.

2.4.1.1 ส่วนต่อประสานกับผู้ใช้และการจินตทัศน์อัลกอริทึม

GeoBench ใช้รูปแบบการต่อประสานของแมคอินทอช รูปที่ 2-17 ผู้ใช้จะเห็นหน้าต่างข่าวสาร (info window) ซึ่งมีข้อมูล เช่น หน่วยความจำที่เหลือ พิกัดของตัวชี้ตำแหน่ง (cursor) ชนิดของวัตถุที่ถูกเลือกในปัจจุบัน การเลือกวัตถุสำหรับรับข้อมูลเข้าสามารถทำโดยการเลือกแผ่น (palette) ที่อยู่ในแต่ละหน้าต่างเรขาคณิต (geometry window) หรือโดยการเลือกเมนูวัตถุ (objects) การคำนวณจะเกิดขึ้นในหน้าต่างเรขาคณิต โดยเลือกการดำเนินการจากเมนูดำเนินการ (operation) ดังในรูปที่ 2-18 เมนูดำเนินการจะแสดงเฉพาะการดำเนินการที่ทำได้สำหรับวัตถุที่เลือก การกระทำตามการดำเนินการที่เลือกจะสร้างหน้าต่างเรขาคณิตใหม่ ซึ่งมีผลลัพธ์จากการดำเนินงาน ในเมนูการจินตทัศน์ (animation) ผู้ใช้จะเลือกอัลกอริทึมที่ต้องการจินตทัศน์ และเมนูตัวเลข (arithmetic) จะเกี่ยวกับชนิดของตัวเลขสำหรับวัตถุที่สร้างใหม่



รูปที่ 2-17 หน้าต่างข่าวสารและหน้าต่างเรขาคณิต



รูปที่ 2-18 เมนูการดำเนินการของ XYZ Geobench

สำหรับการจินตทัศน์อัลกอริทึม สามารถที่จะเลือกได้ว่าจะรวมรหัสของโปรแกรมในส่วนของการสร้างภาพเคลื่อนไหวในการแปลโปรแกรมหรือไม่ รหัสนี้จะตรวจสอบว่ามีการเลือกใช้การจินตทัศน์สำหรับอัลกอริทึมหรือไม่ ถ้าใช่จะทำการปรับปรุงสถานะปัจจุบันที่มองเห็นได้ และรอผู้ใช้อนุญาตให้ดำเนินการต่อ รหัสสำหรับการสร้างภาพเคลื่อนไหวมีโครงสร้างดังรูปที่ 2-19

```
{ Geometric algorithm changing internal state. }
{ $IFC myAlgAnim }
   if animationFlag[myAlgAnim] then
   { Update graphical state information , usually draw some objects. }
     waitForClick(animationFlag[myAlgAnim]
   { Update graphical state information, usually erase som objects. }
   end;
{ $ENDC }
```

รูปที่ 2-19 ตัวอย่างสำหรับการสร้างภาพเคลื่อนไหว[27]

กระบวนงาน "waitForClick" เป็นส่วนต่อประสานระหว่างผู้ใช้และอัลกอริทึมที่กำลังทำการจินตทัศน์อยู่ ซึ่งสนับสนุนภาวะการทำงานทีละขั้น (single step mode) และภาวะภาพยนตร์ (movie mode) ผู้ใช้สามารถกำหนดอัตราเร็วได้

[27] Ibid.

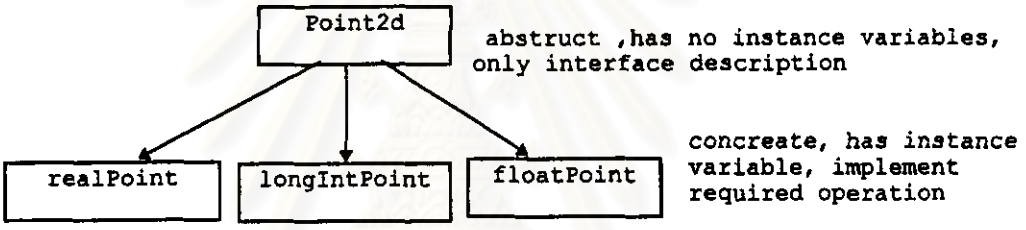
2.4.1.2 การคำนวณพื้นฐานด้านเรขาคณิต (Geometric primitives)

การคำนวณพื้นฐานในระบบส่วนใหญ่เป็นการคำนวณสำหรับชนิดข้อมูล point2d ซึ่งสนับสนุนชนิดตัวเลข 3 แบบ คือ จำนวนเต็ม จำนวนจริง และเลขอิงดรรชนี (floating point) โดยส่วนใหญ่ของโปรแกรมในคลังจะใช้การคำนวณพื้นฐานทั้งหมดประมาณ 15 แบบ ดังตัวอย่างต่อไปนี้

```
function whichSide(p,q,r:point2d):(-1, 0,+1);/*หาว่าจุด q อยู่ด้านไหนของเส้นตรง pr */
function cressProduct(p,q,r,s:point2d):real; /*คำนวณหาผลคูณไขว้เชิงเวกเตอร์ (p-q)x(s-r) */
function distance (p, q : point2d) : real; /* หาระยะยูกลิดระหว่าง p และ q */
```

2.4.1.3 การสับเปลี่ยนชนิดตัวเลขและการทำเลขอิงดรรชนีให้เป็นพารามิเตอร์

การเลือกชนิดของตัวเลขอาจจะมีนัยสำคัญกับการพัฒนาอัลกอริทึมในบางกรณี การทดลองใช้ชนิดตัวเลขที่ต่าง ๆ กันจึงมีความจำเป็นที่ต้องมีความสามารถในการสับเปลี่ยนชนิดของตัวเลขสำหรับการพัฒนาอัลกอริทึม ในระบบจึงกำหนดกลุ่มข้อมูลนามธรรม (abstract class) ดังรูปที่ 2-20



รูปที่ 2-20 กลุ่มข้อมูลนามธรรม point2d[28]

การกำหนดข้อมูลนี้ มีการคำนวณด้านเรขาคณิตพื้นฐานตามชนิดของตัวเลข อัลกอริทึมเพียงแต่ใช้ฟังก์ชัน หรือกระบวนงานที่ได้ระบุไว้ในข้อมูลนามธรรม โดยสามารถเรียกใช้ได้กับข้อมูลทั้งสามชนิด

2.4.1.4 แบบชนิดข้อมูลนามธรรม (Abstract data types)

ในส่วนนี้อธิบายตัวอย่างของชนิดของข้อมูลแบบนามธรรมในระบบบางชนิด คือ

1. *ลำดับ (Sequence)* กลุ่มนามธรรม "sequence" คือกลุ่มของวัตถุใด ๆ ซึ่งพัฒนาโดยใช้โครงสร้างรายการโยง (linked list) และ แถวลำดับ(แบบพลวัต) การพัฒนาโดยใช้รายการโยงจะมีประโยชน์ในกรณีที่ไม่ทราบจำนวนสมาชิกล่วงหน้าและใช้การประมวลผลแบบลำดับ
2. *พจนานุกรม (Dictionary)* ใช้แนวคิดการอ้างอิงเป็นหลัก มีการดำเนินการในลักษณะการค้นหา และการดำเนินการสำหรับทำการเปลี่ยนแปลงโครงสร้างข้อมูล การพัฒนาพจนานุกรม ใช้ต้นไม้ AVL หรือรายการเรียงลำดับแล้ว หรือ เวกเตอร์แบบเรียงลำดับ
3. *แถวคอยแบบมีบุริมภาพ (Priority queue)* ชนิดข้อมูลแบบนามธรรม แถวคอยแบบมีบุริมภาพได้รับการพัฒนาโดยใช้ ฮีปและพจนานุกรม

---

[28] Ibid.

2.4.1.5 **การดำเนินการทั่วไป (Universal operations)**

เป็นการดำเนินการซึ่งทุก ๆ วัตถุภายในระบบสามารถเข้าใจ ซึ่งการดำเนินการนี้อยู่ที่รากของลำดับชั้นของวัตถุ ทุกวัตถุในระบบเป็นระดับล่างของวัตถุรากสามารถใช้การดำเนินการเหล่านี้ร่วมกัน การดำเนินการทั่วไปที่ระบบจัดเตรียมมีดังนี้

1. *การจัดการหน่วยความจำ (Memory management)* ซึ่งมีวิธีในการสร้าง การทำลาย และการทำซ้ำของวัตถุ
2. *การรับข้อมูลเข้าและส่งข้อมูลออกเชิงโต้ตอบ (Interactive input/output)* ในระบบสนับสนุน การแสดงวัตถุด้านเรขาคณิตบนหน้าจอ การแสดงวัตถุในลักษณะสว่าง การกระพริบวัตถุ ให้ผู้ใช้สามารถป้อนวัตถุได้ในแบบโต้ตอบ โดยทั่วไปทำได้โดยการลากด้วยเมาส์
3. *การรับข้อมูลเข้าและส่งข้อมูลออกโดยใช้แฟ้ม (File input/output)* เมื่อประมวลผลวัตถุทางเรขาคณิตอาจต้องการจัดเก็บอย่างถาวรในหน่วยเก็บข้อมูลสำรอง ในที่นี้ใช้รูปแบบแฟ้มแบบกระแสไบต์ (byte stream)
4. *การแปลงทางเรขาคณิต (Geometric transformations)* ประกอบด้วยการแปลงดังต่อไปนี้ คือ การหมุน การกำหนดอัตราส่วน การสร้างภาพสะท้อน
5. *การคำนวณเกี่ยวกับชนิด (Type computations)* เป็นการคำนวณเกี่ยวกับการทดสอบชนิดของวัตถุ

2.5 AVis[29]

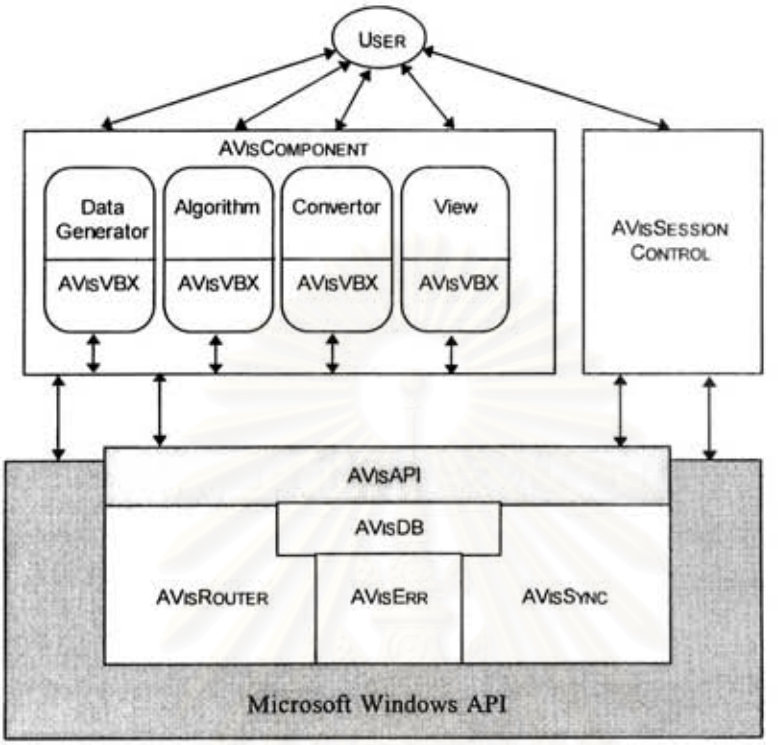
AVis เป็นระบบการจินตทัศน์อัลกอริทึมที่ทำงานบนระบบปฏิบัติการไมโครซอฟต์วินโดวส์รุ่น 3.1 มีจุดมุ่งหมายหลักในการออกแบบเพื่อให้มีการแบ่งการจินตทัศน์อัลกอริทึมออกเป็นส่วนประกอบย่อย ทำให้สามารถแยกองค์ประกอบแต่ละส่วนไปพัฒนาต่างหากได้ โดยส่วนต่าง ๆ ของระบบเป็นอิสระจากกัน ผู้พัฒนาจะคำนึงถึงเฉพาะส่วนที่ต้องการพัฒนาเท่านั้น นอกจากนั้นงานที่พัฒนาควรถูกเก็บอยู่ในคลังคำสั่งและสามารถนำไปใช้ใหม่ได้ การปรับปรุงประสิทธิภาพของคลังคำสั่งควรกระทำได้โดยไม่มีผลกระทบต่อส่วนประกอบอื่น ๆ ในระบบ

2.5.1 โครงสร้างของ AVis

โครงสร้างและความสัมพันธ์ระหว่างส่วนประกอบต่าง ๆ ของระบบ รวมทั้งผู้ใช้งาน แสดงดังรูปที่ 2-21 AVis จะแบ่งการจินตทัศน์อัลกอริทึมออกเป็นส่วนประกอบย่อย ๆ สี่ประเภท คือ ส่วนสร้างข้อมูล (data generator) ส่วนอัลกอริทึม (algorithm) ส่วนแปลงคำสั่ง (converter) และส่วนแสดงผล (view) ผู้พัฒนาองค์ประกอบการจินตทัศน์จะพัฒนาแต่ละส่วนแยกจากกัน โดยมีการกำหนดข้อความคำสั่งในการติดต่อประสานงานระหว่างส่วนประกอบแต่ละส่วน หลังจากนั้นในการนำไปใช้งานจะนำส่วนประกอบแต่ละส่วนมารวมกันเพื่อสร้างเป็นการจินตทัศน์อัลกอริทึม สำหรับรายละเอียดของ AVis จะกล่าวถึงในบทที่ 4

---

[29] ชัชวาล วงศ์ศิริประเสริฐ, "การออกแบบและพัฒนาแก่นกลางของระบบจินตทัศน์อัลกอริทึม," (วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, 2539) .



รูปที่ 2-21 โครงสร้างและความสัมพันธ์ของส่วนประกอบต่าง ๆ ใน AVis[30]

2.6 *สรุป*

ระบบจินตทัศน์อัลกอริทึมส่วนใหญ่มีจุดมุ่งหมายเพื่ออำนวยความสะดวกแก่ผู้พัฒนาการจินตทัศน์และผู้ใช้ปลายทาง โดยในส่วนของผู้พัฒนาการจินตทัศน์ จะมุ่งเน้นในเรื่องของความอิสระในการพัฒนาส่วนของอัลกอริทึมและส่วนของการแสดงผล รวมทั้งการนำส่วนที่พัฒนาแล้วมาใช้ร่วมกัน ส่วนของผู้ใช้ปลายทาง ระบบส่วนใหญ่มีการจัดเตรียมหน้าที่ในส่วนที่ทำให้ผู้ใช้สามารถปรับความเร็วในการแสดงผล เลือกรูปแบบการแสดงผล และสามารถให้ทำงานทีละขั้นหรือกำหนดจุดหยุด ซึ่งจะคล้าย ๆ กันทุกระบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

---

[30] ชัชวาล วงศ์ศิริประเสริฐ, "การออกแบบและพัฒนาแก่นกลางของระบบจินตทัศน์อัลกอริทึม," (วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, 2539) หน้า 36.