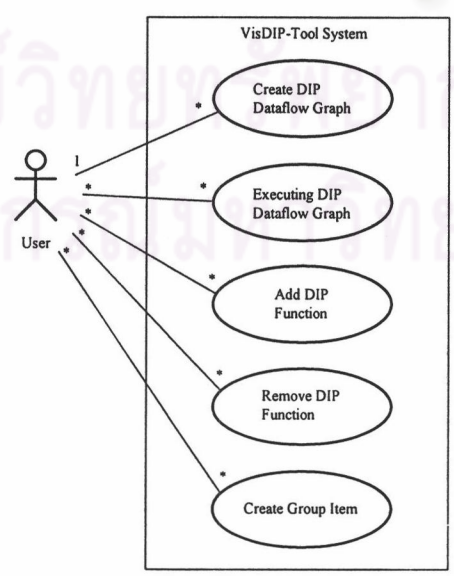
บทที่ 4
การวิเคราะห์และออกแบบระบบ

เครื่องมือประมวลผลภาพดิจิทัล VisDIP-Tool ในวิทยานิพนธ์นี้ถูกออกแบบและพัฒนาขึ้นเพื่อใช้สำหรับออกแบบและทดสอบขั้นตอนวิธีในการประมวลผลภาพโดยวาดเป็นกราฟกระแสข้อมูลดีไอพี (DIP Dataflow Graph) ซึ่งทำให้เห็นขั้นตอนของการประมวลผลภาพโดยรวม ในบทนี้จะกล่าวถึงการออกแบบระบบบรรณาธิกรของเครื่องมือโดยใช้วิธีการเชิงวัตถุในการวิเคราะห์และออกแบบ ซึ่งใช้แผนภาพยูสเคส (Usecase diagram) แผนภาพคลาส (Class diagram) และแผนภาพซีเควนซ์ (Sequence diagram) เป็นเครื่องมือในการวิเคราะห์และออกแบบโดยมีรายละเอียดดังต่อไปนี้

4.1 การวิเคราะห์การทำงานของเครื่องมือ

4.1.1 แผนภาพยูสเคสรวมของระบบ

แผนภาพยูสเคสเป็นแผนภาพที่ใช้แสดงหน้าที่ต่าง ๆ ของระบบ ซึ่งสามารถแสดงให้ทราบว่าผู้ใช้งานระบบสามารถใช้งานส่วนใดของระบบได้บ้าง แผนภาพยูสเคสในรูปที่ 4.1 แสดงหน้าที่การทำงานของเครื่องมือ VisDIP-Tool สำหรับออกแบบขั้นตอนวิธีในการประมวลผลภาพโดยการสร้างเป็นกราฟกระแสข้อมูลดีไอพี นอกจากนี้ผู้ใช้ยังสามารถทดสอบขั้นตอนวิธีในการประมวลผลภาพโดยสั่งเครื่องมือ VisDIP-Tool ให้ดำเนินการประมวลผลข้อมูลตามกราฟที่ได้ออกแบบไว้ได้



รูปที่ 4.1 ยูสเคสรวมของระบบ

แผนภาพยูสเคสในรูปที่ 4.1 แสดงหน้าที่ทั้งหมดของเครื่องมือ VisDIP-Tool โดยมีรายละเอียดดังนี้

1) Create DIP Dataflow Graph

ระบบบรรณาธิกรจะต้องสามารถสร้างกราฟกระแสข้อมูลดีไอพีโดยการเลือก Item ชนิดต่าง ๆ มาวาดบนหน้าจอแล้วเชื่อมเส้นระหว่าง Port ของ Item ได้

2) Executing DIP Dataflow Graph

เครื่องมือจะต้องสามารถประมวลผลภาพตามกราฟกระแสข้อมูลดีไอพีที่สร้างจากระบบบรรณาธิกรได้

3) Add DIP Function

เครื่องมือจะต้องสามารถเพิ่มฟังก์ชันประมวลผลภาพได้

4) Remove DIP Function

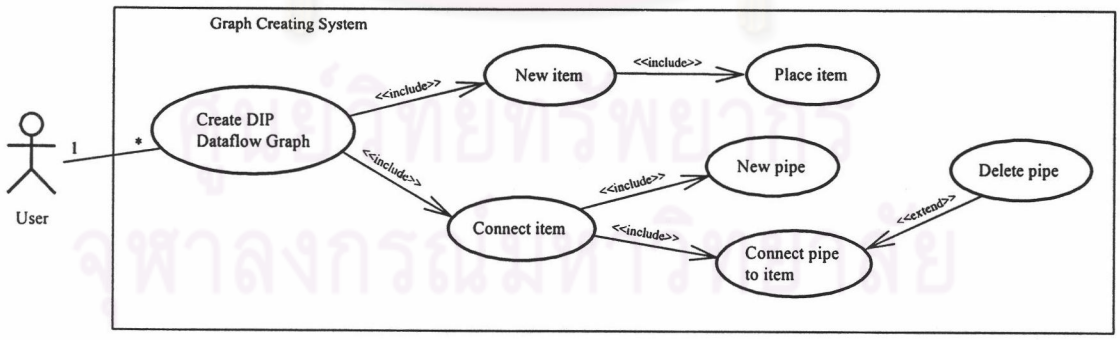
เครื่องมือจะต้องสามารถลบฟังก์ชันประมวลผลภาพได้

5) Create Group Item

เครื่องมือจะต้องสามารถสร้าง Group Item จากกราฟกระแสข้อมูลดีไอพีได้

4.1.2 แผนภาพยูสเคสของการสร้างกราฟกระแสข้อมูลดีไอพี (Create DIP Dataflow Graph)

แผนภาพยูสเคสในรูปที่ 4.2 กำหนดหน้าที่การทำงานของระบบบรรณาธิกรสำหรับสร้างกราฟกระแสข้อมูลดีไอพีเพื่อใช้ออกแบบและทดสอบขั้นตอนวิธีในการประมวลผลภาพ โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.2 ยูสเคสของระบบการสร้างกราฟ

1) New Item

เมื่อผู้ใช้เลือก Item จากเมนูหลักแล้วคลิกเมาส์บนบริเวณบรรณาธิกร ระบบบรรณาธิกรจะต้องสร้าง Item ใหม่ขึ้นมาซึ่งมีสามชนิดได้แก่ Data Item ใช้แทนข้อมูล Process Item ใช้แทนฟังก์ชันประมวลผลภาพ และ Group Item ใช้แทนกราฟกระแสข้อมูลดีไอพีย่อย

2) Place Item

เมื่อ Item ใหม่ถูกสร้างขึ้นมาแล้วจะต้องถูกวางบนบริเวณที่ผู้ใช้คลิกเมาส์

3) Connect Item

ผู้ใช้ต้องสามารถสร้างเส้นเชื่อมต่อ Item ต่าง ๆ เพื่อกำหนดเส้นทางการไหลของข้อมูลได้

4) New Pipe

ในการสร้างเส้นเชื่อม ระบบบรรณาธิกรจะต้องสร้าง Pipe ขึ้นมาใช้สำหรับเชื่อมต่อ Item สองภาพเข้าด้วยกัน

5) Connect pipe to item

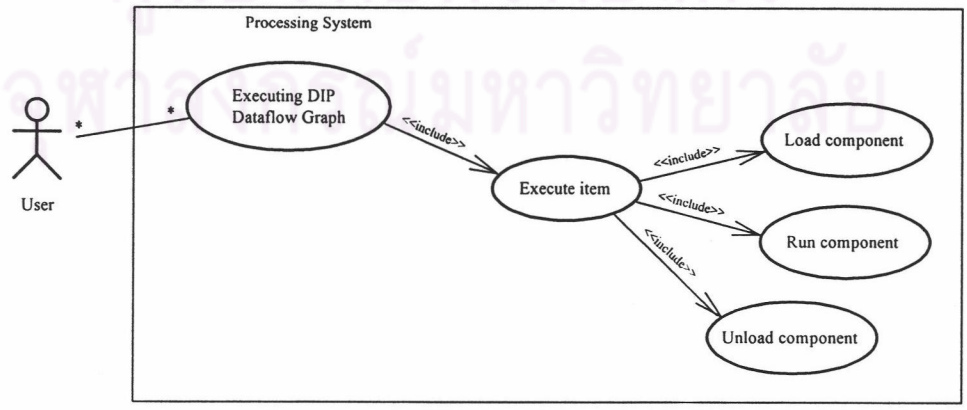
หลังจาก Pipe ถูกสร้างขึ้นมาแล้ว ระบบบรรณาธิกรจะต้องสามารถเชื่อมต่อปลายของเส้น Pipe ทั้งสองข้างเข้ากับ Item สองภาพที่ผู้ใช้กำหนด

6) Delete pipe

ถ้าการเชื่อมต่อ Pipe ผิดปกติ เช่น ปลายของเส้น Pipe ทั้งสองข้างถูกต่อเข้าที่ Item เดียวกัน ระบบบรรณาธิกรจะต้องลบ Pipe โดยอัตโนมัติ

4.1.3 แผนภาพยูสเคสของการประมวลผล (Executing DIP Dataflow Graph)

แผนภาพยูสเคสในรูปที่ 4.3 กำหนดหน้าที่การทำงานของระบบประมวลผลสำหรับทดลองประมวลผลภาพตามกราฟกระแสข้อมูลดีไอพีเพื่อทดสอบขั้นตอนวิธีในการประมวลผลภาพ โดยเริ่มต้นจากการสั่งให้ Data Item ต้นทางทั้งหมดของกราฟให้เริ่มทำงาน จากนั้น Data Item จะส่งข้อมูลต่อไปตาม Pipe ไปยัง Item ตัวต่อไปเพื่อทำการประมวลผลแล้วจึงส่งผ่าน Pipe ต่อไปจนกว่าการประมวลผลจะเสร็จสิ้นหมดทั้งกราฟ ซึ่งมีรายละเอียดดังนี้



รูปที่ 4.3 ยูสเคสของระบบประมวลผล

1) Executing item

เมื่อข้อมูลถูกส่งมาถึง Process Item การประมวลผลภาพจะต้องเกิดขึ้น โดยระบบต้องทำการสั่ง Process Item ให้ทำงาน

2) Load component

เมื่อ Process Item ถูกสั่งให้ทำงาน Process Item ต้องทำการโหลดส่วนประกอบดีไอพีที่มีฟังก์ชันประมวลผลภาพที่ตรงกับ Item สู่หน่วยความจำของเครื่องคอมพิวเตอร์

3) Run component

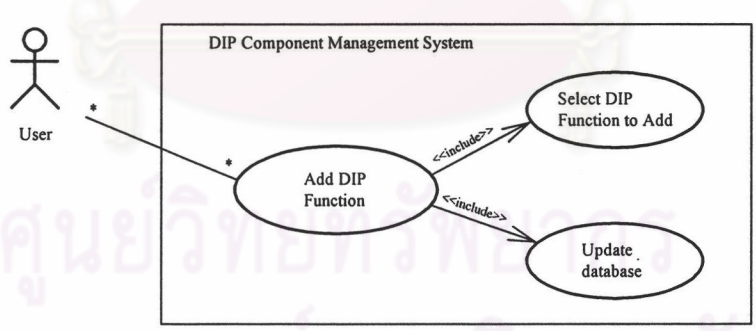
เมื่อโหลดส่วนประกอบดีไอพีสู่หน่วยความจำแล้ว Process Item ต้องส่งข้อมูลให้ฟังก์ชันประมวลผลภาพที่อยู่หน่วยความจำทำการประมวลผลข้อมูล

4) Unload component

ส่วนประกอบดีไอพีต้องถูกลบออกจากหน่วยความจำของเครื่องคอมพิวเตอร์หลังจากฟังก์ชันประมวลผลภาพทำงานเสร็จสิ้น

4.1.4 แผนภาพยูสเคสของการเพิ่มฟังก์ชันประมวลผลภาพ (Add DIP Function)

แผนภาพยูสเคสในรูปที่ 4.4 กำหนดหน้าที่การทำงานของระบบจัดการส่วนประกอบดีไอพีเพื่อเพิ่มฟังก์ชันประมวลผลภาพให้กับเครื่องมือ VisDIP-Tool ซึ่งมีรายละเอียดดังนี้



รูปที่ 4.4 ยูสเคสของการเพิ่มฟังก์ชันประมวลผลภาพ

1) Select DIP function to Add

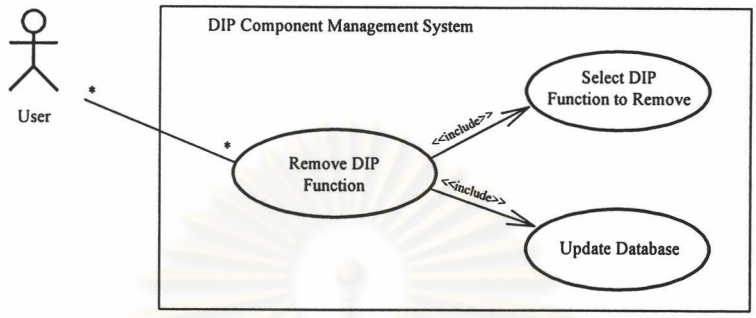
ผู้ใช้ต้องสามารถเลือกฟังก์ชันประมวลผลภาพที่ต้องการจะเพิ่มให้กับเครื่องมือ VisDIP-Tool ได้

2) Update Database

เครื่องมือจะต้องปรับปรุงฐานข้อมูลสำหรับเก็บรายละเอียดฟังก์ชันประมวลผลภาพที่เพิ่มเข้ามา

4.1.5 แผนภาพยูสเคสของการลบฟังก์ชันประมวลผลภาพ (Remove DIP Function)

แผนภาพยูสเคสในรูปที่ 4.5 กำหนดหน้าที่การทำงานของระบบจัดการส่วนประกอบดีไอพี เพื่อลบฟังก์ชันประมวลผลภาพออกจากเครื่องมือ VisDIP-Tool ซึ่งมีรายละเอียดดังนี้



รูปที่ 4.5 ยูสเคสของการลบฟังก์ชันประมวลผลภาพ

1) Select DIP function to Remove

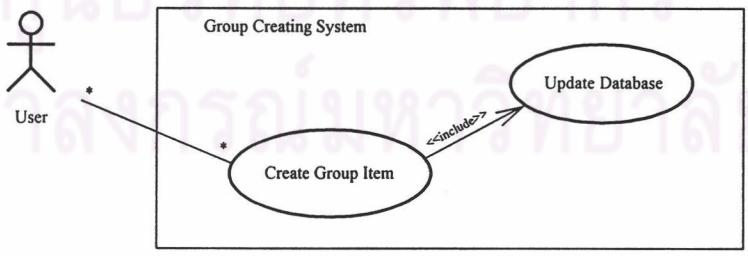
ผู้ใช้ต้องสามารถเลือกฟังก์ชันประมวลผลภาพที่ต้องการจะลบออกจากเครื่องมือ VisDIP-Tool ได้

2) Update Database

เครื่องมือจะต้องปรับปรุงฐานข้อมูลสำหรับเก็บรายละเอียดฟังก์ชันประมวลผลภาพที่ลบออกจากเครื่องมือ VisDIP-Tool

4.1.6 แผนภาพยูสเคสของการสร้าง Group Item (Create Group Item)

แผนภาพยูสเคสในรูปที่ 4.6 กำหนดหน้าที่การทำงานของระบบการสร้าง Group Item เพื่อใช้แทนกราฟกระแสข้อมูลดีไอพีย่อย ซึ่งมีรายละเอียดดังนี้



รูปที่ 4.6 ยูสเคสของการสร้าง Group Item

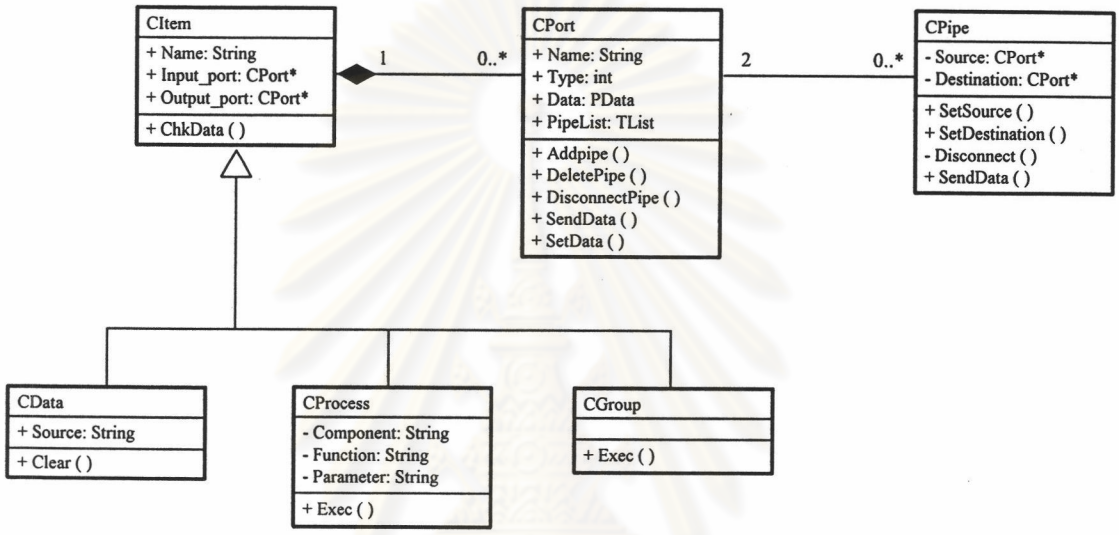
1) Update Database

เครื่องมือจะต้องปรับปรุงฐานข้อมูลสำหรับเก็บรายละเอียดของ Group Item และกราฟกระแสข้อมูลดีไอพีย่อยซึ่งแทนที่ด้วย Group Item

4.2 การออกแบบระบบบรรณาธิกรและระบบการประมวลผล

4.2.1 แผนภาพคลาส

แผนภาพคลาสคือแผนภาพที่ใช้แสดงโครงสร้างความสัมพันธ์ของแต่ละคลาสในระบบ แผนภาพคลาสในหัวข้อนี้เป็นแผนภาพคลาสที่ได้จากการวิเคราะห์และออกแบบระบบ โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.7 แผนภาพคลาสทั้งหมดของระบบ VisDIP-Tool

จากรูปที่ 4.7 แสดงแผนภาพคลาสของระบบของเครื่องมือ VisDIP-Tool ซึ่งได้กำหนดตัวแปรสมาชิกและฟังก์ชันสมาชิกที่สำคัญให้กับคลาสและกำหนดความสัมพันธ์ของแต่ละคลาสสำหรับสร้างกราฟกระแสข้อมูลดีไอพีและใช้เพื่อทดลองประมวลผลภาพ เพื่อให้การสร้างกราฟมีลักษณะเป็นวิชวลที่สามารถจัดเรียงและแก้ไขกราฟได้ง่ายและสะดวกในวิทยานิพนธ์นี้จึงได้ออกแบบระบบบรรณาธิกรขึ้นสำหรับติดต่อกับผู้ใช้โดยระบบนี้กำหนดให้มีบริเวณบรรณาธิกร (Editor Area) สำหรับสร้างและแก้ไขกราฟโดยจัดวางเรียงวัตถุลงบนบริเวณนี้ และออกแบบคลาสเพื่อใช้สร้างวัตถุสำหรับแสดงจุดยอดและเส้นบนบริเวณบรรณาธิกรโดยแบ่งออกเป็นสองกลุ่ม คือ *กลุ่มคลาสไอเท็ม (Item Class)* ได้แก่คลาส CProcess กับคลาส CData และ*กลุ่มคลาสช่องทาง (Channel Class)* ได้แก่คลาส CPort กับคลาส CPipe

คลาสที่กำหนดในแผนภาพคลาสมีหน้าที่ดังต่อไปนี้

1) คลาส CPort คือคลาสที่ใช้เป็นตัวแทนของช่องทางสื่อสารข้อมูลของ Item

2) คลาส CItem คือคลาสที่ใช้เพื่อเป็นจุดยอดของกราฟกระแสข้อมูลดีไอพี

3) คลาส CData คือคลาสที่ทำหน้าที่เก็บข้อมูลชั่วคราว

4) คลาส CProcess คือคลาสที่ทำหน้าที่เป็นตัวแทนของฟังก์ชันประมวลผลภาพ

5) คลาส CGroup คือคลาสที่ทำหน้าที่เป็นตัวแทนกราฟกระแสข้อมูลดีไอพีย่อย

6) คลาส CPipe คือคลาสที่ทำหน้าที่เป็น Pipe ซึ่งเป็นเส้นเชื่อมของกราฟกระแสข้อมูลดีไอพีและใช้เชื่อมโยงข้อมูลระหว่างคลาส CPort ด้วยกัน

คลาสในกลุ่มคลาสไอเท็มจะถูกใช้เพื่อเป็นตัวแทนจุดยอดของกราฟโดยคลาส CProcess ทำหน้าที่เป็นตัวแทนของฟังก์ชันประมวลผลภาพ คลาส CData ใช้แทนข้อมูลและเป็นที่พักข้อมูลชั่วคราว คลาส CGroup ทำหน้าที่เป็นตัวแทนของกราฟกระแสข้อมูลดีไอพีย่อย โดยทั้งสามคลาสสืบทอดมาจากคลาส CItem

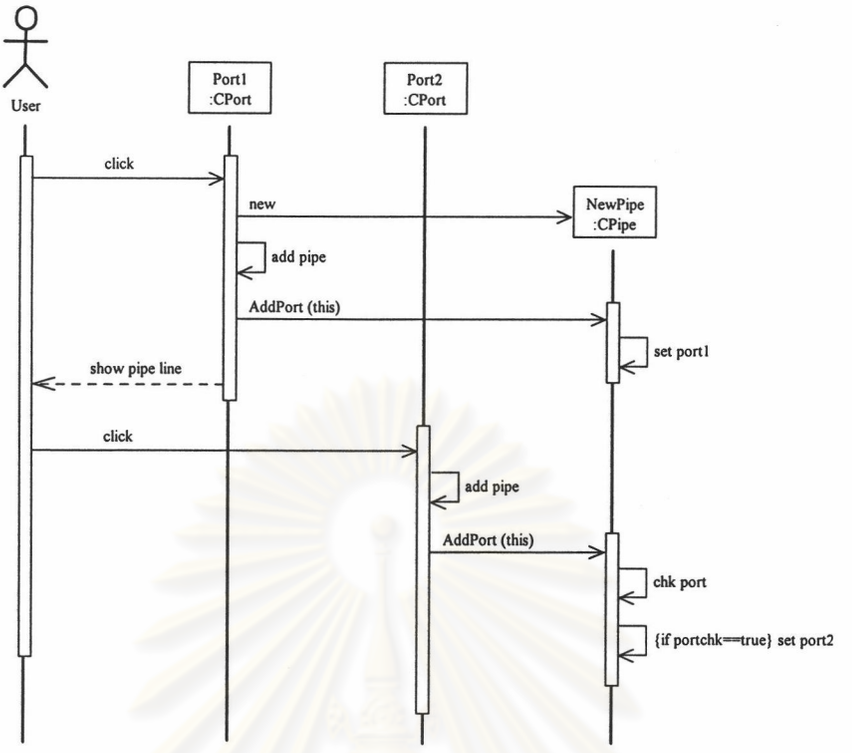
คลาสในกลุ่มคลาสช่องทางจะถูกใช้เพื่อเป็นช่องทางสื่อสารข้อมูลโดยคลาส CPort จะทำหน้าที่เป็นช่องทางสื่อสารข้อมูลให้กับคลาส CItem ส่วนคลาส CPipe จะถูกใช้สำหรับแทนเส้นเชื่อมของกราฟเพื่อกำหนดเส้นทางการไหลของข้อมูลและทำหน้าที่เป็นอุโมงค์ (Tunnel) สำหรับลำเลียงข้อมูลระหว่างคลาส CPort

4.2.2 แผนภาพซีเควนซ์

แผนภาพซีเควนซ์คือแผนภาพที่ใช้สำหรับแสดงการโต้ตอบของวัตถุในระบบ โดยแสดงให้เห็นถึงลำดับการส่งข้อความร้องขอของวัตถุในระบบ แผนภาพซีเควนซ์ที่แสดงต่อไปนี้เป็นแผนภาพที่อธิบายการโต้ตอบของวัตถุในระบบโดยจำแนกตามหน้าที่การทำงานที่แสดงไว้แล้วในแผนภาพยูสเคสดังนี้

1) การสร้างกราฟกระแสข้อมูลดีไอพี

การสร้างกราฟกระแสข้อมูลให้ปรากฏบนหน้าจอผู้ใช้ต้องทำการสร้างจุดยอดของกราฟก่อนโดยเลือกว่าจะสร้างวัตถุข้อมูลจากคลาส CData ซึ่งแสดงเป็น Data Item หรือวัตถุประมวลผลจากคลาส CProcess ซึ่งแสดงเป็น Process Item โดยเลือกฟังก์ชันประมวลผลภาพที่ต้องการจากเมนูจากนั้นผู้ใช้จะต้องคลิกเมาส์ยังบริเวณบรรณาธิกรเพื่อวางวัตถุลงบนหน้าจอโดยระบบบรรณาธิกรจะต้องสร้างวัตถุตามที่ผู้ใช้เลือกแล้ววางลงบนหน้าจอตามตำแหน่งที่ผู้ใช้คลิกเมาส์จากนั้นจึงทำการสร้างเส้นเชื่อมโดยคลิกเมาส์ตรงบริเวณพอร์ตของวัตถุสองวัตถุที่ต้องการเชื่อมเส้นวัตถุท่อซึ่งแสดงเป็น Pipe จะถูกสร้างขึ้นมาจากคลาส CPipe เพื่อเชื่อมโยงระหว่างพอร์ตสองพอร์ต แผนภาพซีเควนซ์ของการเชื่อมเส้นกราฟกระแสข้อมูลดีไอพีแสดงในรูปที่ 4.8



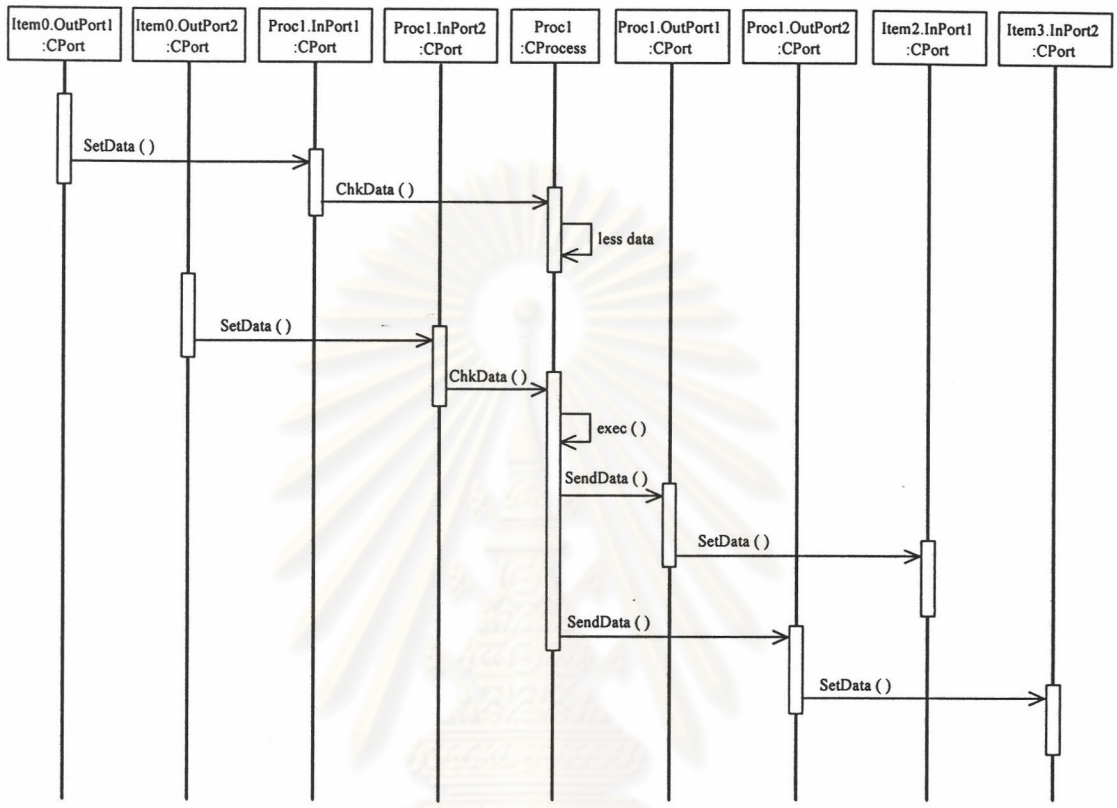
รูปที่ 4.8 แผนภาพซีเควนซ์ของการสร้างเส้นเชื่อมกราฟกระแสข้อมูลดีไอพี

2) การประมวลผลภาพตามกราฟกระแสข้อมูลดีไอพี

หลังจากออกแบบขั้นตอนวิธีในการประมวลผลภาพโดยการสร้างเป็นกราฟกระแสข้อมูลดีไอพีแล้วจำเป็นต้องทำการทดลองประมวลผลภาพเพื่อทดสอบขั้นตอนวิธีในการประมวลผลภาพว่าสามารถทำงานงานได้ถูกต้องตามที่ออกแบบไว้หรือไม่ ซึ่งการใช้วิธีการเชิงวัตถุในการออกแบบและพัฒนาระบบทำให้สามารถเพิ่มคุณสมบัติในการประมวลผลภาพสำหรับเครื่องมือ VisDIP-Tool ได้ง่าย โดยออกแบบให้คลาส CProcess มีคุณสมบัติเพิ่มเติมจาก CItem ที่สืบทอดมาให้สามารถเรียกฟังก์ชันประมวลผลภาพให้ทำการประมวลผลข้อมูลได้

การประมวลผลภาพตามกราฟกระแสข้อมูลผู้ใช้ต้องสั่งให้เครื่องมือทำการประมวลผล จากนั้นระบบประมวลผลจะเริ่มทำงานโดยสั่งให้วัตถุข้อมูลที่เป็นต้นทางของกราฟทั้งหมด (วัตถุข้อมูลที่เป็นต้นทางคือวัตถุข้อมูลที่ไม่มีการเชื่อมต่อท่อที่พอร์ตนำเข้าของวัตถุข้อมูล) ทำการส่งข้อมูล วัตถุข้อมูลจะส่งข้อมูลออกไปที่พอร์ตส่งออกทั้งหมด พอร์ตส่งออกจะส่งข้อมูลต่อไปให้วัตถุท่อที่เชื่อมต่ออยู่ทั้งหมด วัตถุท่อจะลำเลียงข้อมูลระหว่างพอร์ตส่งออกและพอร์ตนำเข้า เมื่อพอร์ตนำเข้าได้รับข้อมูลจากวัตถุท่อจะสั่งให้วัตถุประมวลผลที่เป็นเจ้าของพอร์ตนำเข้าทำการประมวลผลข้อมูลและรอจนกระทั่งประมวลผลข้อมูลเสร็จก็จะส่งข้อมูลผลลัพธ์ออกไปที่พอร์ตส่งออก และจะ

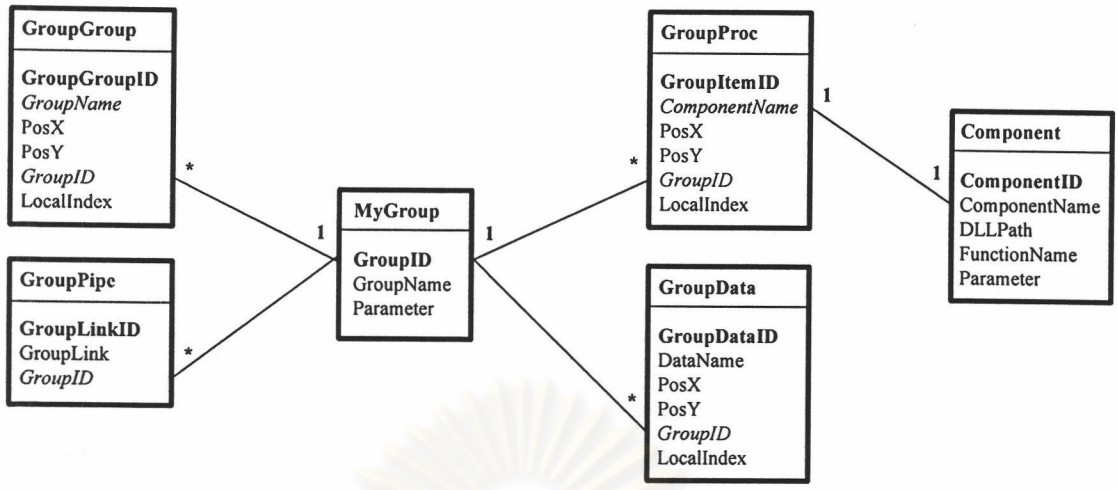
เกิดเหตุการณ์เช่นนี้ซ้ำ ๆ ไปจนกระทั่งครบทั้งกราฟ แผนภาพซีเควนซ์ของการประมวลผลแสดงในรูปที่ 4.9



รูปที่ 4.9 แผนภาพซีเควนซ์ของการประมวลผล

4.3 การออกแบบฐานข้อมูลสำหรับระบบ

ในวิทยานิพนธ์นี้ได้เลือกใช้ระบบจัดการฐานข้อมูลเพื่อใช้สำหรับจัดเก็บข้อมูลฟังก์ชันประมวลผลภาพซึ่งอยู่ในส่วนประกอบดีไอพีที่นำมาเชื่อมต่อเข้ากับเครื่องมือ VisDIP-Tool การออกแบบฐานข้อมูลในหัวข้อนี้ยึดแนวทางการออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational database) โดยแสดงความสัมพันธ์ของข้อมูลผ่านแผนผังแสดงความสัมพันธ์ระหว่างเอนทิตี (Entity Relationship Diagram) ซึ่งประกอบไปด้วย 6 เอนทิตีโดยแบ่งเป็นเอนทิตีที่สำคัญสองกลุ่ม คือ เอนทิตีสำหรับเก็บรายละเอียดของส่วนประกอบดีไอพี 1 เอนทิตี และเอนทิตีสำหรับเก็บข้อมูล Group Item ที่สร้างขึ้นโดยผู้ใช้ 5 เอนทิตี โดยมีรายละเอียดดังรูปที่ 4.10 และในส่วนของคำอธิบายเอนทิตีใช้ในการอธิบายลักษณะประจำ (Attribute) ภายในเอนทิตีโดยมีรายละเอียดดังนี้



รูปที่ 4.10 แผนผังแสดงความสัมพันธ์ระหว่างเอนทิตี

คำอธิบายเอนทิตี

รายละเอียดต่อไปนี้เป็นคำอธิบายชื่อความหมาย รูปแบบและขนาดของฟิลด์ต่าง ๆ โดยแบ่งข้อมูลออกเป็นสองกลุ่ม คือ ฐานข้อมูลสำหรับจัดเก็บรายละเอียดส่วนประกอบดีไอพี และฐานข้อมูลสำหรับจัดเก็บรายละเอียด Group Item ซึ่งมีรายละเอียดดังต่อไปนี้

1. ฐานข้อมูลส่วนประกอบดีไอพี

ฐานข้อมูลส่วนประกอบดีไอพีมีไว้เพื่อจัดเก็บข้อมูลรายละเอียดของฟังก์ชันประมวลผลภาพที่อยู่ในส่วนประกอบดีไอพี เครื่องมือ VisDIP-Tool จะใช้ข้อมูลส่วนนี้เป็นแนวทางในการเรียกใช้งานฟังก์ชันประมวลผลภาพที่อยู่ในส่วนประกอบดีไอพี ฐานข้อมูลส่วนประกอบดีไอพีประกอบไปด้วย 1 ตาราง คือ ตาราง Component

ตาราง Component ใช้สำหรับเก็บข้อมูลรายละเอียดฟังก์ชันประมวลผลภาพและรายละเอียดของส่วนประกอบดีไอพีโดยมี ComponentID เป็นคีย์หลักดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ตาราง Component

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. ComponentID	Long Integer	4 ไบต์	รหัสส่วนประกอบ
2. ComponentName	ตัวอักษร	50	ชื่อฟังก์ชันส่วนประกอบ
3. DLLPath	Memo		ที่อยู่แฟ้มดีแอลแอล
4. FunctionName	Memo		ชื่อฟังก์ชัน
5. Parameter	Memo		รายการพารามิเตอร์

2. ฐานข้อมูลกลุ่มไอเท็ม

ฐานข้อมูลกลุ่มไอเท็มเป็นฐานข้อมูลที่เก็บรายละเอียดของกราฟกระแสข้อมูลดีไอพีย่อย และ Group Item ที่ผู้ใช้สร้างขึ้นเป็นตัวแทนกราฟกระแสข้อมูลดีไอพีย่อยเพื่อลดความซับซ้อนในการแสดงขั้นตอนวิธีในการประมวลผลภาพ โดยมีการอ้างอิง Process Item ที่อยู่ในฐานข้อมูล ส่วนแรก ฐานข้อมูลกลุ่มไอเท็มประกอบไปด้วย 5 ตาราง โดยมี 1 ตารางใช้สำหรับเก็บรายละเอียดของ Group Item และอีก 4 ตารางใช้สำหรับเก็บรายละเอียดของกราฟกระแสข้อมูลดีไอพีย่อย ซึ่งมีรายละเอียดดังต่อไปนี้

2.1 **MyGroup** สำหรับจัดเก็บข้อมูลรายละเอียดของ Group Item เพื่อใช้เป็นตัวแทนของกราฟกระแสข้อมูลดีไอพีย่อยซึ่งประกอบด้วยข้อมูลชื่อของ Group Item ข้อมูล Port นำเข้าทั้งหมดของ Group Item และ ข้อมูล Port ส่งออกทั้งหมดของ Group Item โดยตารางนี้มี GroupID เป็นคีย์หลักดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 ตาราง MyGroup

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. GroupID	Long Integer	4 ไบต์	รหัสกลุ่ม
2. GroupName	ตัวอักษร	50	ชื่อกลุ่ม
3. Parameter	Memo		รายการพารามิเตอร์

2.2 **GroupData** สำหรับเก็บข้อมูล Data Item ภายในกราฟกระแสข้อมูลดีไอพีย่อย โดยชื่อของ Data Item ต้นทางและชื่อของ Data Item ปลายทางจะนำมาใช้เป็นชื่อ Port นำเข้าและชื่อ Port ส่งออกของ Group Item ตารางนี้มี GroupDataID เป็นคีย์หลักและมี GroupID เป็นคีย์นอก ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ตาราง GroupData

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. GroupDataID	Long Integer	4 ไบต์	รหัสกลุ่มข้อมูล
2. DataName	ตัวอักษร	50	ชื่อข้อมูล
3. PosX	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน X
4. PosY	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน Y
5. GroupID*	Long Integer	4 ไบต์	รหัสกลุ่ม
6. LocalIndex	Long Integer	4 ไบต์	ดัชนีภายในกลุ่ม

2.3 **GroupProc** สำหรับเก็บข้อมูล Process Item ภายในกราฟกระแสข้อมูลดีไอพีย่อย โดยตารางนี้มี GroupProcID เป็นคีย์หลักและมี GroupID เป็นคีย์นอกดังแสดงในตารางที่ 4.4

ตารางที่ 4.4 ตาราง GroupProc

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. GroupProcID	Long Integer	4 ไบต์	รหัสกลุ่มประมวลผล
2. ComponentName*	ตัวอักษร	50	ชื่อฟังก์ชันส่วนประกอบ
3. PosX	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน X
4. PosY	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน Y
5. GroupID*	Long Integer	4 ไบต์	รหัสกลุ่ม
6. LocalIndex	Long Integer	4 ไบต์	ดัชนีภายในกลุ่ม

2.4 **GroupGroup** สำหรับเก็บข้อมูล Group Item ที่อยู่ภายในกราฟกระแสข้อมูลดีไอพีย่อย โดยตารางนี้จะมี GroupGroupID เป็นคีย์หลักและมี GroupID เป็นคีย์นอกดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 ตาราง GroupGroup

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. GroupGroupID	Long Integer	4 ไบต์	รหัสกลุ่มประมวลผล
2. GroupName	ตัวอักษร	50	ชื่อกลุ่มฟังก์ชันประมวลผลภาพ
3. PosX	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน X
4. PosY	Long Integer	4 ไบต์	ตำแหน่งพิกัดแกน Y
5. GroupID*	Long Integer	4 ไบต์	รหัสกลุ่ม
6. LocalIndex	Long Integer	4 ไบต์	ดัชนีภายในกลุ่ม

2.5 **GroupPipe** สำหรับเก็บข้อมูลการเชื่อมโยง Port ของ Item ต่าง ๆ ซึ่งถูกระบุโดย Pipe ภายในกราฟกระแสข้อมูลดีไอพีย่อย โดยตารางนี้มี GroupLinkID เป็นคีย์หลักและมี GroupID เป็นคีย์นอกดังแสดงในตารางที่ 4.6

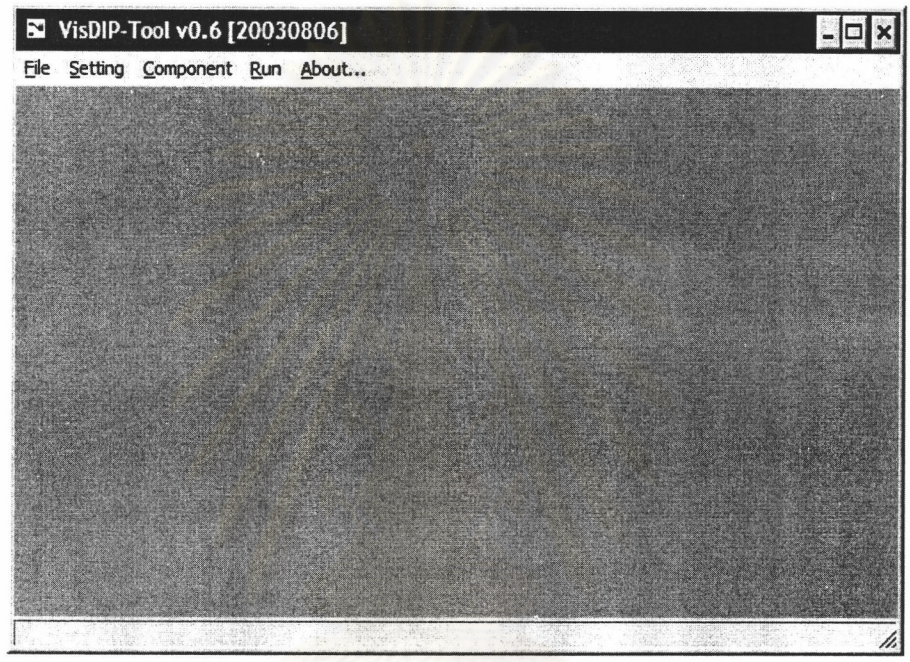
ตารางที่ 4.6 ตาราง GroupPipe

ชื่อเขตข้อมูล	ชนิด	ขนาด	ความหมาย
1. GroupLinkID	Long Integer	4 ไบต์	รหัสกลุ่มเชื่อมโยง
2. GroupLink	Memo		ข้อมูลเชื่อมโยงกลุ่ม
3. GroupID*	Long Integer	4 ไบต์	รหัสกลุ่ม

4.4 การออกแบบส่วนติดต่อผู้ใช้

4.4.1 หน้าต่างหลัก

หน้าต่างหลักของเครื่องมือประมวลผลภาพดิจิทัล VisDIP-Tool เป็นหน้าต่างที่มีบริเวณบรรณาธิกรสำหรับสร้างกราฟกระแสข้อมูลดีไอพีดังแสดงในรูปที่ 4.11 โดยได้กำหนดให้มีองค์ประกอบดังต่อไปนี้



รูปที่ 4.11 หน้าต่างหลักของบรรณาธิกรสำหรับสร้างกราฟกระแสข้อมูลดีไอพี

1) เมนู File

เป็นเมนูสำหรับจัดการแฟ้มโครงการที่ใช้บันทึกกราฟกระแสข้อมูลดีไอพี

2) เมนู Setting

เป็นเมนูสำหรับกำหนดการทำงานต่าง ๆ ของระบบของเครื่องมือ VisDIP-Tool เช่น เลือกที่อยู่แฟ้มฐานข้อมูลให้กับเครื่องมือ

3) เมนู Component

เป็นเมนูสำหรับเลือกไอเท็มเพื่อนำมาวางบนหน้าจอเพื่อสร้างกราฟกระแสข้อมูลดีไอพี เมนูย่อยสำหรับสร้าง Group Item และเมนูย่อยสำหรับจัดการเพิ่มลดส่วนประกอบดีไอพี

4) เมนู Run

ใช้สำหรับสั่งให้เครื่องมือดำเนินการทดลองประมวลผลภาพตามกราฟกระแสข้อมูลดีไอพีบนหน้าจอที่ออกแบบไว้

5) เมนู About

แสดงรายละเอียดเกี่ยวกับซอฟต์แวร์นี้

6) แถบสถานะ

แถบสถานะคือแถบที่ใช้แสดงถึงสถานะปัจจุบันของเครื่องมือ แถบสถานะอยู่ตอนล่างของหน้าต่างมีหน้าที่แสดงสถานะต่าง ๆ ให้ผู้ใช้ทราบ เช่น ตำแหน่งของเมาส์บนหน้าจอ

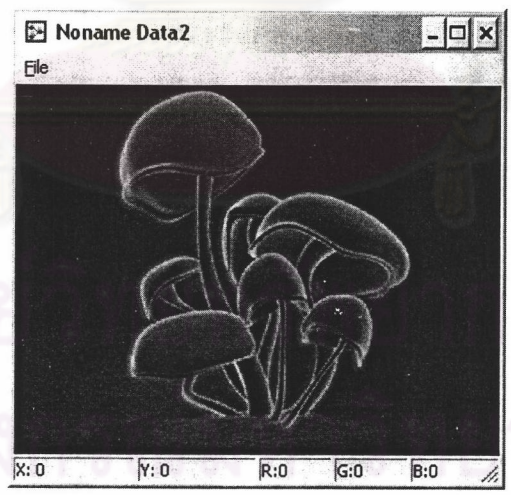
7) บริเวณบรรณาธิกร

บริเวณบรรณาธิกรคือพื้นที่ตรงกลางของหน้าต่างหลักสำหรับใช้วาดกราฟกระแสข้อมูลดีไอพี

4.4.2 หน้าต่างแสดงและแก้ไขข้อมูล

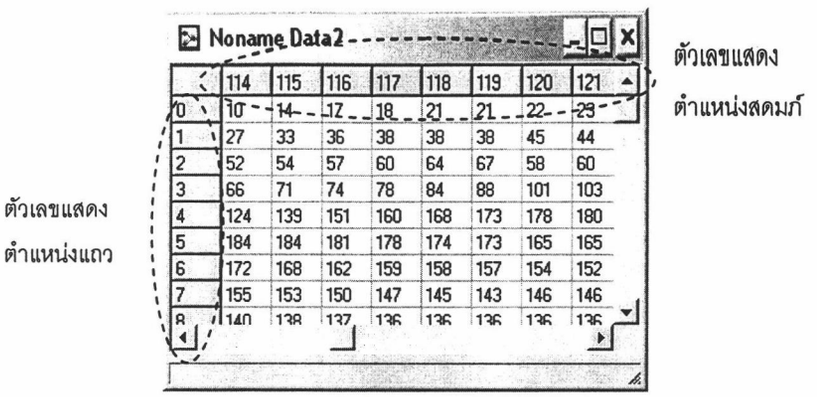
หน้าต่างสำหรับแสดงและแก้ไขข้อมูลคือหน้าจอที่ใช้สำหรับแสดงข้อมูลที่อยู่ใน Data Item บนบริเวณบรรณาธิกร การแสดงข้อมูลแบ่งออกเป็นสองลักษณะคือ แสดงข้อมูลเป็นภาพบิตแมป และแสดงข้อมูลเป็นตารางตัวเลข

การแสดงข้อมูลเป็นภาพบิตแมปคือการแสดงข้อมูลเป็นความเข้มแสงหรือเฉดสีบนจอคอมพิวเตอร์ดังรูปที่ 4.12 ส่วนการแสดงข้อมูลเป็นตารางตัวเลขคือการแสดงข้อมูลเป็นค่าตัวเลขบนตาราง ดังรูปที่ 4.13 ซึ่งผู้ใช้ควรสามารถแก้ไขตัวเลขในตารางได้



รูปที่ 4.12 หน้าต่างแสดงข้อมูลเป็นบิตแมป

ด้านล่างของหน้าต่างแสดงภาพออกแบบให้มีแถบสถานะซึ่งเมื่อเลื่อนเมาส์บนบริเวณภาพในหน้าต่าง แถบสถานะจะแสดงตำแหน่งพิกัด (x , y) ของภาพและแสดงค่าแม่สีแสงของจุดภาพที่ตำแหน่งพิกัด (x , y) ออกมา

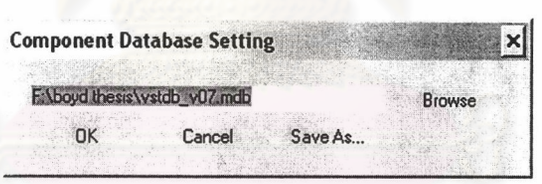


รูปที่ 4.13 หน้าต่างแสดงข้อมูลเป็นตัวเลขในตาราง

ตารางในหน้าต่างแสดงข้อมูลจะถูกแบ่งเป็นแถวและสดมภ์ซึ่งจะอ้างอิงตำแหน่งที่ตรงกับพิกัด (x , y) ในหน้าต่างแสดงภาพ

4.4.3 หน้าต่างสำหรับตั้งค่าเลือกใช้ฐานข้อมูล

หน้าต่างสำหรับตั้งค่าเลือกใช้ฐานข้อมูลคือหน้าต่างที่ใช้แสดงและกำหนดแฟ้มฐานข้อมูลให้กับเครื่องมือ VisDIP-Tool ดังแสดงในรูปที่ 4.14 ค่าที่แสดงในหน้าต่างนี้จะเป็นตำแหน่งของแฟ้มฐานข้อมูลบนเครื่องคอมพิวเตอร์

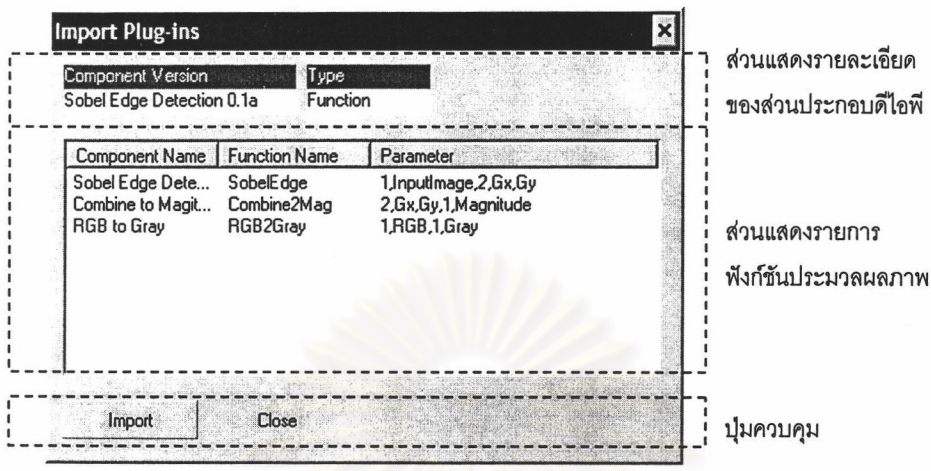


รูปที่ 4.14 หน้าต่างสำหรับตั้งค่าเลือกใช้ฐานข้อมูล

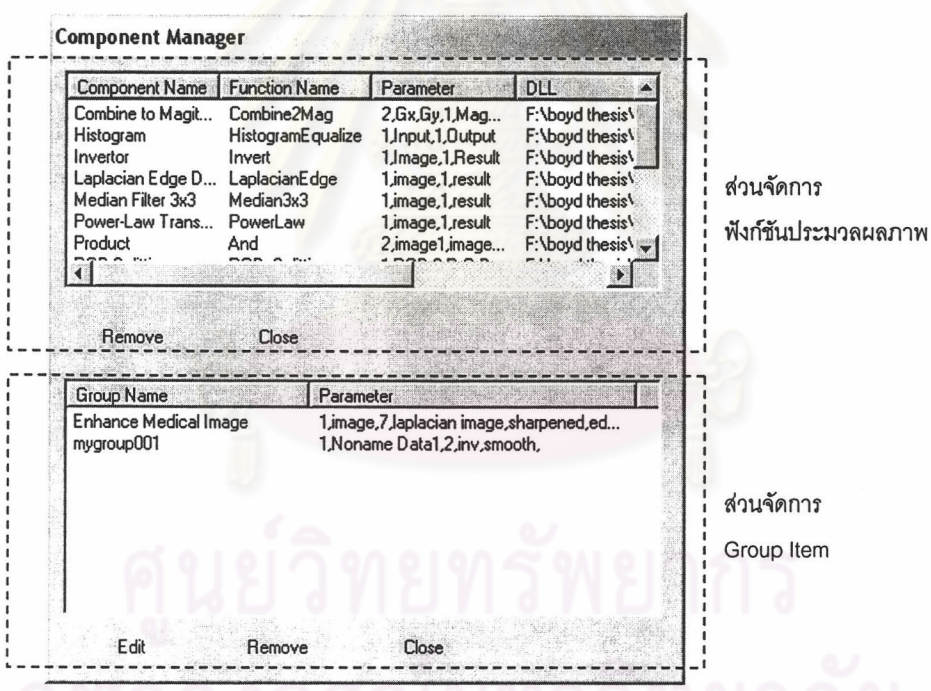
4.4.4 หน้าต่างสำหรับจัดการส่วนประกอบ

การจัดการส่วนประกอบในงานวิจัยนี้แบ่งเป็นสองงานคือ การเพิ่มส่วนประกอบ และการลดส่วนประกอบ การเพิ่มส่วนประกอบคือการเพิ่มฟังก์ชันประมวลผลภาพให้กับระบบของเครื่องมือ VisDIP-Tool หน้าต่างสำหรับเพิ่มส่วนประกอบดีไอพีออกแบบหน้าจอแบ่งเป็นสามส่วนดังแสดงในรูปที่ 4.15 ส่วนแรกเพื่อใช้แสดงละเอียดของส่วนประกอบดีไอพี ได้แก่ รุ่นและชนิด ส่วนที่สองเป็นส่วนที่แสดงรายการฟังก์ชันประมวลผลภาพที่อยู่ในส่วนประกอบดีไอพีเพื่อให้ผู้ใช้สามารถเลือกเพียงบางฟังก์ชันที่ต้องการเพิ่มได้ และส่วนที่สามคือปุ่มควบคุมสำหรับสั่งให้ระบบทำการเพิ่มฟังก์ชันประมวลผลภาพโดยบันทึกรายละเอียดลงฐานข้อมูล การลดส่วนประกอบคือการลบข้อมูลฟังก์ชันประมวลผลภาพที่ไม่ต้องการออกจากฐานข้อมูล โดยหน้าจอนี้แบ่งการจัดการเป็น

สองส่วนดังแสดงในรูปที่ 4.16 คือ ส่วนการจัดการฟังก์ชันประมวลผลภาพ และ ส่วนการจัดการ Group Item



รูปที่ 4.15 หน้าต่างสำหรับเพิ่มส่วนประกอบดีไอพี

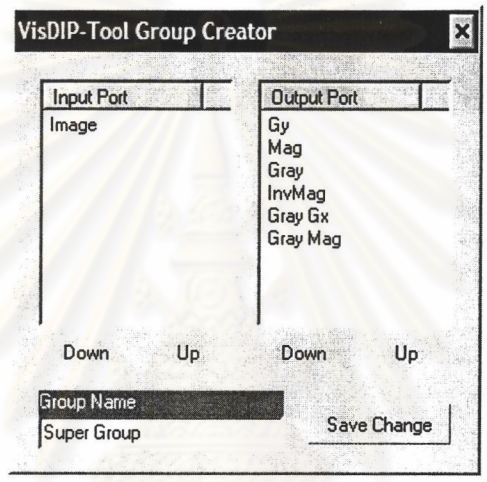


รูปที่ 4.16 หน้าต่างสำหรับลดส่วนประกอบดีไอพี

4.4.5 หน้าต่างสำหรับสร้างและแก้ไขกลุ่ม

หน้าต่างสำหรับสร้างและแก้ไขกลุ่ม คือ หน้าต่างสำหรับสร้างและแก้ไข Group Item ซึ่งใช้เป็นตัวแทนของกราฟกระแสข้อมูลดีไอพีย่อยทำให้ผู้ใช้สามารถลดความซับซ้อนของกราฟกระแสข้อมูลดีไอพีเพื่อความสะดวกในการออกแบบขั้นตอนวิธีในการประมวลผลภาพ ผู้ใช้

สามารถสร้าง Group Item ได้โดยการสร้างกราฟกระแสข้อมูลดีไอพีที่สมบูรณ์บนหน้าจอหลักแล้วเลือกเมนู "Component -> Create Group" หน้าต่างสำหรับสร้างกลุ่มจะปรากฏบนหน้าจอดังแสดงในรูปที่ 4.17 หน้าจอนี้จะแสดงชื่อ Group Item ซึ่งผู้ใช้สามารถแก้ไขเปลี่ยนแปลงได้ ชื่อ Port นำเข้าและชื่อ Port ส่งออกทั้งหมดของ Group Item ซึ่งในหน้าจอนี้ผู้ใช้ควรสามารถย้ายลำดับตำแหน่งของ Port ได้ จึงออกแบบให้มีปุ่ม Down และ Up เพื่อใช้เลื่อนลำดับของ Port ขึ้นหรือลง หน้าจอนี้จะเป็นหน้าจอเดียวกับหน้าจอสำหรับแก้ไขกลุ่มซึ่งผู้ใช้สามารถเปลี่ยนลำดับตำแหน่งของพอร์ตและสามารถแก้ชื่อ Group Item ได้ หน้าจอทั้งสองแบบจะต่างกันคือปุ่ม Save Change จะเปลี่ยนเป็น Create เมื่อเป็นหน้าจอสำหรับสร้างกลุ่ม



รูปที่ 4.17 หน้าต่างสำหรับสร้างและแก้ไขกลุ่ม

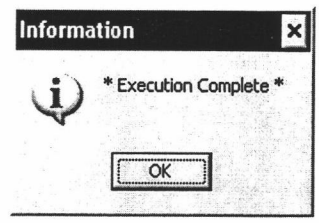
4.4.6 การแสดงสถานะ ข่าวสารและข้อผิดพลาด

การแสดงสถานะ ข่าวสาร และข้อผิดพลาด คือ การโต้ตอบกับผู้ใช้โดยผ่านทางส่วนติดต่อผู้ใช้แบบกราฟิกซึ่งในวิทยานิพนธ์นี้ได้แบ่งเป็น 3 ประเภทคือ การแสดงสถานะ การแจ้งข่าวสาร และการแจ้งข้อผิดพลาด

1) การแสดงสถานะในวิทยานิพนธ์นี้คือการแสดงสถานะของฟังก์ชันประมวลผลภาพโดยจะแสดงเป็นสัญญาณสีที่ Item บนบริเวณบรรณาธิกรโดยตรง ซึ่งมีทั้งหมด 4 สถานะได้แก่

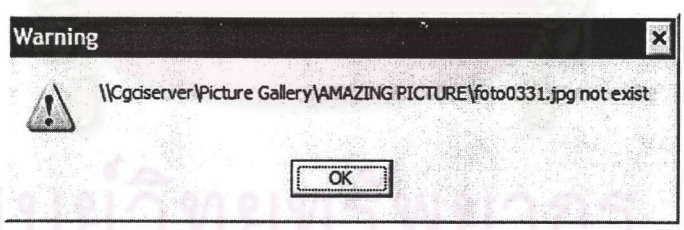
- พร้อมทำงาน (Standby) – วัตถุที่อยู่ในสถานะนี้จะแสดงสีกรอบ Item เป็นสีดำ ซึ่งแสดงถึงสถานะปกติ
- กำลังทำงาน – วัตถุที่อยู่ในสถานะนี้จะแสดงสีกรอบ Item เป็นสีแดง
- ทำงานเสร็จสิ้น – วัตถุที่อยู่ในสถานะนี้จะแสดงสีกรอบ Item เป็นสีเขียว
- ถูกเลือก (Selected) – วัตถุที่อยู่ในสถานะนี้จะแสดงสีกรอบ Item เป็นสีน้ำเงิน

2) การแจ้งข่าวสาร หมายถึงการแสดงข้อความอธิบายการทำงานของระบบให้ผู้ใช้รับทราบ การแสดงข่าวสารจะแสดงโดยผ่านหน้าต่างโต้ตอบแบบผุดขึ้น ดังรูปที่ 4.18

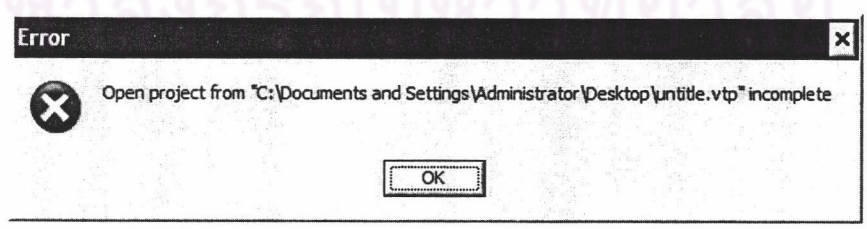


รูปที่ 4.18 หน้าต่างแจ้งข่าวสารแบบผุดขึ้น

3) การแจ้งข้อผิดพลาด ในวิทยานิพนธ์นี้หมายถึงการแจ้งข้อความอธิบายถึงข้อผิดพลาดที่เกิดขึ้นขณะประมวลผลซึ่งจะขึ้นอยู่กับรหัสข้อผิดพลาด (Error code) ที่ได้รับจากฟังก์ชันประมวลผลภาพ รวมถึงการแจ้งข้อความอธิบายถึงข้อผิดพลาดของระบบการทำงานอื่น ๆ ของเครื่องมือ VisDIP-Tool ซึ่งต้องแบ่งประเภทรูปแบบในการแสดงข้อความเป็นสองแบบ คือ การแจ้งเตือนและการแจ้งข้อผิดพลาด การแจ้งเตือนนั้นหมายถึงการแจ้งข้อความแสดงความผิดพลาดหรือความผิดปกติที่ไม่ร้ายแรงซึ่งระบบสามารถละเลยเพื่อทำงานต่อได้ การแจ้งข้อผิดพลาดคือการแจ้งความผิดปกติที่ทำให้ระบบไม่สามารถทำงานต่อได้ ระบบจะหยุดการทำงานนั้น ๆ แล้วเข้าสู่โหมดเตรียมพร้อม (Standby) การแสดงข้อความจะแสดงโดยผ่านหน้าต่างโต้ตอบแบบผุดขึ้น ดังรูปที่ 4.19 ถึง 4.20



รูปที่ 4.19 หน้าต่างการแจ้งเตือนแบบผุดขึ้น



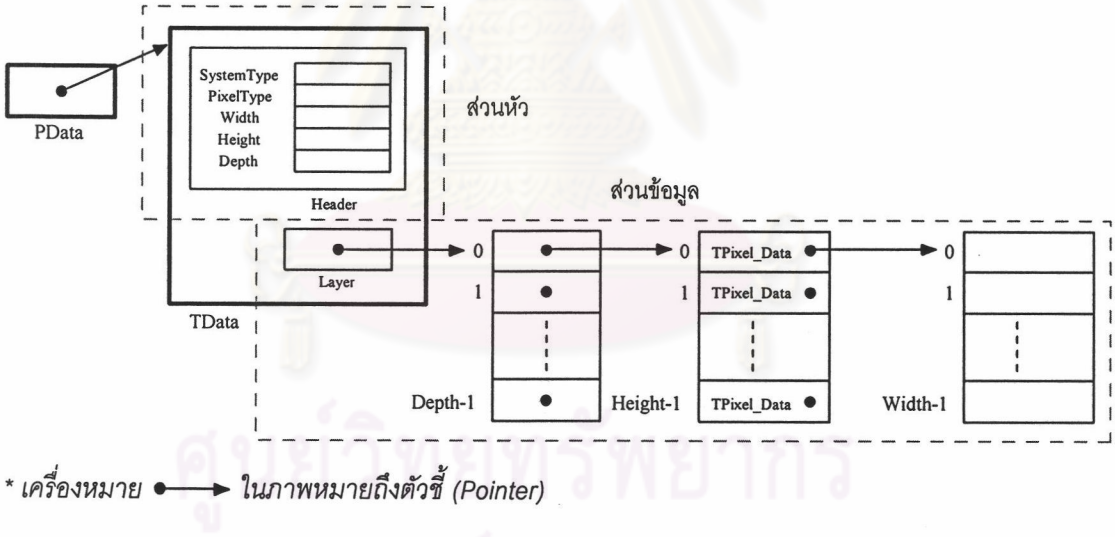
รูปที่ 4.20 หน้าต่างแจ้งข้อผิดพลาดแบบผุดขึ้น

4.5 การออกแบบส่วนประกอบดีไอพี

ส่วนประกอบซอฟต์แวร์สำหรับใช้งานทางด้านการประมวลผลภาพในวิทยานิพนธ์นี้มีชื่อเรียกว่า *ส่วนประกอบดีไอพี (DIP Component)* ในบทนี้จะกล่าวถึงการออกแบบส่วนประกอบดีไอพีดังกล่าวซึ่งจะครอบคลุมถึง การออกแบบโครงสร้างข้อมูล การออกแบบเอพีไอ และ การกำหนดโครงแบบ

4.5.1 โครงสร้างข้อมูลสำหรับดีไอพี

เนื่องจากโปรแกรมภาษาแต่ละโปรแกรมมีชนิดข้อมูล (Data Type) สำหรับเก็บข้อมูลภาพไม่เหมือนกัน ดังนั้นจึงต้องออกแบบโครงสร้างข้อมูล [9] ชนิดใหม่ขึ้นเพื่อใช้งานร่วมกันระหว่างโปรแกรมภาษาแต่ละโปรแกรม โดยหัวข้อนี้เป็นการออกแบบโครงสร้างข้อมูลเพื่อใช้เป็นสื่อสำหรับเก็บและสื่อสารข้อมูลทางด้านการประมวลผลภาพ โครงสร้างข้อมูลนี้ต้องสามารถรองรับข้อมูลภาพได้หลากหลายชนิดและสามารถบรรจุข้อมูลได้สูงสุด 3 มิติ เพื่อให้โครงสร้างข้อมูลนี้สามารถนำไปใช้เก็บชุดข้อมูลได้ เช่น ภาพเคลื่อนไหว ชุดข้อมูลภาพตัดขวางเอ็มอาร์ไอ เป็นต้น



รูปที่ 4.21 โครงสร้างข้อมูลสำหรับการประมวลผลภาพ

จากรูปที่ 4.21 แสดงโครงสร้างข้อมูลสำหรับการประมวลผลภาพดิจิทัลที่ชื่อว่า PData ซึ่งมีโครงสร้างข้อมูลที่สำคัญสองส่วนได้แก่ส่วนหัวสำหรับเก็บรายละเอียดของข้อมูลและส่วนข้อมูลเป็นส่วนสำหรับเก็บข้อมูล โดยมีรหัสโปรแกรม (Source Code) แสดงดังรูปที่ 4.22 และมีรายละเอียดดังต่อไปนี้

```
struct TPixel_Data
{
      union
      {
            char*          scVal;
            unsigned char* ucVal;
            signed short*  ssVal;
            unsigned short* usVal;
            signed int*    siVal;
            unsigned int*  uiVal;
            bool*          bVal;
            signed long*   slVal;
            unsigned long* ulVal;
            float*         fltVal;
            double*        sdVal;
            long double*   sldVal;
      };
};
typedef TPixel_Data* PData_Layer;
typedef TPixel_Data** PPData_Layer;
struct TData_Header {
      int SystemType;
      int PixelType;
      int Width,Height,Depth;
};
typedef TData_Header* PData_Header;
struct TData {
      TData_Header Header;
      PPData_Layer Layer;
};
typedef TData * PData;
```

รูปที่ 4.22 รหัสโปรแกรมโครงสร้างข้อมูลอิงตามภาษา C/C++

ส่วนหัวในโครงสร้างข้อมูล PData ส่วนที่ใช้สำหรับเก็บรายละเอียดต่าง ๆ ของข้อมูลที่จำเป็นต่อการประมวลผลภาพ โดยข้อมูลที่เก็บในส่วนหัวมีรายละเอียดดังต่อไปนี้

1) SystemType: คือ ตัวแปรสำหรับเก็บรหัสแทนชนิดของข้อมูลเช่น 1 แทน B/W, 2 แทน Grayscale, 3 แทน RGB, 9 แทน YUV เป็นต้น รหัสชนิดข้อมูลต่าง ๆ แสดงในตารางที่ 4.7

2) PixelType: คือ ตัวแปรสำหรับเก็บชนิดของจุดภาพซึ่งใช้กำหนดความละเอียดของจุดภาพด้วย เช่น unsigned char จะได้ความละเอียดจุดภาพเป็น 8 บิต/จุดภาพ เป็นต้น รหัสสำหรับชนิดจุดภาพแสดงในตารางที่ 4.8

3) Width: คือ ตัวแปรสำหรับเก็บความกว้างของภาพ มีหน่วยเป็นจุดภาพ

4) Height: คือ ตัวแปรสำหรับเก็บความสูงของภาพ มีหน่วยเป็นจุดภาพ

5) Depth: คือ ตัวแปรสำหรับเก็บความลึกของภาพ มีหน่วยเป็นชั้นหรือเฟรม

ตารางที่ 4.7 รหัสชนิดข้อมูลสำหรับโครงสร้างข้อมูล PData

รหัส SystemType	ชนิดของภาพดิจิทัล
0	ไม่ระบุ
1	ขาว/ดำ (B/W: Black & White)
2	ระดับเทา (Grayscale)
3	สีอาร์จีบี (RGB color model)
4	สีซีเอ็มวาย (CMY color model)
5	สีเอชเอสไอ (HSI color model)
6	จำนวนเชิงซ้อน (Complex Number)
7	ป้าย (Label)
8	ภาพแท่งความถี่ (Histogram)
9	สีวายยูวี (YUV color model)
10	สีซีเอ็มวายเค (CMYK color model)

ตารางที่ 4.8 รหัสชนิดจุดภาพสำหรับโครงสร้างข้อมูล PData

รหัส PixelType	ชนิดข้อมูลของจุดภาพ	ขอบเขตข้อมูล
0	8 บิต (signed char)	-128 <= X <= 127
1	8 บิต (unsigned char)	0 <= X <= 255
2	16 บิต (signed short)	-32,768 <= X <= 32,767
3	16 บิต (unsigned short)	0 <= X <= 65,536
4	32 บิต (signed int)	-2,147,483,648 <= X <= 2,147,483,647
5	32 บิต (unsigned int)	0 <= X <= 4,294,967,295
6	1 บิต (bool)	true / false
7	32 บิต (signed long)	-2,147,483,648 <= X <= 2,147,483,647
8	32 บิต (unsigned long)	0 <= X <= 4,294,967,295
9	32 บิต (float)	1.18 10^-38 < |X| < 3.40 10^38 (ทศนิยม 7 ตำแหน่ง)
10	64 บิต (signed double)	2.23 10^-308 < |X| < 1.79 10^308 (ทศนิยม 15 ตำแหน่ง)
11	80 บิต (signed long double)	3.37 10^-4932 < |X| < 1.18 10^4932 (ทศนิยม 18 ตำแหน่ง)

ส่วนข้อมูลในโครงสร้างข้อมูล PData จะเป็นชนิดตัวชี้ (Pointer) สามชั้น โดยเวลาใช้งานต้องจองหน่วยความจำทีละชั้นจนครบทั้งสามชั้น ในการจองหน่วยความจำชั้นแรกจะจองข้อมูลเท่ากับจำนวนชั้นของภาพ (ถ้าใช้กับภาพเคลื่อนไหวหมายถึงจำนวนเฟรม) ซึ่งถ้าใช้เก็บข้อมูลภาพ 2 มิติ จะมีจำนวนชั้นเท่ากับ 1 จากนั้นชั้นที่สองจะต้องจองหน่วยความจำเท่ากับความสูงของภาพ และสุดท้ายต้องจองหน่วยความจำสำหรับชั้นที่สามเท่ากับความกว้างของภาพ ตัวอย่างรหัสโปรแกรมในการจองหน่วยความจำสำหรับโครงสร้างข้อมูลแสดงในภาคผนวก ก

4.5.2 เอพีไอสำหรับดีไอพี (DIP's API)

เอพีไอ (API: Application Program Interface) คือวิธีการเฉพาะที่ทำให้โปรแกรมเมอร์สามารถเขียนโปรแกรมเพื่อเรียกใช้งานซอฟต์แวร์อื่นได้ ซึ่งส่วนมากจะเป็นฟังก์ชันที่ซอฟต์แวร์อื่นให้บริการเพื่อติดต่อหรือสั่งงานซอฟต์แวร์ ในวิทยานิพนธ์นี้นำหลักการเอพีไอมาใช้เพื่อเป็นช่องทางสำหรับติดต่อสั่งงานฟังก์ชันประมวลผลภาพในส่วนประกอบดีไอพีให้ทำงาน และเนื่องจากฟังก์ชันประมวลผลภาพมีมากมายและหลากหลายดังนั้นจึงต้องมีการกำหนดรูปแบบของเอพีไอโดยให้ชื่อว่า*เอพีไอสำหรับดีไอพี*ซึ่งก็คือการกำหนดรูปแบบในการประกาศฟังก์ชันประมวลผลภาพเพื่อให้การประกาศฟังก์ชันประมวลผลภาพเป็นไปในรูปแบบเดียวกันและเป็นระเบียบ สามารถนำไปสร้างเป็นส่วนประกอบดีไอพีได้ การกำหนดรูปแบบการประกาศฟังก์ชันมีประโยชน์ทำให้สามารถรู้วิธีใช้งานฟังก์ชันที่แน่นอนและเมื่อนำไปใช้งานร่วมกับเครื่องมือ VisDIP-Tool จะสามารถนำฟังก์ชันเหล่านี้มาประกอบกันเป็นระบบเพื่อใช้งานร่วมกันได้

```
int DIP_Function_Name (PData input1, PData input2, ... , PData inputN,
                       PData &output1, PData &output2, ... , PData &outputM);
```

รูปที่ 4.23 วากยสัมพันธ์การประกาศฟังก์ชันประมวลผลภาพ

จากรูปที่ 4.23 แสดงวากยสัมพันธ์การประกาศฟังก์ชันประมวลผลภาพโดยอ้างอิงตามภาษา C/C++ ซึ่งแต่ละตำแหน่งมีความหมายดังนี้

- int คือชนิดข้อมูลสำหรับส่งกลับ (Return type) จากฟังก์ชันประมวลผลภาพ ซึ่งกำหนดให้เป็น int เพื่อใช้บอกข้อผิดพลาดที่ทำให้ไม่สามารถประมวลผลข้อมูลได้โดยกำหนดให้ใช้รหัสข้อผิดพลาดตามตารางที่ 4.9
- DIP_Function_Name คือชื่อของฟังก์ชันประมวลผลภาพ
- PData คือชนิดข้อมูลพารามิเตอร์ของฟังก์ชันประมวลผลภาพซึ่งเป็นไปตามโครงสร้างข้อมูลในหัวข้อ 4.5.1
- input1, input2, ..., inputN คือรายชื่อพารามิเตอร์นำเข้าทั้งหมดของฟังก์ชันประมวลผลภาพ
- output1, output2, ..., outputM คือรายชื่อพารามิเตอร์ส่งออกทั้งหมดของฟังก์ชันประมวลผลภาพ

I21175834

ตารางที่ 4.9 รหัสข้อผิดพลาดทั้งหมดสำหรับระบบของเครื่องมือ VisDIP-Tool

รหัสข้อผิดพลาด	ข้อความอธิบายข้อผิดพลาด
0	ปกติ
-1	ยกเลิกโดยผู้ใช้ (User cancel)
-2	ข้อมูลนำเข้าว่างเปล่า (NULL Input)
-3	ชนิดข้อมูลไม่ถูกต้อง (Incorrect SystemType)
-4	ความละเอียดหรือชนิดจุดภาพไม่ถูกต้อง (Invalid PixelType)
-5	ค้นหาไม่พบ (Find not found)
-6	ขนาดภาพไม่ถูกต้อง
-126	เกิดความผิดพลาดภายใน (Internal error)
-127	ข้อผิดพลาดที่ไม่สามารถระบุได้ (Unknown error)

รหัสข้อผิดพลาดดังแสดงในตารางที่ 4.9 เป็นรหัสที่กำหนดขึ้นเองเพื่อใช้กับเครื่องมือในวิทยานิพนธ์นี้ โดยแบ่งรหัสเป็นสองกลุ่มคือ รหัสข้อผิดพลาดที่สามารถระบุได้จะใช้หมายเลขรหัสตั้งแต่ -1 ลงไป และรหัสข้อผิดพลาดที่ไม่สามารถระบุได้จะใช้หมายเลขรหัสตั้งแต่ -127 ขึ้นมาซึ่งในขณะนี้มีรหัสข้อผิดพลาดที่สามารถระบุตั้งแต่ -1 ถึง -6 และรหัสข้อผิดพลาดที่ไม่สามารถระบุได้ตั้งแต่ -127 ถึง -126 ซึ่งเมื่อมีการพัฒนาเครื่องมือรุ่นต่อไปจะทำให้สามารถเพิ่มเติมรหัสข้อผิดพลาดได้โดยยังคงการแบ่งกลุ่มของรหัสข้อผิดพลาดได้อย่างชัดเจน

4.5.3 โครงแบบ (Configuration)

โครงแบบในวิทยานิพนธ์นี้หมายถึงส่วนที่ใช้เพื่อแสดงรายละเอียดต่าง ๆ ของส่วนประกอบดีไอพี โดยโครงแบบนี้จะถูกออกแบบให้รวมกับส่วนประกอบดีไอพีเป็นแฟ้มเดียวกันเพื่อความสะดวกและเป็นระเบียบ โครงแบบนี้สามารถให้ข้อมูลกับซอฟต์แวร์ได้ในขณะทำงาน (Runtime) ทำให้สามารถรู้รายละเอียดฟังก์ชันประมวลผลภาพที่อยู่ในส่วนประกอบดีไอพีได้แบบพลวัต (Dynamic) โครงแบบที่ใช้ในส่วนประกอบดีไอพีกำหนดให้เขียนเป็นฟังก์ชันจำนวน 5 ฟังก์ชันซึ่งมีรายละเอียดดังต่อไปนี้

1) int ComponentType () คือฟังก์ชันที่ทำหน้าที่ส่งค่ากลับเป็นรหัสแสดงชนิดของส่วนประกอบซึ่งในวิทยานิพนธ์นี้มีเพียงหนึ่งชนิด คือ รหัสหมายเลข "1" ส่วนประกอบชนิดฟังก์ชันประมวลผลภาพ ฟังก์ชันโครงแบบนี้ออกแบบมาเพื่อรองรับการเพิ่มชนิดของส่วนประกอบชนิดอื่น ๆ ในอนาคต เช่น รหัสหมายเลข "2" ส่วนประกอบชนิดฟังก์ชันสำหรับโหลดและบันทึกแฟ้มข้อมูลภาพชนิดใหม่ เป็นต้น

2) char *ComponentVersion () คือฟังก์ชันที่ทำหน้าที่แสดงรุ่นและรายละเอียดต่าง ๆ ของส่วนประกอบโดยส่งข้อมูลกลับมาเป็นสายอักขระเพื่อบอกให้ผู้ใช้ทราบ ผู้สร้างส่วนประกอบสามารถกำหนดข้อมูลรายละเอียดต่าง ๆ เกี่ยวกับส่วนประกอบดีไอพีลงในฟังก์ชันนี้ เช่น รุ่น ผู้ผลิต วันที่แจกจ่าย เป็นต้น โดยข้อมูลเหล่านี้ต้องส่งกลับพร้อม ๆ กันเป็นสายอักขระเดียวกันโดยคั่นด้วยรหัสขึ้นบรรทัดใหม่ "\n"

3) int GetFunctionCount () คือฟังก์ชันที่ส่งข้อมูลกลับเป็นตัวเลขเพื่อแสดงจำนวนฟังก์ชันประมวลผลภาพทั้งหมดที่สามารถเรียกใช้งานได้จากส่วนประกอบดีไอพี

4) char *GetFunction (int index) คือฟังก์ชันที่แสดงรายละเอียดของฟังก์ชันประมวลผลภาพเพื่อเรียกใช้งานโดยต้องระบุค่า index ซึ่งมีค่าระหว่าง 0 ถึง GetFunctionCount ()-1 โดยฟังก์ชันโครงแบบนี้ต้องส่งข้อมูลกลับเป็นสายอักขระซึ่งมีรูปแบบดังต่อไปนี้

```
"Component_Name,DIP_Function_Name,
 InputCount,input1name,input2name, … inputNname,
 OutputCount,output1name,output2name, … outputMname"
```

รูปที่ 4.24 รูปแบบรายละเอียดของฟังก์ชันประมวลผลภาพ

จากรูปที่ 4.24 แสดงรูปแบบรายละเอียดของฟังก์ชันประมวลผลภาพสำหรับใช้เป็นค่าส่งกลับจากฟังก์ชันโครงแบบ GetFunction ซึ่งแต่ละตำแหน่งมีความหมายดังต่อไปนี้

- Component_Name คือข้อความที่ใช้แสดงชื่อของวิธีการประมวลผลภาพที่นำมาสร้างเป็นฟังก์ชันประมวลผลภาพ เช่น "Sobel Edge Detection" และใช้แสดงเป็นชื่อของ Item บนบริเวณบรรณาธิกรของเครื่องมือ
- DIP_Function_Name คือชื่อของฟังก์ชันประมวลผลภาพที่อยู่ในส่วนประกอบดีไอพีเพื่อใช้สำหรับอ้างอิงเวลาเรียกใช้
- InputCount คือตัวเลขจำนวนพารามิเตอร์นำเข้าของฟังก์ชันประมวลผลภาพ
- input1name..inputNname คือรายชื่อพอร์ตนำเข้าที่ตรงกับตำแหน่งพารามิเตอร์นำเข้าของฟังก์ชันประมวลผลภาพ
- OutputCount คือตัวเลขจำนวนพารามิเตอร์ส่งออกของฟังก์ชันประมวลผลภาพ
- output1name..outputMname คือรายชื่อพอร์ตส่งออกที่ตรงกับตำแหน่งพารามิเตอร์ส่งออกของฟังก์ชันประมวลผลภาพ

ตัวอย่างการเขียนรายละเอียดของฟังก์ชันประมวลผลภาพตามรูปแบบข้างต้น เช่น ฟังก์ชันหาขอบภาพซึ่งประกาศฟังก์ชันตามหัวข้อที่ 4.5.2 ดังนี้

```
int SobelEdge(PData image,PData &Gx,PData &Gy);
```

จะสามารถเขียนเป็นรูปแบบตามรูปที่ 4.24 ได้ดังนี้

```
"Sobel Edge Detector,SobelEdge,1,image,2,Gx,Gy"
```

5) char *GetFunctionHint (int index) คือฟังก์ชันที่ใช้สำหรับแสดงคำแนะนำในการใช้งานฟังก์ชันประมวลผลภาพแต่ละฟังก์ชันโดยต้องระบุค่า index ซึ่งมีค่าระหว่าง 0 ถึง GetFunctionCount ()-1 ฟังก์ชันนี้ต้องส่งข้อมูลกลับเป็นสายอักขระ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย