บทที่ 3

การออกแบบส่วนประกอบของโปรแกรม

ในบทนี้จะเป็นการประยุกต์แนวคิดและทฤษฎีต่าง ๆ ที่ได้กล่าวไว้ในบทที่ 2 เพื่อสร้างโปรแกรมการค้นคืนสารสนเทศ โดยใช้พื้นฐานแนวคิดของปริภูมิเวกเตอร์เป็นกลไกในการค้นคืน

3.1 การออกแบบฐานข้อมูล

โปรแกรมที่สร้างขึ้นในการศึกษานี้เป็นโปรแกรมฐานข้อมูลแบบหนึ่ง ดังนั้นในการออกแบบจึงต้องคำนึงถึงส่วนประกอบพื้นฐานของระบบฐานข้อมูล

3.1.1 แบบจำลองข้อมูล (Data Models)

ดังได้กล่าวถึงคุณลักษณะของฐานข้อมูลข้อความในบทที่ 2 จะเห็นว่าการสร้างแบบจำลองข้อมูลสำหรับฐานข้อมูลข้อความนั้น ไม่ใช่ปัญหาสำคัญดังเช่นที่พบในฐานข้อมูลทั่วไปทางธุรกิจ ซึ่งการสร้างแบบจำลองนามธรรมเพื่อแทนข้อมูล (abstract models to represent represent the data) เป็นเรื่องที่มีความสำคัญมาก สำหรับฐานข้อมูลข้อความมีลักษณะเป็นแฟ้มลำดับที่เก็บรวบรวมข้อมูลเอกสารไว้ที่เดียวกัน และเพื่อให้สามารถค้นคืนสารสนเทศตามที่ต้องการจากเอกสารได้ ปัญหาของฐานข้อมูลประเภทนี้คือ ปัญหาการวิเคราะห์เนื้อความเพื่อแทนตัวเอกสาร ซึ่งจะเป็นตัวแปรสำคัญที่ใช้ในการค้นคืนสารสนเทศเพื่อให้ได้ตรงกับความต้องการ ในการวิจัยนี้จะสร้างโปรแกรมการค้นคืนสารสนเทศของฐานข้อมูลบรรณานุกรม ระเบียนข้อมูลบรรณานุกรมจะเป็นแฟ้มเอกสารที่สร้างโดยใช้โปรแกรมประมวลคำ ในแต่ละเขตข้อมูลจะมี

บรรทัดบอกเขตข้อมูล โดยต้นบรรทัดเป็นมหัพภาคตามด้วยชื่อเขตข้อมูล บรรทัดต่อ ๆ มาเป็นข้อความของเขตข้อมูลนั้น ๆ ดังตัวอย่างในรูปที่ 3.1

.TI
<การทำแห้ง>ในสภาพ<แช่แข็ง>ของ<กุ้ง>และ<เห็ดหอม>
.AU
ผริรณี ลิขิตวรรณการ
.AB
ได้ศึกษา<การทำแห้ง>ในสภาพ<แช่แข็ง> เพื่อ<เก็บรักษา><กุ้ง>และ<เห็ดหอม> โดยน้ำใน<กุ้ง>และ<เห็ดหอม>จะถูก<แช่แข็ง>แล้วเปลี่ยนสถานะเป็นของแข็ง(น้ำแข็ง) จากนั้นน้ำแข็งจะระเหิดกลายเป็นไอ โดยการควบคุมอุณหภูมิและความดันรอบ ๆ <อาหาร>ตัวอย่าง จากการศึกษาพบว่าอัตราเร็วของการระเหิดจะสูงที่อุณหภูมิสูง และความดันต่ำ

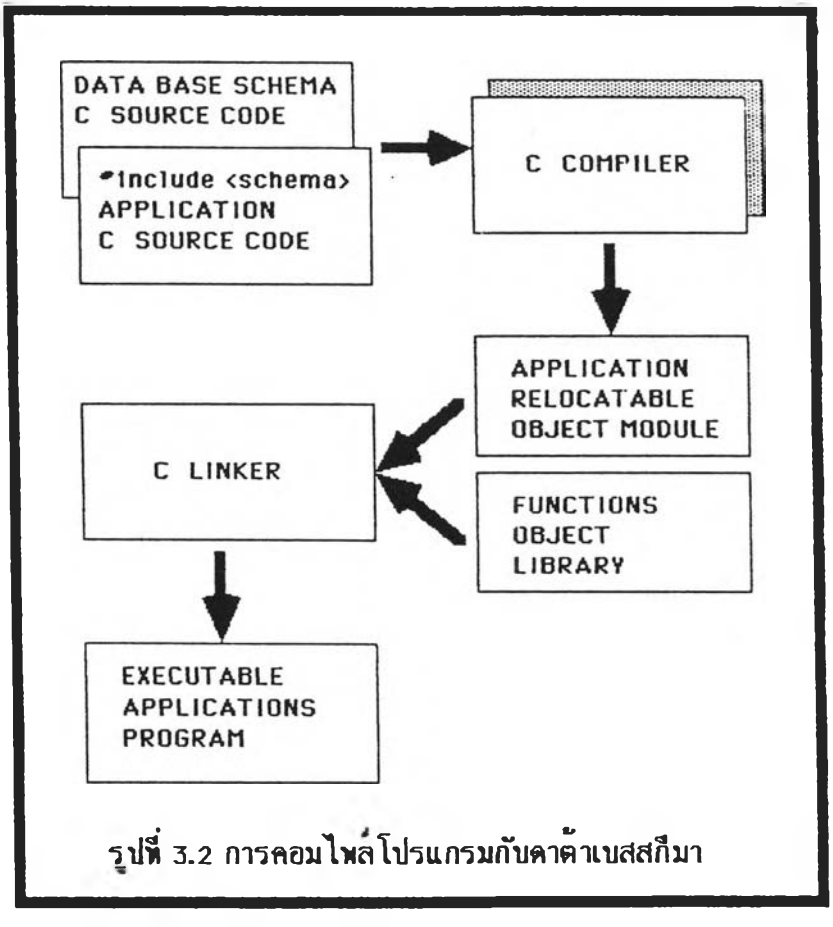
ได้ศึกษาสมบัติทางกายภาพ และองค์ประกอบทางเคมีของ<อาหาร>ตัวอย่างที่<ทำแห้ง>ในสภาพ<แช่แข็ง> โดยเปรียบเทียบกับ<อาหาร>ตัวอย่างที่<ทำแห้ง>โดยใช้<แสงแดด>และ<อาหาร>ตัวอย่างก่อน<ทำแห้ง> พบว่า<อาหาร>ตัวอย่างที่<ทำแห้ง>โดยวิธีนี้ มีคุณภาพใกล้เคียงกับ<อาหาร>ตัวอย่างก่อน<ทำแห้ง>แต่พบว่า <อาหาร>ตัวอย่างที่<ทำแห้ง>โดยใช้<แสงแดด>มีคุณภาพค่อนข้างต่ำ

รูปที่ 3.1 ตัวอย่างแฟ้มเอกสารของฐานข้อมูลบรรณานุกรม

3.1.2 บทนิยามข้อมูล (Data Definition Language)

การออกแบบเพื่อกำหนดเขตข้อมูลต่าง ๆ ของฐานข้อมูล จะใช้ความสามารถ

ในส่วนของพรีโปรเซสเซอร์แมโครในภาษาซี เป็นตัวให้บทนิยามของฐานข้อมูล ซึ่งในส่วนนี้จะเรียกว่า Data base schema C source code ดังนั้นเมื่อต้องการสร้างฐานข้อมูลที่มีโครงสร้างใหม่ก็กำหนดส่วนนี้ขึ้นมาใหม่ แล้วนำไปคอมไพล์ร่วมกับโปรแกรมประยุกต์ต้นฉบับ ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 การคอมไพล์โปรแกรมกับดาต้าเบสสกีมา

ตัวอย่าง Data base schema C source code ของฐานข้อมูลบรรณานุกรมในรูปที่ 3.1 ดังแสดงในรูปที่ 3.3

```
#define TI 0
#define AU 1
#define AB 2

char *full_fieldnames [ ] = {
   "TITLE",
   "AUTHOR",
   "ABSTRACT",
   (char *)0
};
char *short_fieldnames [ ] = {
   ".TI",
   ".AU",
   ".AB",
   (char *)0
};

int field_index [ ] = {
   TI,
   AB,
   -1
};
```

รูปที่ 3.3 DATA BASE SCHEMA C SOURCE CODE

จากรูปที่ 3.3 จะบ่งบอกว่า ฐานข้อมูลบรรณานุกรมนี้ประกอบด้วย 3 เขตข้อมูล คือ TITLE, AUTHOR และ ABSTRACT โดยจะมีบรรทัดแสดงเขตข้อมูลในแฟ้มเอกสารคือ บรรทัด .TI, .AU และ .AB แสดงเขตข้อมูลของ TITLE, AUTHOR และ ABSTRACT ตามลำดับ สำหรับเขตข้อมูลที่จะนำไปทำดรรชนี ซึ่งในที่นี้หมายถึงนำไปสร้างเวกเตอร์ของเอกสาร จะใช้เฉพาะเขตข้อมูล TITLE และ ABSTRACT ซึ่งบ่งบอกโดยอาศัยแถวลำดับของเลขจำนวนเต็มชื่อ field_index

3.1.3 กลไกการค้นคืนสารสนเทศ (Information Retrieval Mechanism)

กลไกที่ใช้ในการสืบค้นข้อมูลในการวิจัยนี้ จะใช้แนวคิดของปริภูมิเวกเตอร์ โดยเอกสารทั้งหมดที่จัดเก็บนั้น จะนำสารสนเทศที่ประกอบในเอกสารมาสร้างเป็นเวกเตอร์ของเอกสาร เพื่อเป็นตัวแทนของเอกสาร สำหรับภาษาสอบถามจะเป็นข้อความที่ประกอบด้วยสารสนเทศที่ต้องการสืบค้น ซึ่งก็จะนำสารสนเทศที่ต้องการมาสร้างเป็นเวกเตอร์ของภาษาสอบถาม ดังนั้นในการค้นคืน ก็จะใช้การคำนวณค่าความใกล้เคียงกันระหว่างเวกเตอร์ของภาษาสอบถามกับเวกเตอร์ของเอกสาร ดังได้กล่าวและยกตัวอย่างไว้ในบทที่ 2

3.2 การออกแบบโครงสร้างข้อมูล

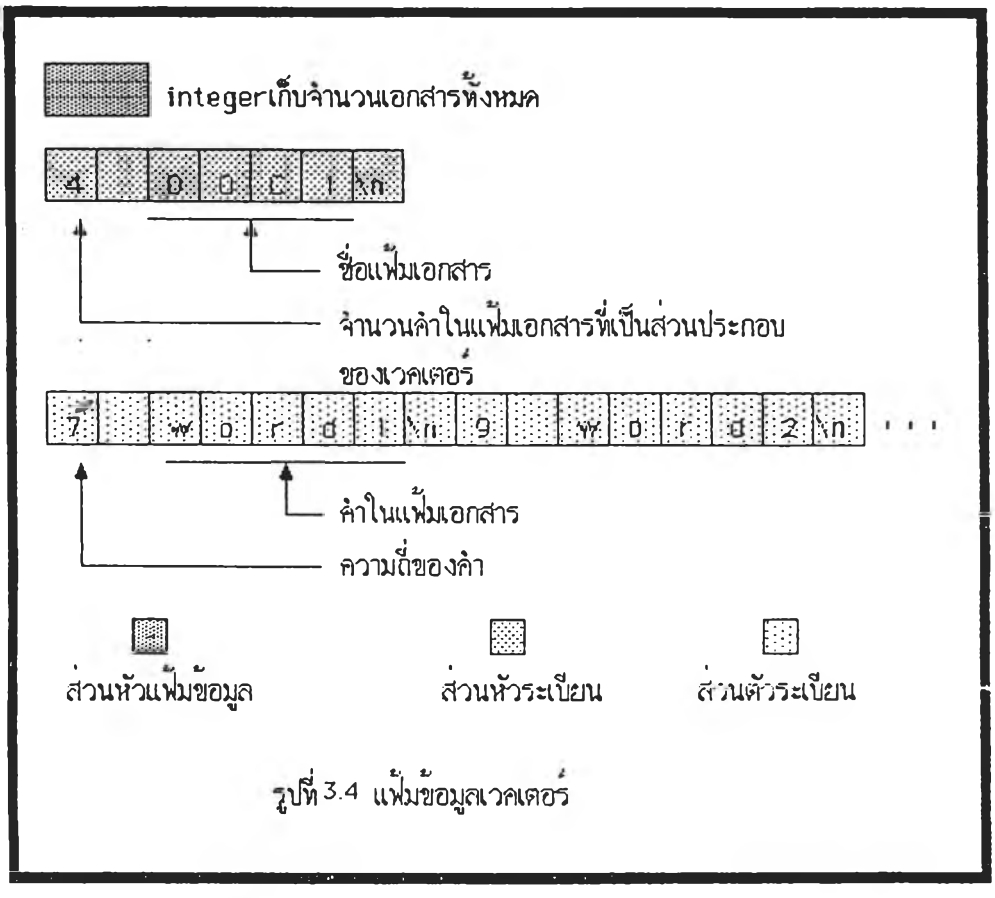
แฟ้มข้อมูลที่ใช้ในการออกแบบโปรแกรมนี้มี 3 ส่วนที่ประกอบกันขึ้นมาคือ

3.2.1 แฟ้มข้อมูลเก็บเอกสาร เป็นแฟ้มที่สร้างขึ้นมาโดยใช้โปรแกรมเครื่องประมวลคำโดยจะมีบรรทัดแสดงเขตข้อมูลของเอกสารตามที่กำหนดไว้ในบทนิยามฐานข้อมูล ตามด้วยรายการข้อมูลของเขตข้อมูลนั้น ๆ ในเขตข้อมูลที่กำหนดให้เป็นดรรชนี จะมีอักขรพิเศษแบ่งแยกคำที่จะนำมาเป็นส่วนประกอบของเวกเตอร์ โดยโปรแกรมจะกวาดตรวจหาคำเหล่านี้ แฟ้มข้อมูลเอกสารจะจัดเก็บหนึ่งแฟ้มข้อมูลต่อหนึ่งเอกสาร ตัวอย่างแฟ้มข้อมูลเอกสารดังรูปที่ 3.1

3.2.2 แฟ้มข้อมูลเวกเตอร์ เป็นแฟ้มข้อมูลที่ใช้เก็บเวกเตอร์ของเอกสารทั้งหมด ในแต่ละระเบียนจะแทนเวกเตอร์ของเอกสารเล่มหนึ่ง ส่วนหัวของแฟ้มข้อมูลจะเก็บจำนวนนับของเวกเตอร์ทั้งหมดที่อยู่ในแฟ้มข้อมูล ซึ่งก็จะเท่ากับจำนวนเอกสารทั้งหมดของฐานข้อมูลในขณะใด ๆ ระเบียนข้อมูลจะมีความยาวแปรได้ โดยมีส่วนประกอบดังนี้

ก. ส่วนหัวระเบียน เป็นข้อมูลตัวเลขของคำศัพท์ทั้งหมดที่ประกอบขึ้นเป็นเวกเตอร์ และชื่อของแฟ้มเอกสารของเวกเตอร์นี้

ข. ส่วนตัวระเบียน ประกอบด้วยความถี่ของคำศัพท์ตามด้วยสายอักขรของคำศัพท์ ในส่วนนี้จะแปรค่าตามตัวเลขของคำศัพท์ที่บอกไว้ในส่วนหัวระเบียน
แฟ้มข้อมูลเวกเตอร์ดังตัวอย่างในรูปที่ 3.4



รูปที่ 3.4 แฟ้มข้อมูลเวคเตอร์

3.2.3 แฟ้มข้อมูลคำสำคัญ คำสำคัญ เป็นคำศัพท์หรือวลีที่สะท้อนถึงสารสนเทศของเอกสาร จึงกำหนดให้เป็นส่วนประกอบของปริภูมิเวกเตอร์ จากสมการที่ 4 ในบทที่ 2 ปริมาณสเกลาร์ของแต่ละเวกเตอร์คำศัพท์จะมีตัวประกอบ 3 ตัว คือ ความถี่ของคำศัพท์ใด ๆ ในเอกสาร (t) ความถี่ของเอกสารที่ใช้คำศัพท์นี้ (d) และจำนวนเอกสาร (N) ดังได้กล่าวมาจะเห็นว่าค่า t และ N เก็บไว้ในแฟ้มข้อมูลเวกเตอร์แล้ว ส่วนค่า d เนื่องจากเป็นค่าที่สามารถเปลี่ยนแปลงได้ตลอดเวลาเมื่อมีเอกสารเพิ่มเข้ามาใหม่ และในทางตรงกันข้ามเมื่อต้องการสร้างเวกเตอร์ของเอกสารขึ้นมาจะต้องสามารถดึงค่า d เพื่อหาค่าปริมาณสเกลาร์ของแต่ละเวกเตอร์ ดังนั้นแฟ้มข้อมูลนี้จะต้องสามารถสนองความต้องการในการเรียกค้นคำศัพท์ให้ได้อย่างรวดเร็ว สำหรับโครงสร้างแฟ้มข้อมูลคำสำคัญที่ใช้ในการวิจัยนี้จะเป็นบีทรี (B-tree)

บีทรีเป็นโครงสร้างข้อมูลที่ถูกพัฒนาขึ้นมาในปี 1970 โดย R. Bayer และ E. McCreight มีคุณสมบัติที่สำคัญคือเป็นทรีที่สมดุล (balanced tree) บีทรีประกอบด้วยรูทโหนด (root node) และโหนดที่มีระดับต่ำกว่าจำนวนตั้งแต่สองโหนดขึ้นไป แต่ละโหนดสามารถกำหนดให้บรรจุคีย์และข่าวสารอื่นได้จำนวนหนึ่งโดยจะจัดเรียงลำดับตามคีย์ ทรีจะมีเพียงรูทโหนดในกรณีเมื่อจำนวนคีย์ทั้งหมดของทรีมีน้อยกว่าหรือเท่ากับจำนวนที่สามารถบรรจุลงในโหนดได้ เมื่อมีการบรรจุข้อมูลเพิ่มลงในโหนดจนเกินกว่าจำนวนที่กำหนด จะมีการสร้างโหนดขึ้นใหม่คู่หนึ่งที่มีระดับต่ำกว่า โดยคีย์ที่อยู่กึ่งกลางจะอยู่บนโหนดเก่าและแบ่งแยกคีย์ด้านบนและล่างลงไปที่โหนดใหม่ทางด้านซ้ายและขวาของโหนดเก่า โหนดที่มีระดับสูงกว่าเรียกว่าพาเรนท์โหนด (parant node) โหนดที่มีระดับต่ำกว่าเรียกไชล์ดโหนด (child node)

ในการวิจัยนี้จะใช้บีทรีจัดเก็บแฟ้มข้อมูลคำศัพท์ลงบนหน่วยความจำสำรอง ซึ่งจะทำให้สามารถจัดเก็บข้อมูลได้มากกว่าบนหน่วยความจำหลัก

3.3 การออกแบบรายละเอียดโปรแกรม

โปรแกรมที่พัฒนาขึ้นเพื่อการวิจัยนี้ ใช้ภาษาซีเป็นเครื่องมือในการพัฒนา เนื่องจากเป็นภาษาที่มีประสิทธิภาพและความยืดหยุ่นสูง สามารถนำไปใช้ได้ในเครื่องคอมพิวเตอร์หลาย ๆ เครื่องโดยไม่ต้องมีการเปลี่ยนแปลงแก้ไข ภาษาซีใช้ฟังก์ชันเป็นหลัก เป็นภาษาของการสร้างคำใหม่จึงเป็นภาษาคอมพิวเตอร์ที่สามารถใช้หลักวิธีการโปรแกรมที่เรียกว่า โปรแกรมโครงสร้าง ดังนั้นการออกแบบระบบโปรแกรมจึงมีรูปแบบแบ่งงานออกเป็นโมดูล โดยคำนึงถึงคุณสมบัติของการออกแบบโมดูลที่ดี คือให้การเชื่อมต่อระหว่างโมดูลมีระดับต่ำ และความยึดแน่นในโมดูลมีระดับสูง

ในการออกแบบโปรแกรมได้เลือกพื้นฐานการจัดบนจอภาพเป็นแบบวินโดว์ และการจัดการโครงสร้างข้อมูลที่สำคัญเป็นบีทรี ดังนั้นจึงได้สร้างฟังก์ชันพื้นฐานเกี่ยวกับวินโดว์ และฟังก์ชันพื้นฐานของบีทรี เพื่อให้สามารถเรียกใช้ร่วมกันในส่วนต่าง ๆ ของระบบโปรแกรม

3.3.1 ฟังก์ชันพื้นฐานของวินโดว์ เป็นฟังก์ชันที่ช่วยจัดการ เกี่ยวกับการดำเนินการเมื่อมีการกำหนดขอบเขตของวินโดว์ โครงสร้างข้อมูลในภาษาซีที่ใช้เป็นหลักในการกำหนดวินโดว์คือ

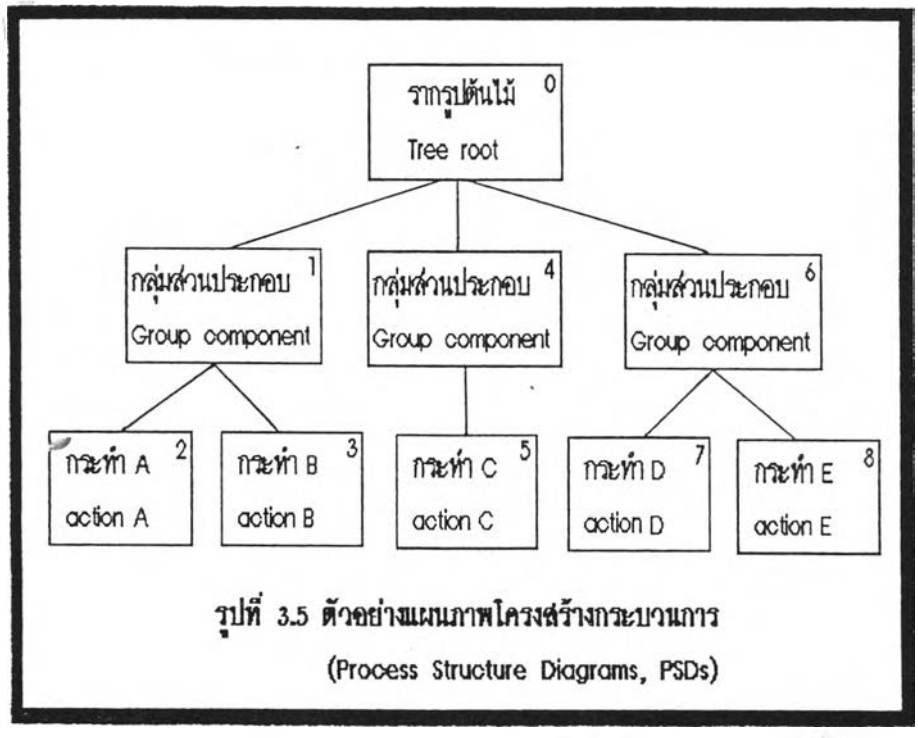
```
struct window_frame {
        int startx, endx, starty, endy;
        int curx, cury;
        unsinged char *p;
        char *header;
        int border, active;
} frame [MAX_FRAME];
```

จำนวนวินโดว์สามารถกำหนดได้โดยอาศัยค่าคงที่ MAX_FRAME สำหรับฟังก์ชันพื้นฐานของวินโดว์ที่สร้างขึ้นในการวิจัยนี้มีรวมทั้งสิ้น 24 ฟังก์ชัน รายละเอียดและหน้าที่ของแต่ละฟังก์ชันได้รวบรวมนำเสนอไว้ในภาคผนวก ก.

3.3.2 ฟังก์ชันพื้นฐานของบีทรี เป็นฟังก์ชันที่ช่วยจัดการเกี่ยวกับการดำเนินการพื้นฐานต่าง ๆ ของโครงสร้างข้อมูลบีทรีบนดิสก์ เช่น การจับจองโหนดบนดิสก์ การอ่านและเขียนข้อมูลของโหนดบนดิสก์ การสอดแทรก การค้นหา การเพิ่ม และการลบข้อมูลบนบีทรี เป็นต้น ฟังก์ชันพื้นฐานของบีทรีมีทั้งสิ้น 12 ฟังก์ชัน รายละเอียดและหน้าที่ของแต่ละฟังก์ชันได้รวบรวมนำเสนอไว้ในภาคผนวก ข.

3.3.3 โครงสร้างของโปรแกรม การพัฒนาโปรแกรมได้ยึดหลักวิธีการโปรแกรมแบบโปรแกรมโครงสร้าง และการออกแบบจากบนลงล่าง (top down design) ซึ่งสามารถแบ่งงานออกเป็นโมดูล โครงสร้างของโปรแกรมที่พัฒนาขึ้นในงานวิจัยนี้ จะนำเสนอโดยอาศัยแผนภาพโครงสร้างกระบวนการ (process structure diagrams) ซึ่งมีลักษณะเป็นรูปต้นไม้ (tree) หรือแผนภาพเชิงลำดับชั้น (tree hierarchy diagrams) ดังรูปที่ 3.5

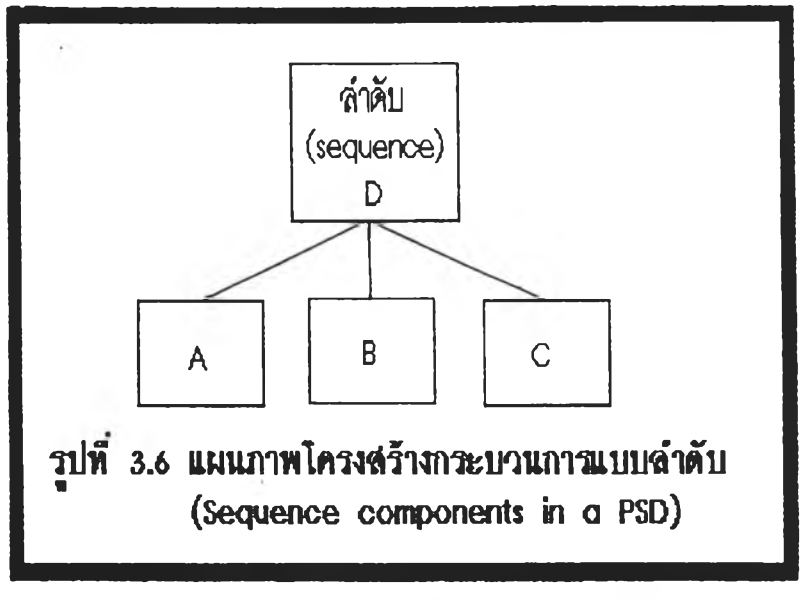
ii7149663



รูปที่ 3.5 ตัวอย่างแผนภาพโครงสร้างกระบวนการ
(Process Structure Diagrams, PSDs)

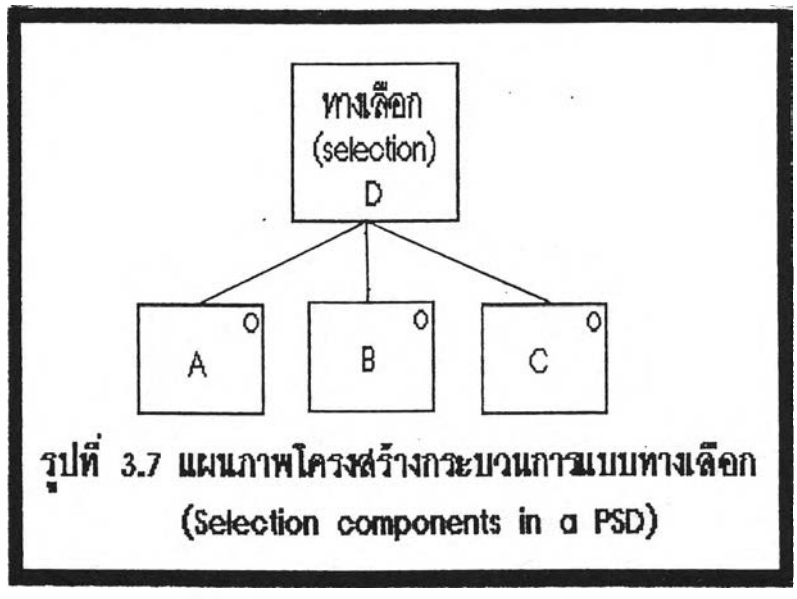
จุดแตกใบ (leaf node) ที่อยู่ตอนล่างสุดของรูปต้นไม้เรียกว่าส่วนประกอบพื้นฐาน (elementary components) จะเป็นส่วนที่แสดงการกระทำ ส่วนจุดแตกกิ่งระดับพ่อแม่ที่อยู่เหนือขึ้นมาจะเป็นการรวมกลุ่มของส่วนประกอบพื้นฐานเข้าด้วยกัน เรียกว่าส่วนประกอบกลุ่ม (group components) ลำดับการอ่านของแผนภาพ จะพิจารณาจากบนลงล่าง และซ้ายไปขวา (ตัวอย่างในรูปที่ 3.5 จะอ่านตามลำดับตัวเลขที่เขียนไว้มุมบนขวาของบล็อก) แผนภาพโครงสร้างกระบวนการนี้จะมี 3 ประเภท คือ

ก. แบบลำดับ (sequence) แสดงกระบวนการกระทำเป็นลำดับจากซ้ายไปขวา ดังรูปที่ 3.6



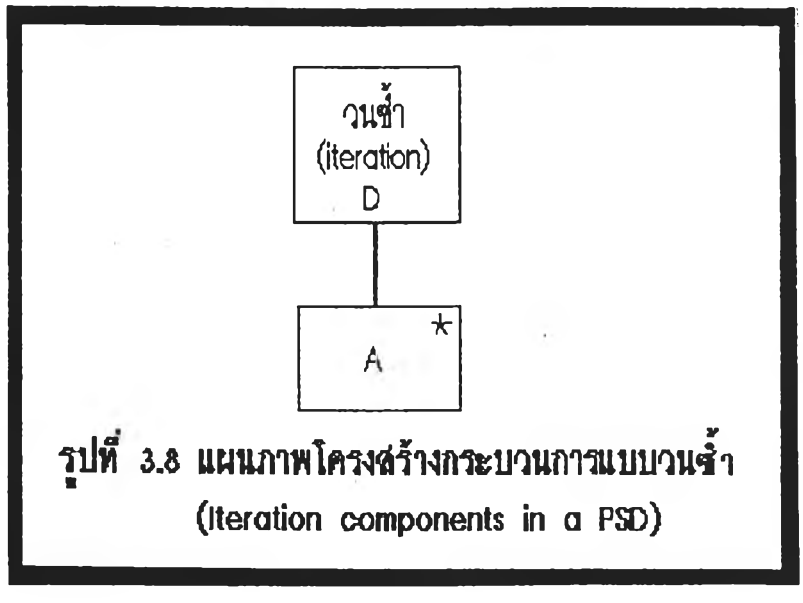
รูปที่ 3.6 แผนภาพโครงสร้างกระบวนการแบบลำดับ
(Sequence components in a PSD)

ข. แบบทางเลือก (selection) แสดงกระบวนการที่เลือกกระทำเพียงตัวใดตัวหนึ่ง ดังรูปที่ 3.7



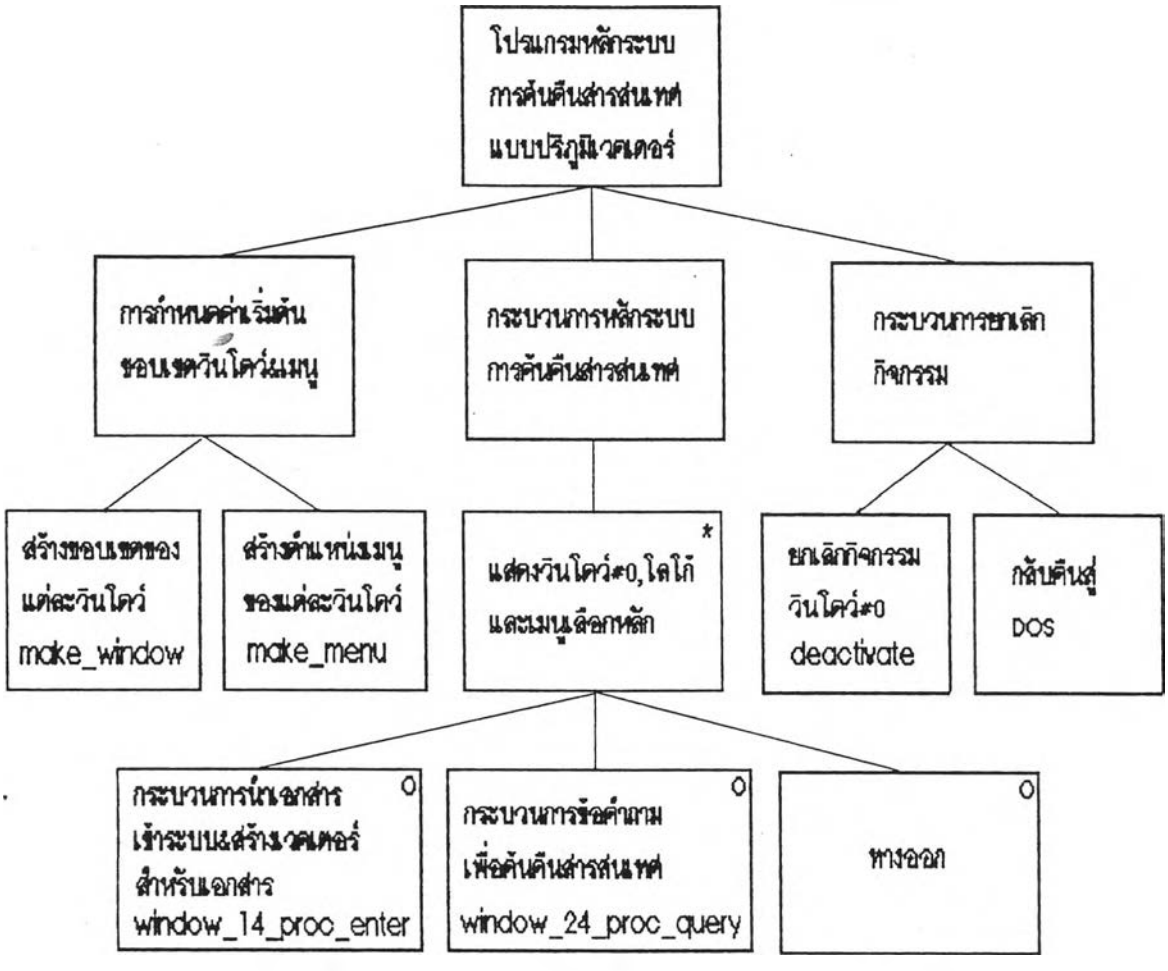
รูปที่ 3.7 แผนภาพโครงสร้างกระบวนการแบบทางเลือก
(Selection components in a PSD)

ค. แบบวนซ้ำ (iteration) แสดงกระบวนการที่มีการกระทำวนซ้ำ โดยเงื่อนไขที่ใช้กำหนดการวนซ้ำจะอยู่บนจุดแตกกิ่งระดับพ่อแม่ ดังรูปที่ 3.8



รูปที่ 3.8 แผนภาพโครงสร้างกระบวนการแบบวนซ้ำ
(Iteration components in a PSD)

โครงสร้างหลักของโปรแกรมที่พัฒนาขึ้นในงานวิจัยนี้ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 แผนภาพโครงสร้างโปรแกรมหลักแบบลำดับชั้น
(Hierarchy Main Program Structure Diagrams)

จากรูปที่ 3.9 เป็นแผนภาพโครงสร้างกระบวนการเชิงลำดับชั้นของโปรแกรมหลักในงานวิจัยนี้ ซึ่งประกอบด้วยจุดแตกกิ่งระดับพ่อแม่ ที่เป็นส่วนประกอบกลุ่ม (group components) 3 กลุ่ม คือ

3.3.3.1 กลุ่มการกำหนดค่าเริ่มต้นขอบเขตของวินโดว์และเมนู เป็นส่วนที่ทำหน้าที่ กำหนดกรอบตำแหน่งวินโดว์ที่ใช้ในโปรแกรมนี้ทั้งหมด รวมทั้งข้อความทางเลือกต่าง ๆ ที่จะปรากฏในแต่ละวินโดว์ จุดแตกใบ (leaf node) ที่แสดงการกระทำของส่วนประกอบกลุ่มนี้ คือ ฟังก์ชัน make_window และ make_menu เป็นฟังก์ชันของวินโดว์ (ดูหน้าที่ของฟังก์ชันนี้ได้ในภาคผนวก ก.)

3.3.3.2 กลุ่มกระบวนการหลักระบบการค้นคืนสารสนเทศ เป็นส่วนที่จะแสดงโลโก้ และกรอบวินโดว์ทางเลือกหลัก ซึ่งจะมี 3 ทางเลือก คือ

3.3.3.2.1 กระบวนการนำเอกสารเข้าระบบและสร้างเวกเตอร์ของเอกสาร (window_14_proc_enter) ดังแสดงในรูปที่ 3.10 จุดแตกใบที่แสดงการกระทำประกอบด้วยฟังก์ชัน ดังนี้

add1doc เป็นกระบวนการแบ่งแยกคำดรรชนี จากแฟ้มเอกสารที่จะนำเข้าระบบ และสร้างรายการโยงของคำดรรชนี (ผังงานดังแสดงในรูปที่ 3.10.1)

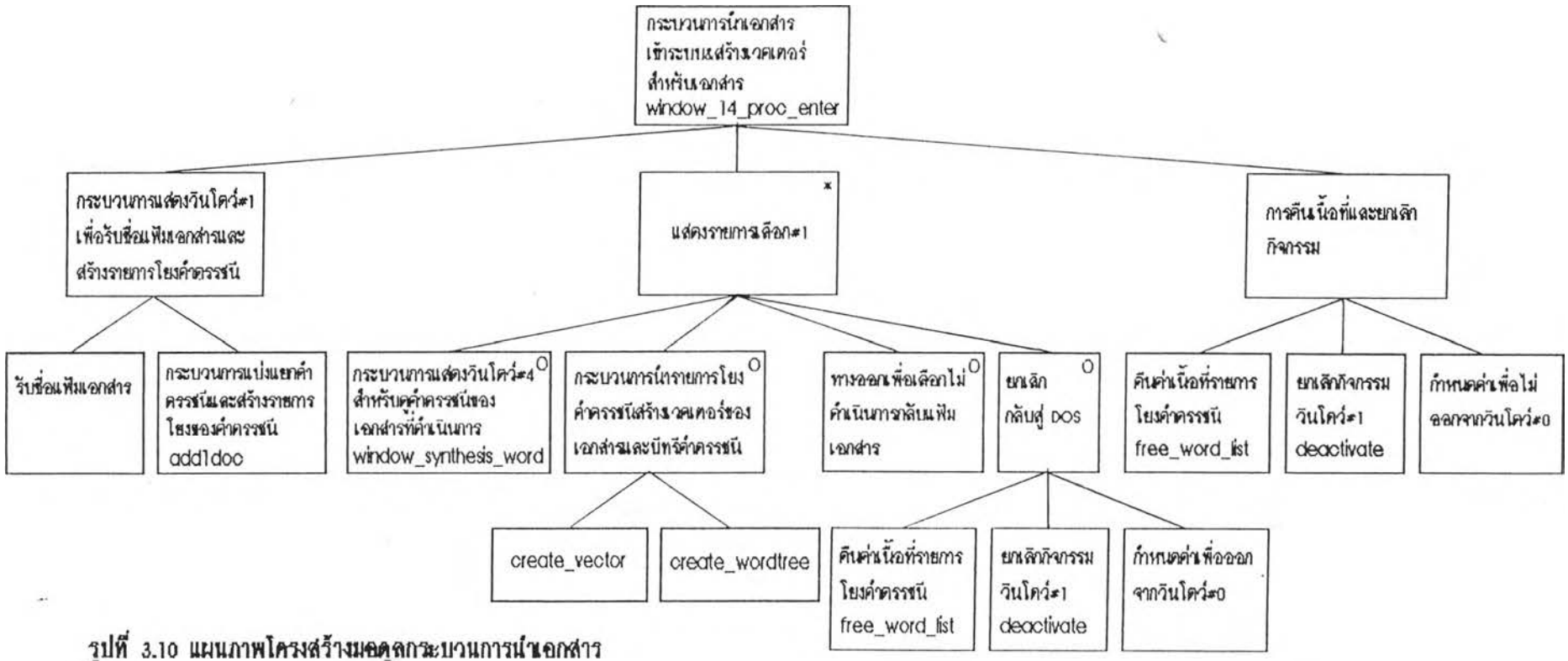
window_synthesis_word เป็นกระบวนการแสดงวินโดว์สำหรับดูคำดรรชนีจากรายการโยงคำดรรชนี (ผังงานดังแสดงในรูปที่ 3.10.2)

create_vector เป็นกระบวนการนำรายการโยงคำดรรชนีของเอกสารมาสร้างเวกเตอร์ให้กับเอกสาร (ผังงานดังแสดงในรูปที่ 3.10.3)

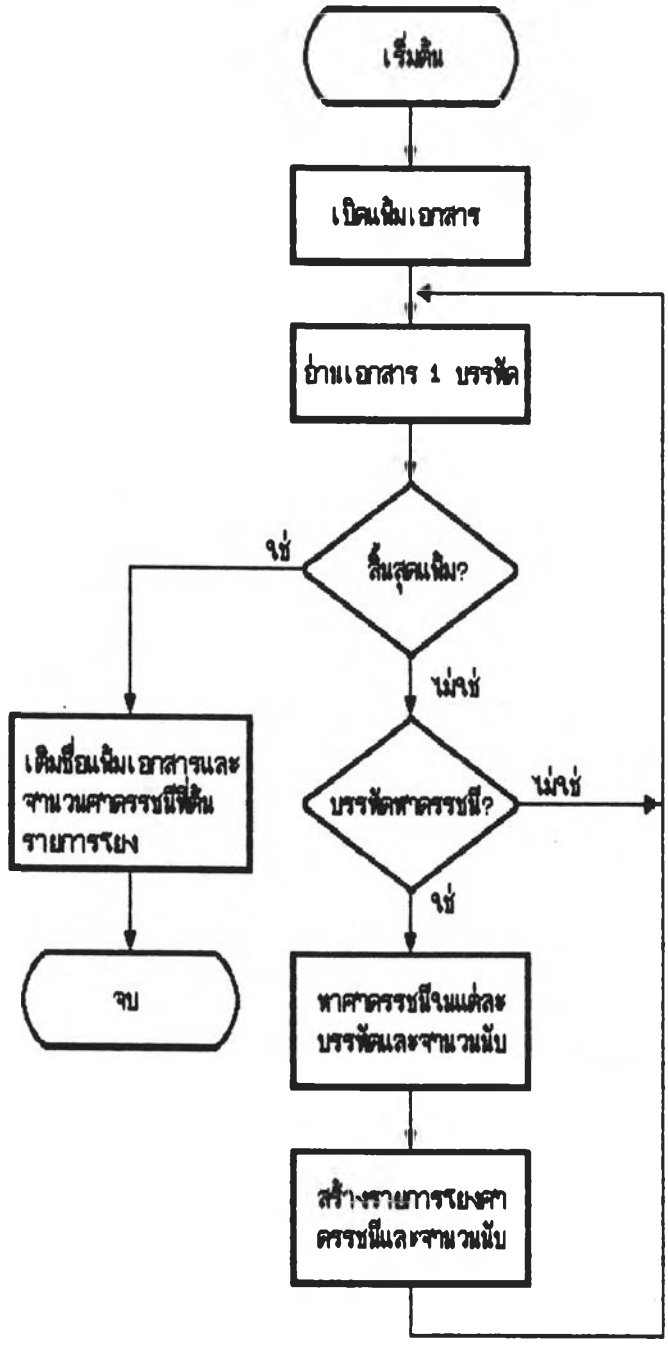
create_wordtree เป็นกระบวนการนำรายการโยงคำดรรชนีของเอกสารใส่ลงในบีทรี (ผังงานดังแสดงในรูปที่ 3.10.4)

free_word_list เป็นกระบวนการคืนค่าเนื้อที่หน่วยความจำ ที่จองไว้ตอนสร้างรายการโยงคำดรรชนีของเอกสาร (ผังงานดังแสดงในรูปที่ 3.10.5)

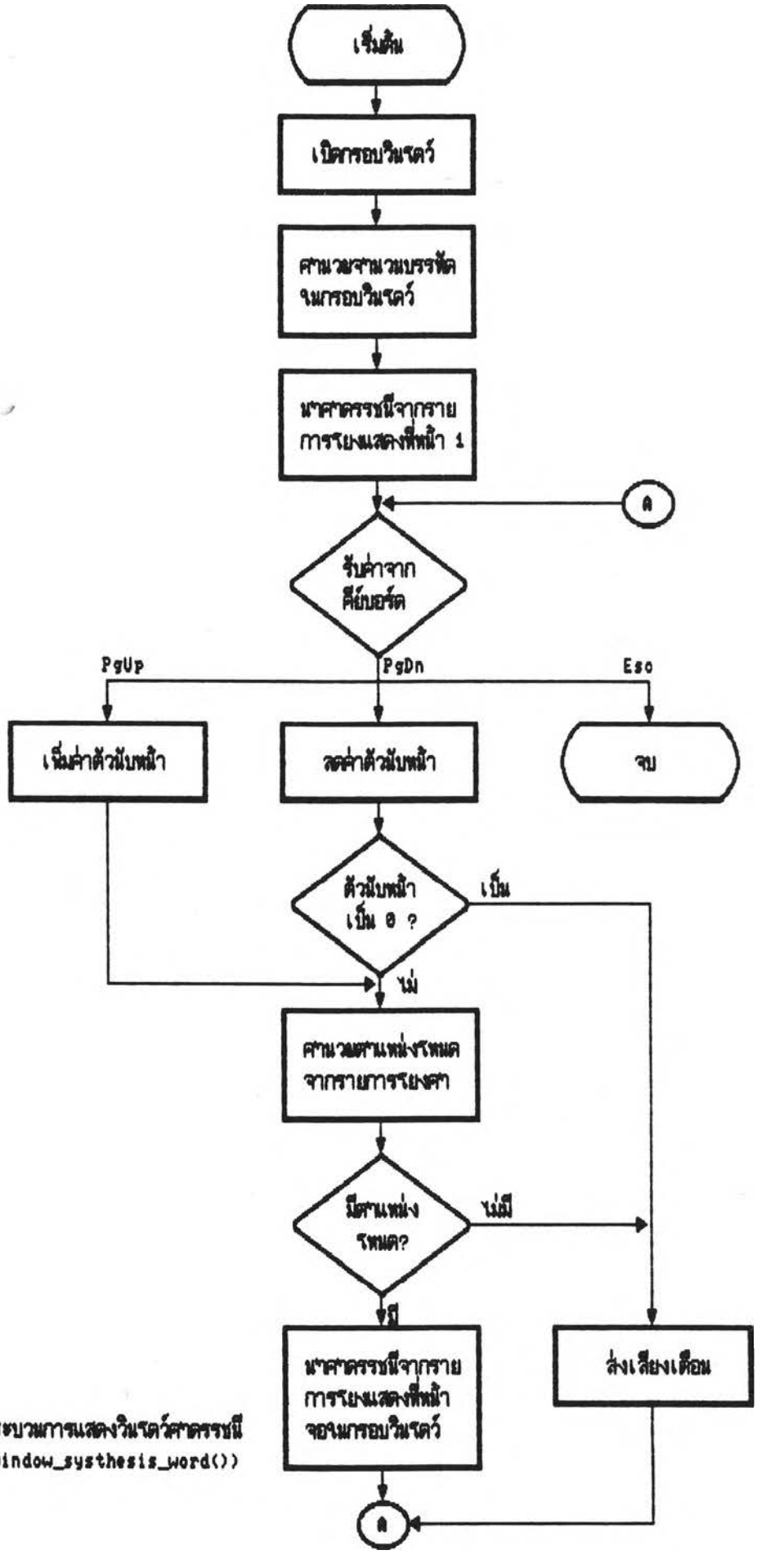
deactivate เป็นกระบวนการยกเลิกการใช้วินโดว์ เป็นฟังก์ชันของวินโดว์ (ดูหน้าที่ของฟังก์ชันนี้ได้ในภาคผนวก ก.)



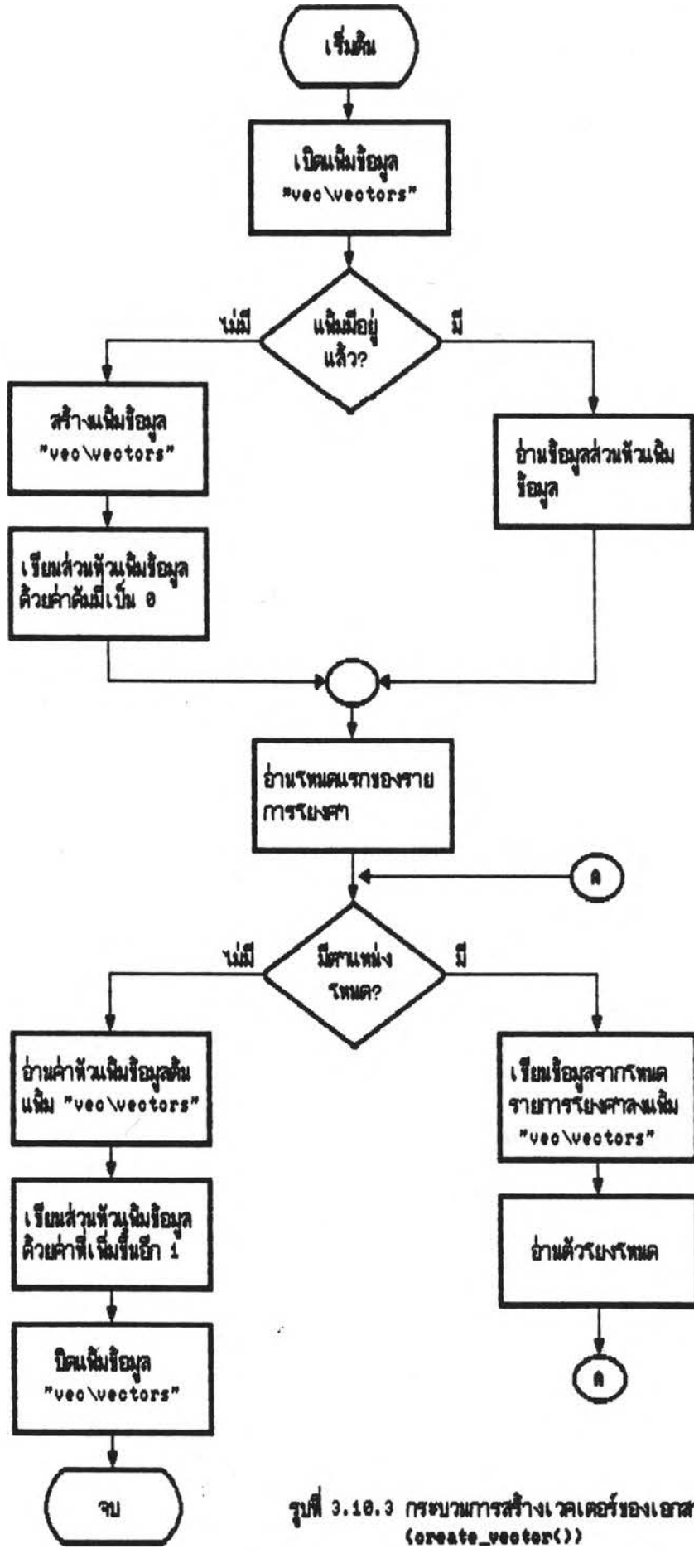
รูปที่ 3.10 แผนภาพโครงสร้างมอดูลกระบวนการนำเอกสารเข้าระบบและสร้างเวคเตอร์สำหรับเอกสาร (Module window_14_proc_enter)



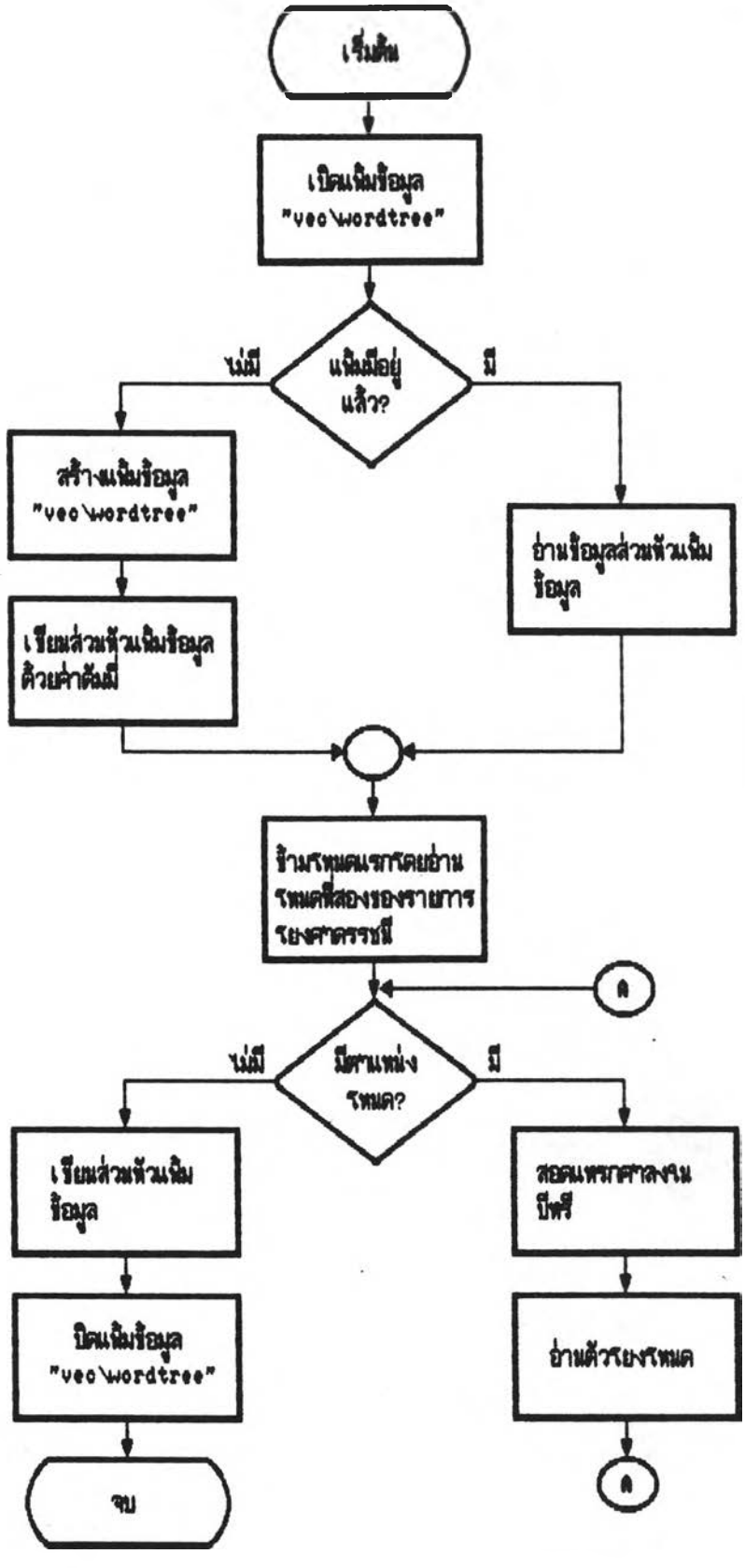
รูปที่ 3.10.1 กระบวนการสร้างรายการย่อยศัพท์ของเอกสาร (addidoc())



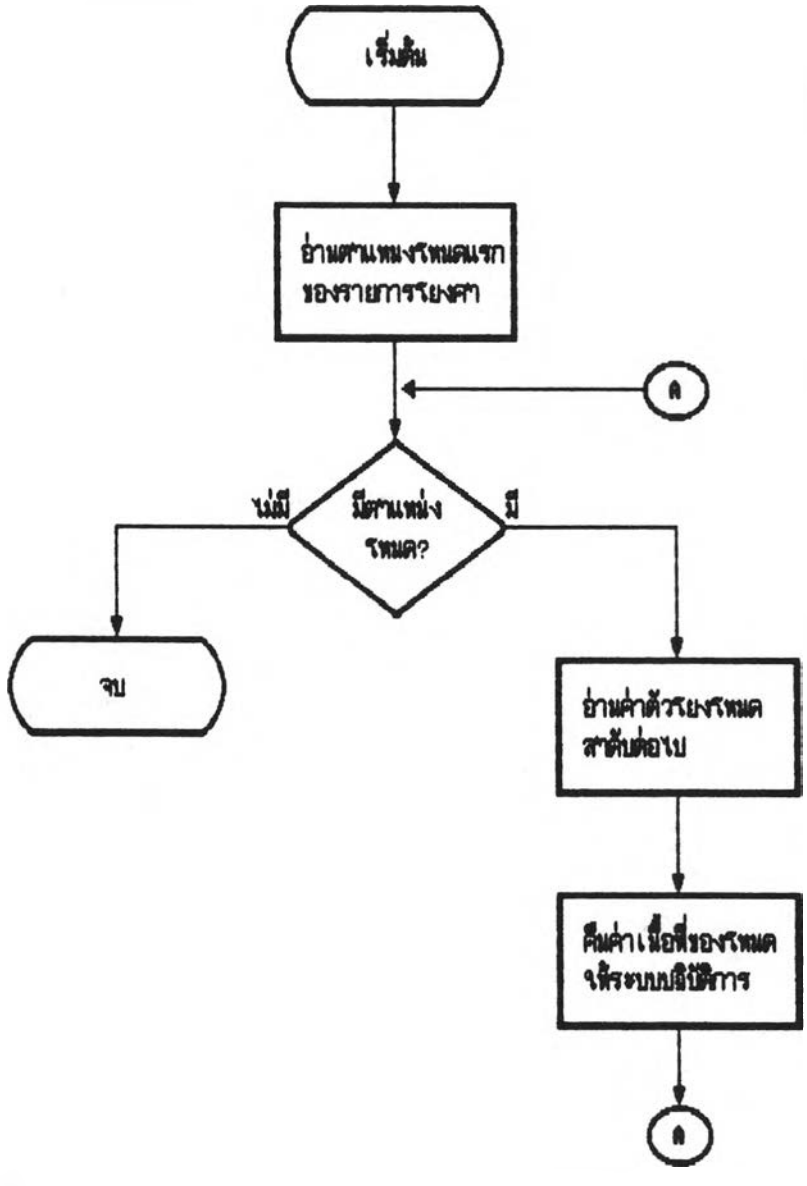
รูปที่ 3.10.2 กระบวนการแสดงวินโดว์ค่าตรรชนี
(window_systhesis_word())



รูปที่ 3.10.3 กระบวนการสร้างเวคเตอร์ของเอกสาร (create_vector())



รูปที่ 3.10.4 กระบวนการนำรายการวิเคราะห์ลงแฟ้มข้อมูล
(create_wordtree())



รูปที่ 3.10.5 กระบวนการคืนค่าเนื้อที่รายการรายงานตรรชี
(free_word_list())

3.3.3.2.2 กระบวนการข้อคำถามเพื่อค้นคืนสารสนเทศ (window_24_proc_query) ดังแสดงในรูปที่ 3.11 จุดแตกใบที่แสดงการกระทำประกอบด้วยฟังก์ชัน ดังนี้

window_notepad เป็นกระบวนการรับข้อมูลอย่างง่าย ภายในกรอบวินโดว์ และในกรณีข้อมูลภาษาไทย จะแสดงจุดตัดคำให้ทราบ โดยตัวอักษรที่จุดตัดคำได้จะแสดงเป็นตัวสว่าง (ผังงานดังแสดงในรูปที่ 3.11.1)

window_seperate_word เป็นกระบวนการแบ่งแยกคำ ของข้อมูลภายในวินโดว์ที่ใช้ฟังก์ชัน window_notepad รับข้อมูล พร้อมทั้งสร้างรายการโยงคำดรรชนี (ผังงานดังแสดงในรูปที่ 3.11.2)

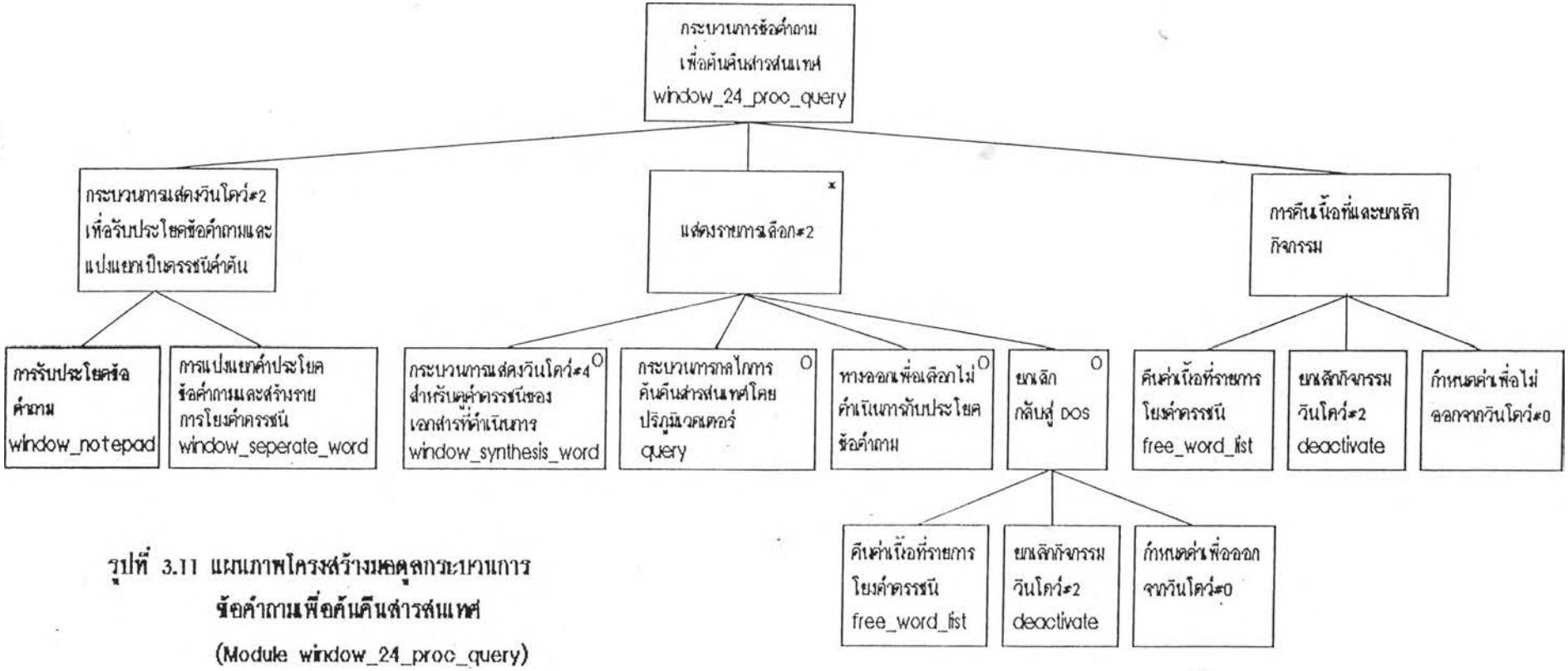
window_systhesis_word เป็นกระบวนการแสดงวินโดว์สำหรับดูคำดรรชนีจากรายการโยงคำดรรชนี (ผังงานดังแสดงในรูปที่ 3.10.2)

query เป็นกระบวนการกลไกหลัก การค้นคืนสารสนเทศโดยปริภูมิเวกเตอร์ (ผังงานดังแสดงในรูปที่ 3.11.3)

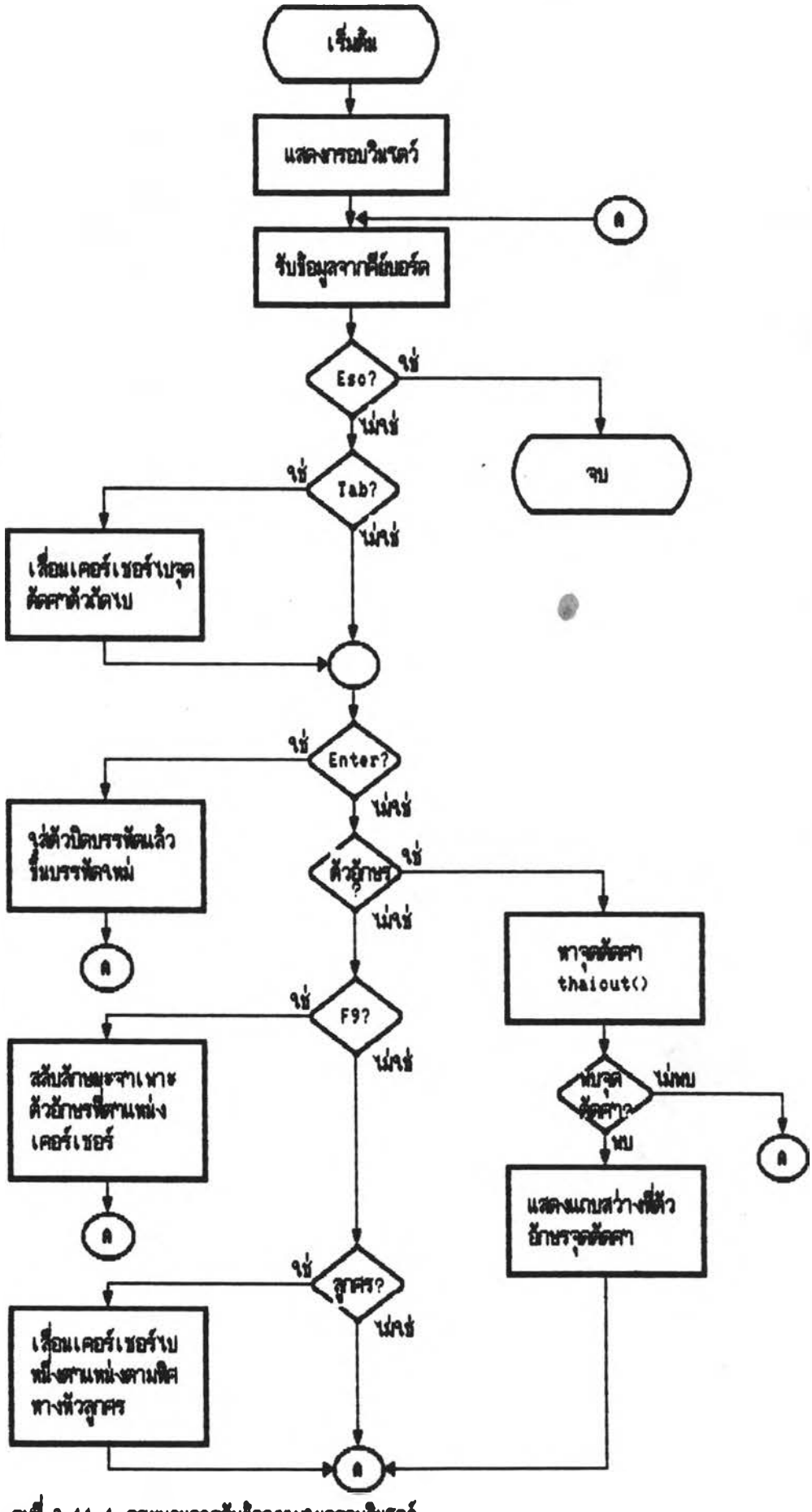
free_word_list เป็นกระบวนการคืนค่าเนื้อที่หน่วยความจำ ที่จองไว้ตอนสร้างรายการโยงคำดรรชนีของเอกสาร (ผังงานดังแสดงในรูปที่ 3.10.5)

deactivate เป็นกระบวนการยกเลิกการใช้วินโดว์ เป็นฟังก์ชันของวินโดว์ (ดูหน้าที่ของฟังก์ชันนี้ได้ในภาคผนวก ก.)

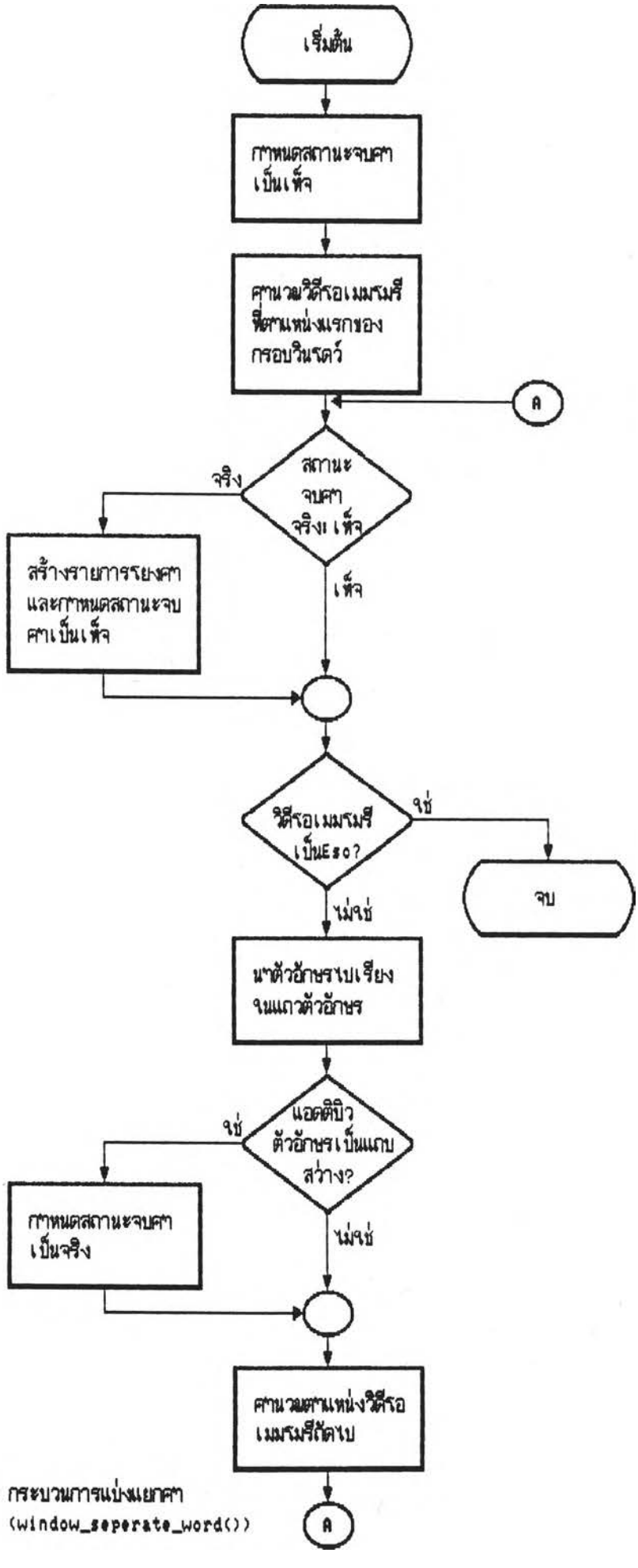
3.3.3.2.3 ทางออก เป็นทางเลือกเพื่อออกจากกิจกรรมหลักของโปรแกรมนี้



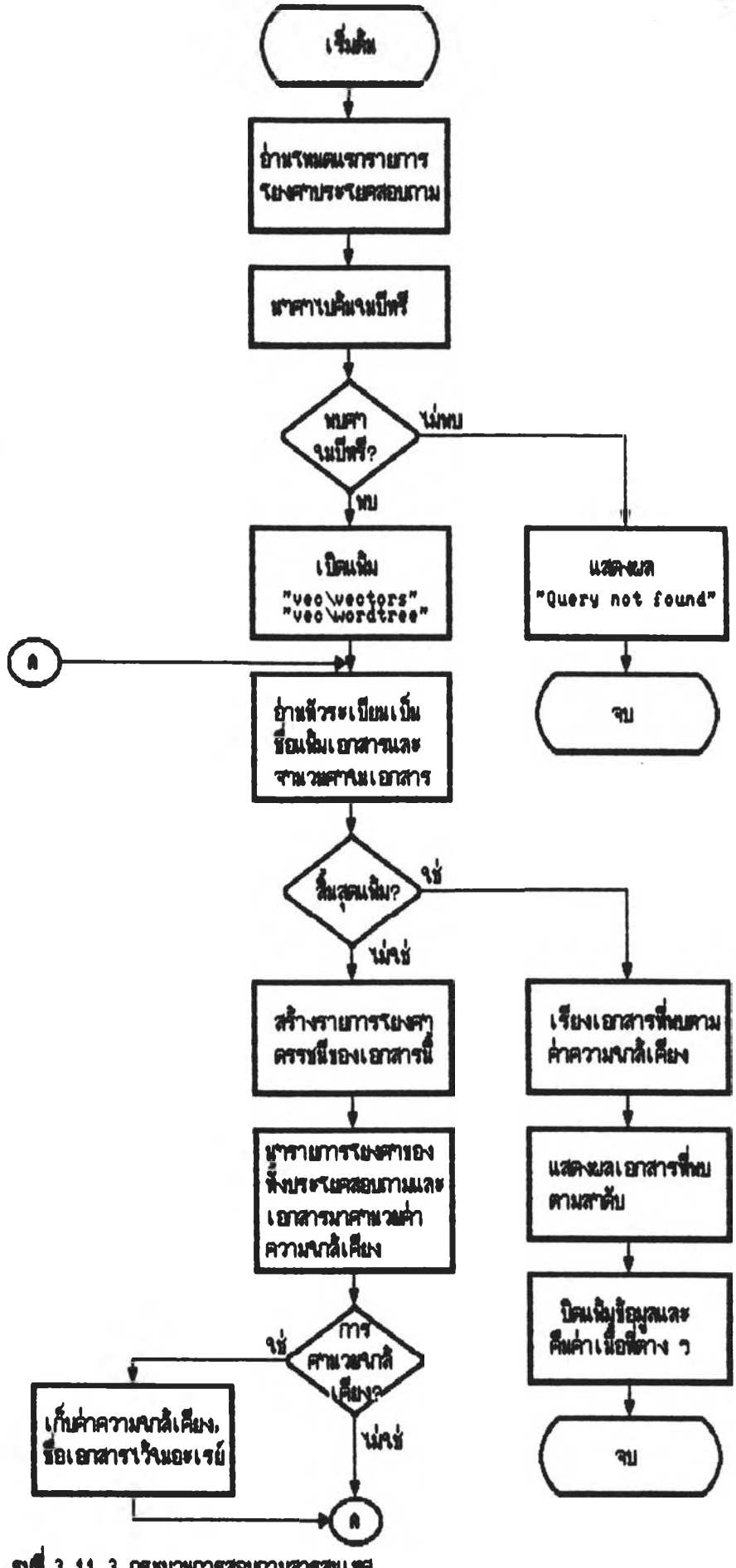
รูปที่ 3.11 แผนภาพโครงสร้างมอดูลกระบวนการข้อคำถามเพื่อค้นคืนสารสนเทศ
(Module window_24_proc_query)



รูปที่ 3.11.1 กระบวนการรับข้อความของกรอบวินโดว์ (window_notepad())



รูปที่ 3.11.2 กระบวนการแบ่งแยกคำ (window_seperate_word())



รูปที่ 3.11.3 กระบวนการสอบถามสารสนเทศ (query())

3.3.3.3 กลุ่มกระบวนการยกเลิกกิจกรรม เป็นส่วนที่จะยกเลิกการใช้วินโดว์ และทำการกลับคืนสู่ระบบปฏิบัติการ (จากรูปที่ 3.9)