

ในการออกแบบตัวเชื่อมโยงมอดูลการตัดคำและขั้นตอนวิธีการแทรกอักขระแบ่งคำ สำหรับวิทยานิพนธ์ฉบับนี้ เพื่อให้สามารถนำไปใช้งานได้จริง จึงได้ทำการพัฒนาส่วนที่เป็นฟังก์ชันอรรถประโยชน์ (utility functions) ขึ้นมา โดยใช้ภาษาซีเป็นเครื่องมือพัฒนา เนื่องจากเป็นภาษาที่มีประสิทธิภาพและความยืดหยุ่นสูง สามารถนำไปใช้ได้บนเครื่องคอมพิวเตอร์หลายๆเครื่อง โดยไม่ต้องมีการเปลี่ยนแปลงแก้ไข

### 6.1 การออกแบบฟังก์ชันอรรถประโยชน์

การออกแบบฟังก์ชันอรรถประโยชน์จะต้องคำนึงถึงว่า ควรจะประกอบด้วยฟังก์ชันอะไรบ้าง เพื่อให้ผู้ใช้สามารถเรียกใช้งานได้อย่างสะดวกและครอบคลุมฟังก์ชันที่ควรจะมีทั้งหมดของมอดูลการตัดคำ

ในการออกแบบนี้ได้ทำการออกแบบโดยมองส่วนประกอบต่างๆที่เกี่ยวข้องกับมอดูลการตัดคำในลักษณะของ "วัตถุ" (object) โดยที่วัตถุแต่ละชิ้นสามารถที่จะเปลี่ยนแปลงไปอยู่ในรูปของวัตถุอื่นได้ เราสามารถแบ่งวัตถุออกได้เป็น 4 รูปแบบ คือ

1. ข้อความเต็ม หมายถึง ข้อความก่อนที่จะถูกทำการตัดคำ
2. ข้อความที่ถูกตัดคำแล้ว หมายถึง ข้อความเต็มที่ผ่านมอดูลการตัดคำและมีการแทรกอักขระแบ่งคำแล้ว
3. จุดตัดคำ หมายถึง ตัวชี้ (pointer) ซึ่งชี้ไปยังตำแหน่งของการตัดคำ
4. อักขระแบ่งคำ หมายถึง อักขระที่ใช้แทรกระหว่างคำแต่ละคำ

การที่วัตถุแต่ละชนิดจะแปลงไปเป็นวัตถุชนิดอื่นนั้น จะต้องมี "การกระทำ" (action) มากกระทำกับวัตถุนั้นเพื่อให้วัตถุเปลี่ยนรูปไป สำหรับการกระทำนี้เราสามารถแบ่งได้เป็น 3 ชนิดคือ

1. การตัดคำ เป็นการกระทำเพื่อหาจุดตัดคำจากข้อความเต็ม
2. การแทรกอักขระแบ่งคำ เป็นการกระทำเพื่อแทรกอักขระแบ่งคำลงในข้อความเต็ม เพื่อสร้างข้อความที่ถูกตัดคำแล้ว
3. การตรวจรู้อักขระแบ่งคำ เป็นการหาตำแหน่งจุดตัดคำจากข้อความที่ถูกตัดคำแล้ว

ความสัมพันธ์ของวัตถุและการกระทำทั้งหมดจะแสดงได้ดังในรูปที่ 6.1 ก และ ข

ในรูปที่ 6.1 ก วัตถุข้อความเต็มถูกเปลี่ยนเป็นวัตถุจุดตัดคำด้วยการกระทำ การตัดคำ และการกระทำ การแทรกอักขระแบ่งคำ จะรวม ข้อความเต็ม, จุดตัดคำ และ อักขระแบ่งคำ เพื่อสร้างข้อความที่ถูกตัดคำแล้ว

ส่วนในรูปที่ 6.1 ข การกระทำ การตรวจรู้อักขระแบ่งคำ จะนำ ข้อความที่ถูกตัดคำแล้ว มาแยกออก จะได้ข้อความเต็ม, จุดตัดคำและอักขระแบ่งคำ กลับคืนมา

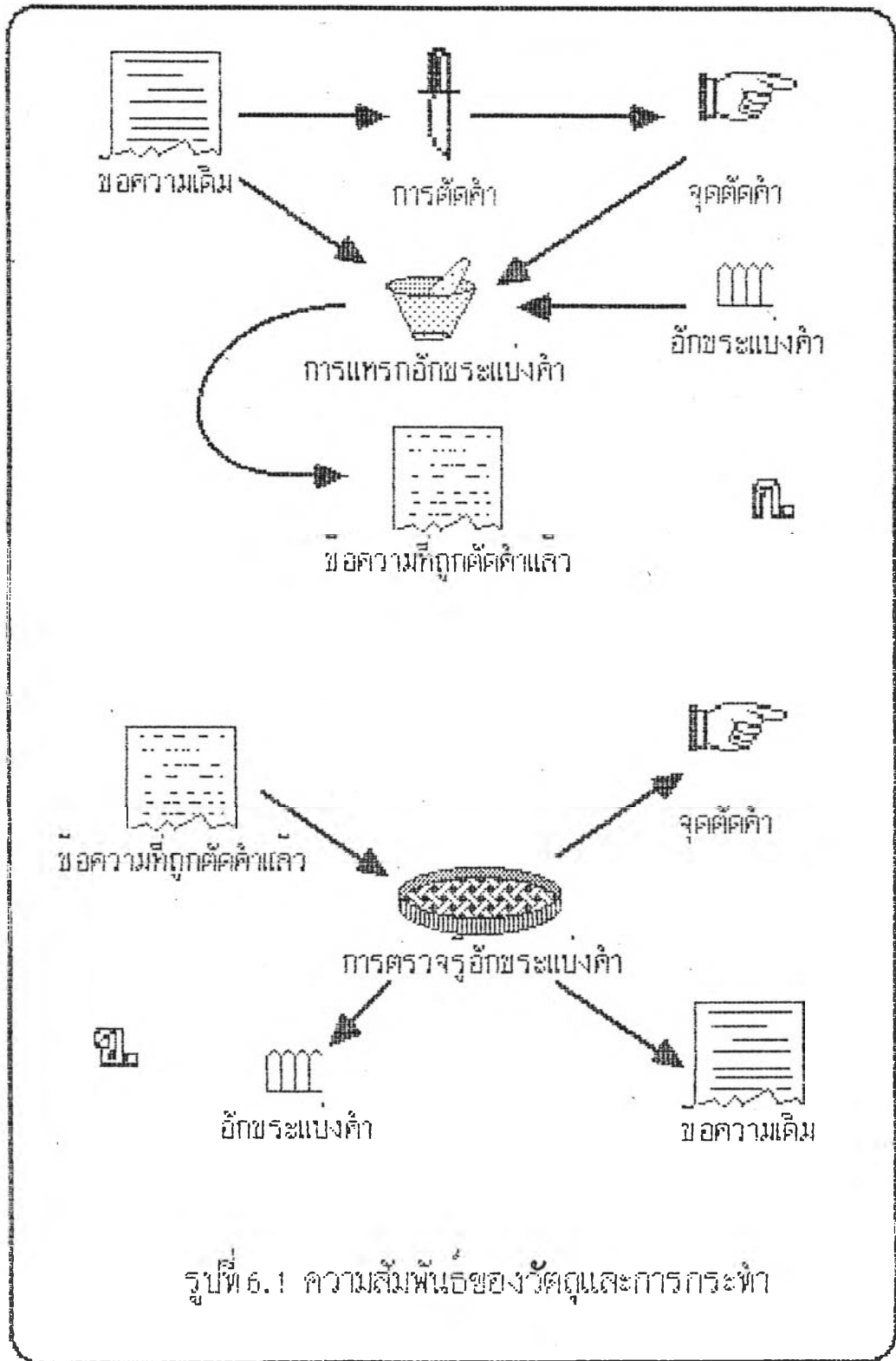
## 6.2 ฟังก์ชันพื้นฐาน

ในการจะสร้างฟังก์ชันอรรถประโยชน์ได้นั้นจำเป็นต้องมีฟังก์ชันพื้นฐานเพื่อรองรับการทำงานในระดับพื้นฐาน คือ การตัดคำ, การแทรกอักขระแบ่งคำ และการตรวจรู้อักขระแบ่งคำ ฟังก์ชันพื้นฐานเหล่านี้จะพัฒนาด้วยภาษาซี และพัฒนาขึ้นจากขั้นตอนวิธีซึ่งได้ออกแบบไว้ ฟังก์ชันเหล่านี้ ได้แก่

thaiCut - พัฒนาจากขั้นตอนวิธีการตัดคำ โดยยึดถือตัวเชื่อมโยงดังที่ได้  
ออกแบบไว้ในบทที่ 4

putSep - พัฒนาจากขั้นตอนวิธีการแทรกอักขระแบ่งคำ ในบทที่ 5

getSep - พัฒนาจากขั้นตอนวิธีการตรวจรู้อักขระแบ่งคำ ในบทที่ 5



6.3 ฟังก์ชันอรรถประโยชน์

ฟังก์ชันอรรถประโยชน์จะสร้างขึ้นบนฟังก์ชันพื้นฐานอีกทีหนึ่ง โดยจะประกอบด้วยฟังก์ชัน 4 ฟังก์ชัน คือ

1. putcut- เป็นฟังก์ชันที่ทำงานในส่วนที่สองของรูปที่ 6.1 ก คือเป็นการรวมข้อความเดิมกับจุดตัดคำและอักขระแบ่งคำ และสร้างข้อความ



ที่ถูกตัดคำแล้ว ฟังก์ชัน `putcut` นี้ จะสร้างบนฟังก์ชันพื้นฐาน `putsep`

2. `getcut`- ทำงานตามรูปที่ 6.2 ข คือวิเคราะห์และแยกข้อความที่ถูกตัดแล้ว ออกเป็นข้อความเต็ม, จุดตัดคำ และอักษรแบ่งคำ ฟังก์ชัน `getcut` จะสร้างบนฟังก์ชันพื้นฐาน `getsep`
3. `cputcut`- ครอบคลุมฟังก์ชันตามรูปที่ 6.1 ก ทั้งหมด คือนำข้อความเต็ม มาหาจุดตัดคำ และสร้างข้อความที่ถูกตัดคำแล้วจากข้อความเต็ม, จุดตัดคำและอักษรแบ่งคำ ฟังก์ชัน `cputcut` จะสร้างบนฟังก์ชันพื้นฐาน `thaicut` และ `putsep`
4. `cgetcut`- ทำงานในส่วนแรกของรูปที่ 6.1 ก คือ หาจุดตัดคำจากข้อความเต็ม ฟังก์ชัน `cgetcut` จะสร้างบนฟังก์ชันพื้นฐาน `thaicut`

#### 6.4 การใช้ฟังก์ชันอรรถประโยชน์

ในหัวข้อนี้จะได้กล่าวถึงขั้นตอนการนำฟังก์ชันอรรถประโยชน์ไปใช้โดยคร่าว สำหรับรายละเอียดวิธีการใช้และรหัสต้นกำเนิด (source code) ได้จัดทำไว้ในรูปแบบของ จานบันทึกชนิดอ่อน (diskette) ขนาด 360 กิโลไบต์ ซึ่งสามารถติดต่อขอสำเนาได้ที่ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ในที่นี้จะยกตัวอย่างเป็นระบบภาษาซีทั้งหมด สำหรับภาษาอื่นนั้นสามารถเลียนแบบ ขั้นตอนการทำงานที่กล่าวถึงนี้ได้

1. ถ้ายังไม่มียอดูลการตัดคำและเห็นว่ามอดูลการตัดคำ `thaicut` ที่แนบมา ในจานบันทึกชนิดอ่อนไม่เหมาะสม ให้เขียนมอดูลการตัดคำขึ้นมาเอง โดยให้ยัดวิธีการเชื่อมต่อใน ลักษณะแบบเดียวกับมอดูล `thaicut` หรือถ้ามีมอดูลการตัดคำเองอยู่แล้วให้แก้ไขโดย

- ก. ตัดส่วนที่เป็นการแทรกอักษรแบ่งคำออกจากมอดูล

- ข. ปรับการเชื่อมต่อให้เข้ากับที่กำหนดไว้ในบทที่ 3 คือเป็นแบบเดียวกับ `thaicut` และเพื่อให้เกิดความสะดวกให้ตั้งชื่อมอดูลว่า `thaicut` สำหรับ `thaicut` ที่แนบมาในจานบันทึกชนิดอ่อน อาศัยการปรับมอดูลการตัดคำในโปรแกรม `CU-writer` ให้เป็น

ไปตามการเชื่อมโยงที่กำหนดไว้ในบทที่ 3 ดังนั้น หากผู้อ่านเคยใช้ CU-writer แล้วก็จะเข้าใจการติดตั้งที่ใช้ข้อมูลเป็นพื้นฐานเพื่อใช้ความสามารถนำข้อมูล thaicut ที่มีในจานบันทึกชนิดอนไลน์ใช้ได้ง่าย

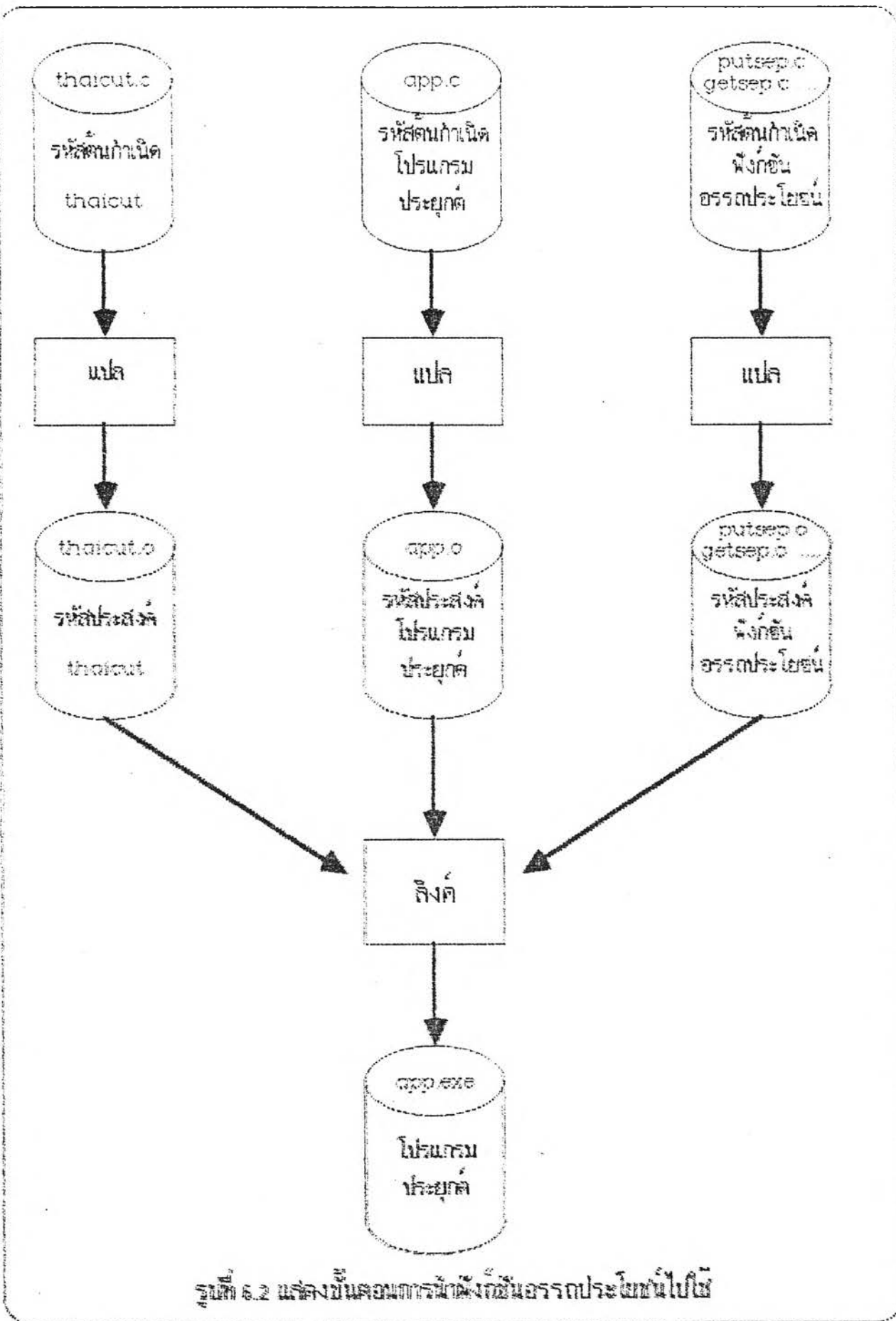
2. แปลงข้อมูลการติดตั้งตามข้อ 1 นี้ได้เป็นรหัสประสมค้ (object code) เก็บไว้เพื่อลิ้งค์ (link) กับโปรแกรมประยุกต์ที่จะเรียกใช้ข้อมูลการติดตั้ง

3. เขียนโปรแกรมประยุกต์ที่จัดการหรือจัดแปลงโปรแกรมประยุกต์ที่มีอยู่ เพื่อให้สามารถเรียกใช้ข้อมูลการติดตั้งเป็นไปตามเดิมเชื่อมโยงในบทที่ 3 (คือเรียกนำข้อมูลการติดตั้งที่ชื่อ thaicut.) สำหรับโปรแกรมประยุกต์ที่ต้องการแทรกอักขระแบบจำสามารถ เรียกใช้ฟังก์ชันอรรถประโยชน์อื่น ๆ นอกเหนือไปจาก thaicut เช่น putsep getsep putcut เป็นต้น

4. แปลงโปรแกรมประยุกต์และลิ้งค์เข้ากับออบเจกต์โคดของฟังก์ชันอรรถประโยชน์ เช่น thaicut putsep getsep เป็นต้น (สำหรับ thaicut อาจไม่ใช้ตัวที่มีให้ไว้ในจานบันทึกชนิดอนไลน์ แต่เขียนขึ้นมาเอง) ก็จะได้ โปรแกรมประยุกต์ที่ใช้งานได้

ขั้นตอนเหล่านี้แสดงไว้ในแผนผังรูปที่ 6.2

ในกรณีที่ เป็นภาษาอื่นนอกเหนือจากภาษาซี จำเป็นต้องเขียนฟังก์ชันอรรถประโยชน์ขึ้นมาใหม่ จะต้องเขียนใหม่ทั้งหมดหรือไม่ขึ้นกับขอบเขตของงานที่ใช้ แต่อย่างน้อยที่สุดจะต้องมีข้อมูล thaicut. ในภาษานั้น ทั้งนี้วิธีการเชื่อมโยงต่าง ๆ ยังคงเป็นไปตามที่กำหนดไว้ในบทที่ 3 สำหรับ thaicut. และบทนี้สำหรับฟังก์ชันอรรถประโยชน์อื่น ๆ



รูปที่ ๓.๒ ขั้นตอนการนำฟังก์ชันอรรถประโยชน์ไปใช้