บทที่ 3
ตัวเชื่อมโยงการตัดคำ

ในบทนี้จะได้กล่าวถึงการกำหนดการเชื่อมต่อระหว่างมอดูลการตัดคำ กับโปรแกรมซึ่งจะนำผลของการตัดคำไปใช้ เช่น โปรแกรมประมวลผลคำ โดยจะพิจารณาแยกเฉพาะหน้าที่ของการตัดคำ กล่าวคือ มอดูลการตัดคำจะวิเคราะห์และให้ผลเป็นตำแหน่งที่จะต้องตัดคำในเนื้อความที่กำหนด สำหรับการแทรกอักขระลงในตำแหน่งตัดคำเพื่อแสดงการตัดคำในเนื้อความนั้น ถือเป็นหน้าที่ปลีกย่อยที่มิได้เกี่ยวข้องโดยตรงกับการตัดคำ จึงจะแยกส่วนนี้ไปกล่าวถึงในบทถัดไป

3.1 การออกแบบโครงสร้าง (structure design)

3.1.1 การเชื่อมต่อ (coupling) และ ความยึดแน่น (cohesion)

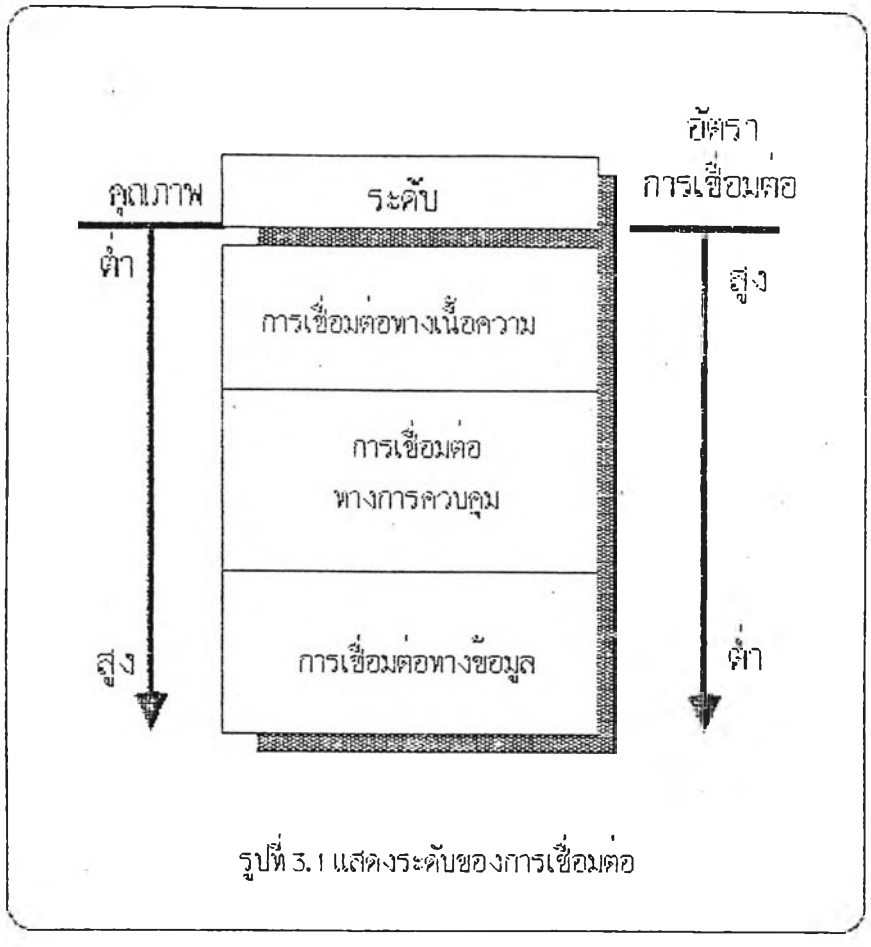
ในการออกแบบมอดูลที่ดีจะต้องคำนึงถึงคุณสมบัติต่างๆ คือ การเชื่อมต่อและความยึดแน่น การเชื่อมต่อ ใช้เพื่อที่จะลดจำนวนและชนิดของการติดต่อระหว่างมอดูล การติดต่อนี้อาจเป็นข้อมูล, การควบคุม หรือการเปลี่ยนแปลงข้อมูลภายในมอดูลอื่น

การเชื่อมต่อ แบ่งออกเป็น 3 ระดับ (Charette, 1987) คือ

1. การเชื่อมต่อทางเนื้อความ (content coupling) เกิดขึ้นเมื่อมอดูลหนึ่งทำการเปลี่ยนแปลงข้อมูลหรือคำสั่งในอีกมอดูลหนึ่ง

2. การเชื่อมต่อทางการควบคุม (control coupling) เกิดขึ้นเมื่อมอดูลหนึ่งผ่านข่าวสารไปยังอีกมอดูลหนึ่ง ซึ่งสามารถควบคุมขั้นตอนการทำงานของมอดูลนั้นได้

3. การเชื่อมต่อทางข้อมูล (data coupling) เกิดขึ้นเมื่อข้อมูลถูกส่งเป็นพารามิเตอร์ (parameter) ระหว่างมอดูล

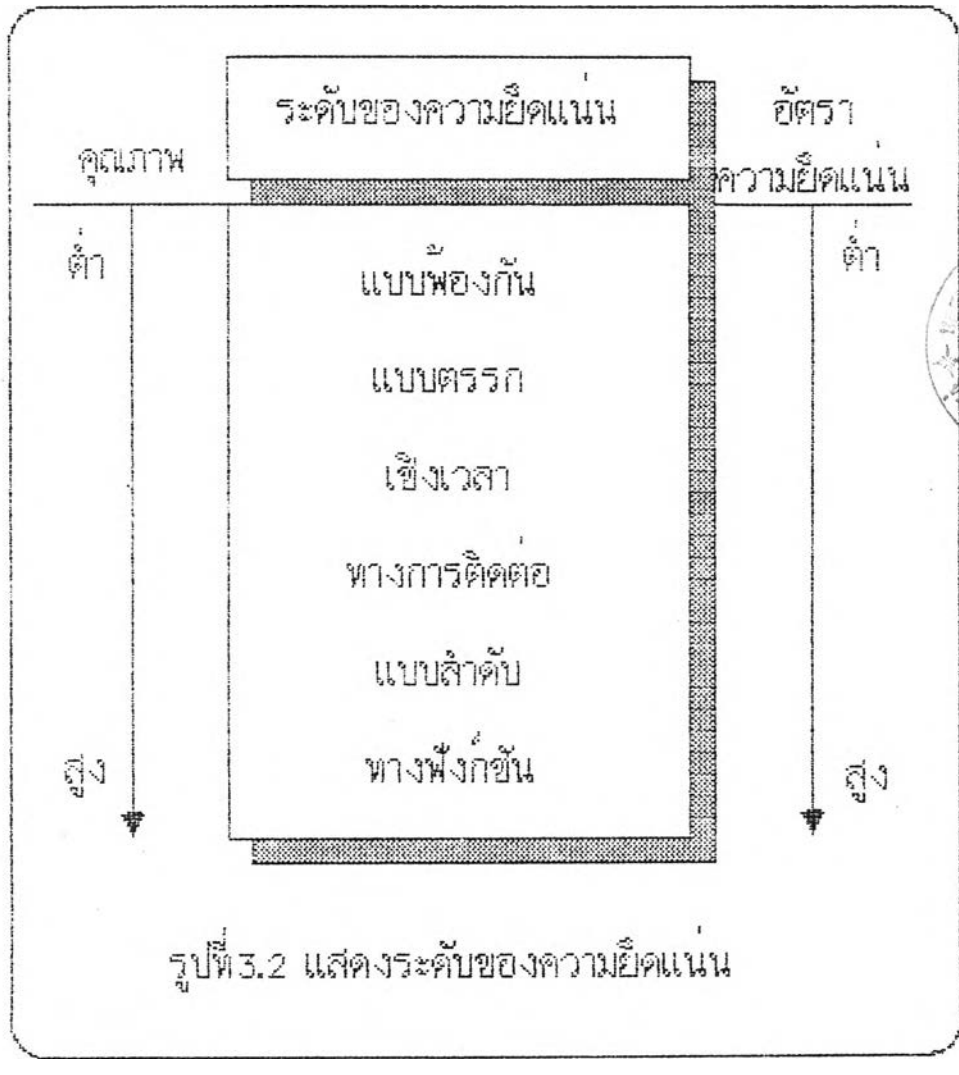


รูปที่ 3.1 แสดงระดับของการเชื่อมต่อ

รูปที่ 3.1 จะแสดงชนิดของการเชื่อมต่อ, ระดับ (degree) ของการเชื่อมต่อและจุดมุ่งหมายในการออกแบบ จะเห็นว่าในการออกแบบที่ดีควรจะให้มีระดับของการเชื่อมต่อต่ำ หรือการเชื่อมต่อทางข้อมูล

ความยึดแน่น เป็นการออกแบบเพื่อเพิ่มความสัมพันธ์ระหว่างแต่ละส่วนภายในมอดูล ถ้าความสัมพันธ์นี้เหนียวแน่นเพียงใดก็จะทำให้มอดูลมีลักษณะสมบูรณ์ในตัวเองยิ่งขึ้น

รูปที่ 3.2 แสดงชนิดของความยึดแน่นและระดับของความยึดแน่น ความยึดแน่นแบบบังเอิญ (coincidental cohesion) คือกรณีที่แต่ละส่วนในมอดูลไม่มีความสัมพันธ์ที่แน่นอนซึ่งกันและกัน ความยึดแน่นแบบตรรก (logical cohesion) เกิดขึ้นเมื่อมีความสัมพันธ์ในบางส่วนและมีเพียงผิวเผิน ความยึดแน่นเชิงเวลา (temporal cohesion) เกิดเมื่อความสัมพันธ์เกิดขึ้นเนื่องจากเวลา เช่นแต่ละส่วนทำงานพร้อมๆกัน ความยึดแน่นทางการติดต่อ (communicational cohesion) หมายถึงแต่ละส่วนในมอดูลใช้ชุดของข้อมูลอินพุตและเอาต์พุตเดียวกัน ความยึดแน่นแบบลำดับ (sequential cohesion) คือกรณีที่เอาต์พุตของส่วนหนึ่งเป็นอินพุตของอีกส่วนหนึ่งในมอดูล และความยึดแน่นด้วยฟังก์ชัน



รูปที่3.2 แสดงระดับของความยึดแน่น

(functional cohesion) เกิดขึ้นเมื่อทุกๆส่วนในมอดูลเกี่ยวข้องกับการกระทำฟังก์ชันเดียวกัน

ในการออกแบบมอดูลที่ดีจะต้องให้ได้ การเชื่อมต่อในระดับต่ำ และความยึดแน่นในระดับสูง กล่าวคือควรจะได้การเชื่อมต่อทางข้อมูลและความยึดแน่นด้วยฟังก์ชัน

3.1.2 <u>การปกปิดข่าว (information hiding)</u>

การปกปิดข่าวเป็นแนวทางที่สำคัญอีกอย่างหนึ่งในการออกแบบมอดูล ความหมายของการปกปิดข่าวก็คือ การที่มอดูลแต่ละมอดูลปกปิดข้อมูลภายในเกี่ยวกับขั้นตอนการทำงานต่างๆภายในมอดูล รวมทั้งโครงสร้างข้อมูลภายใน มอดูลจะกำหนดตัวเชื่อมโยง (interface) ที่แน่นอนไว้เพื่อให้มอดูลอื่นติดต่อได้ ทำให้มอดูลสามารถเปลี่ยนแปลงได้ง่าย โดยไม่มีผลกระทบต่อส่วนอื่น

ลักษณะเช่นนี้จะสอดคล้องกับการออกแบบให้มีการเชื่อมต่อทางข้อมูล ซึ่งจะเป็นการติดต่อกันผ่านพารามิเตอร์ และการที่ข้อมูลส่วนที่เกี่ยวข้องจะถูกรวมอยู่ด้วยกัน จะทำให้มีความยึดแน่นสูง

3.2 ตัวเชื่อมโยง (interface)

จากวิธีการออกแบบมอดูลข้างต้นจะเห็นได้ว่า ในการติดต่อกับมอดูลแต่ละมอดูลนั้น จะติดต่อผ่านตัวเชื่อมโยง Spencer (1986) ได้ให้คำจำกัดความของตัวเชื่อมโยงว่า " ตัวเชื่อมโยงหมายถึง ข้อกำหนดของการติดต่อระหว่างสองหน่วยย่อยของโปรแกรม การกำหนดตัวเชื่อมโยงที่ดีจะทำให้สามารถใช้ส่วนของโปรแกรมได้ โดยไม่ต้องรู้ถึงการทำงานภายในของส่วนย่อยนั้น "

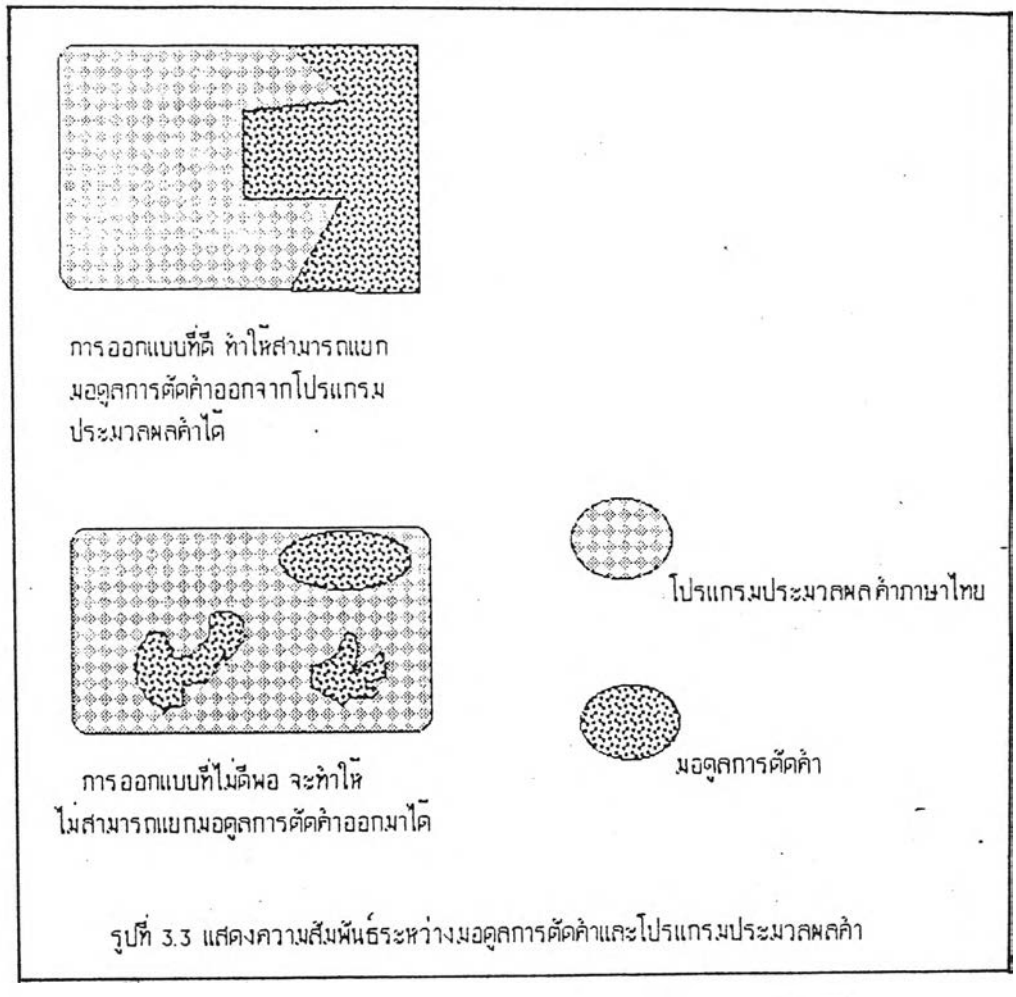
3.2.1 ตัวเชื่อมโยงมอดูลการตัดคำในโปรแกรมประมวลผลคำ(word processor)

ดังที่ได้กล่าวมาแล้วในบทนำ (บทที่ 1) ว่า ผู้ออกแบบโปรแกรมประมวลผลคำภาษาไทย (Thai word processor) มักจะออกแบบโดยมองฟังก์ชันของการตัดคำเป็นส่วนหนึ่งของโปรแกรมประมวลผลคำ การออกแบบในส่วนของมอดูลการตัดคำนี้ หากผู้ออกแบบคำนึงถึงวิธีการออกแบบที่ดี ก็จะสามารถแยกมอดูลการตัดคำนี้ออกมาจากโปรแกรมประมวลผลคำได้ แต่ถ้าหากมีการออกแบบไม่ดีพอ มอดูลการตัดคำก็อาจจะถูกยึดติดเป็นส่วนหนึ่งของโปรแกรมประมวลผลคำทำให้ไม่สามารถแยกออกได้ ดังแสดงในรูปที่ 3.3

อย่างไรก็ตาม สำหรับมอดูลการตัดคำที่ได้รับการออกแบบที่ดี การที่ผู้ออกแบบคำนึงถึงโปรแกรมประมวลผลคำเป็นหลัก ลักษณะของตัวเชื่อมโยงที่ใช้มักจะมีลักษณะเฉพาะที่เป็นของโปรแกรมประมวลผลคำ ตัวอย่างเช่น ตัวเชื่อมโยงมอดูลการตัดคำในโปรแกรม Cu-writer ซึ่งเป็นโปรแกรมประมวลผลคำโปรแกรมหนึ่ง จะเขียนด้วยภาษาซีได้ดังนี้

```
char *FINDCUT(LEFTMARGIN, RIGHTMARGIN, ENDOFTEXT)
```

จะเห็นว่าพารามิเตอร์ที่ใช้ในตัวเชื่อมโยง คือ LEFTMARGIN (ตำแหน่งกั้นหน้าซ้าย) และ RIGHTMARGIN (ตำแหน่งกั้นหน้าขวา) นั้น เป็นลักษณะของข้อมูลที่มีเฉพาะในโปรแกรมประเภทประมวลผลคำเท่านั้น ดังนั้น การนำมอดูลการตัดคำนี้ไปใช้กับโปรแกรมประเภทอื่นๆในทันทีจึงทำไม่ได้ ดังแสดงในรูปที่ 3.4 ต้องมีการแก้ไขตัวมอดูลการตัดคำ รวมทั้งตัวเชื่อมโยงด้วย



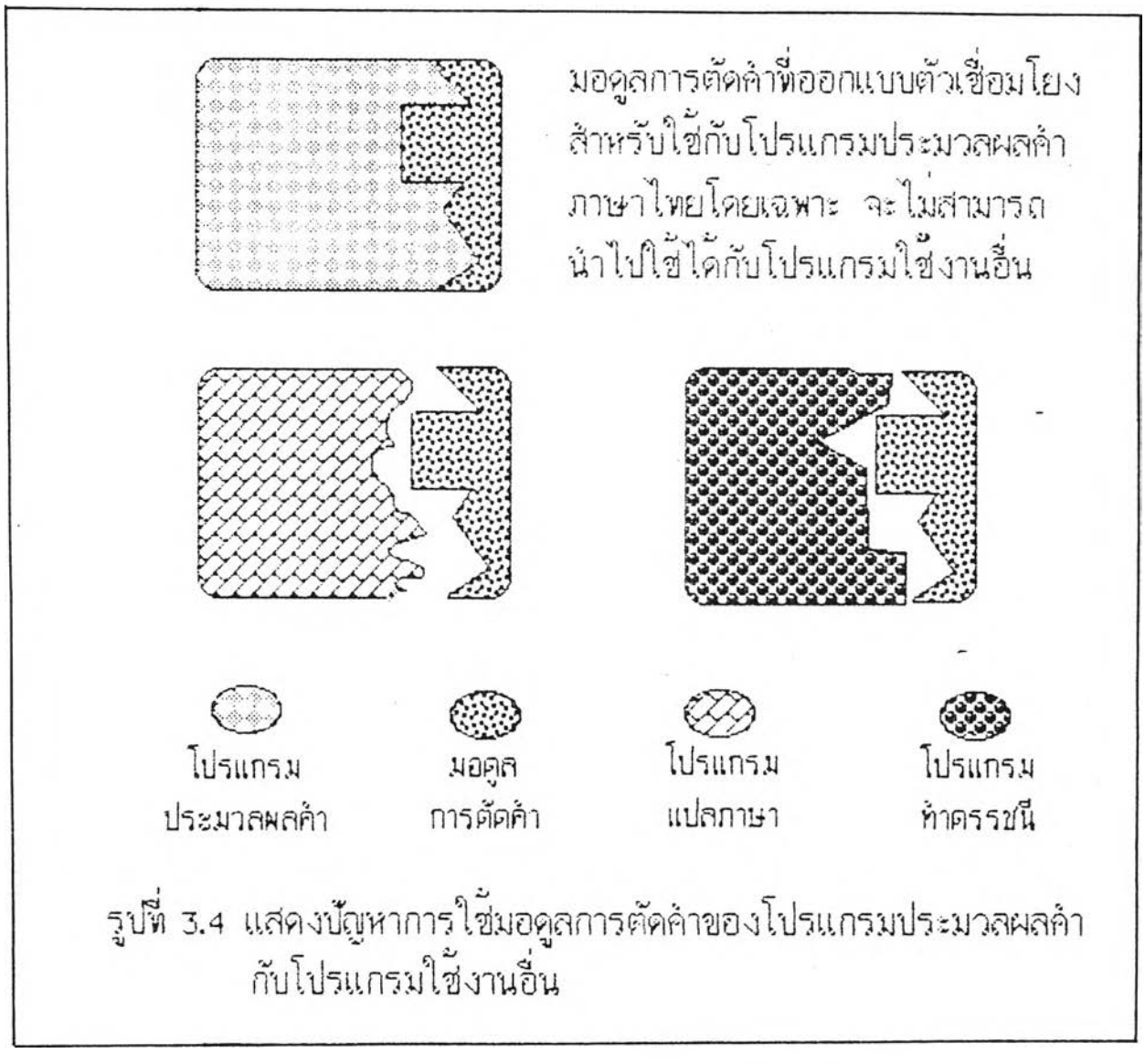
รูปที่ 3.3 แสดงความสัมพันธ์ระหว่างมอดูลการตัดคำและโปรแกรมประมวลผลคำ

3.2.2 รูปแบบของการตัดคำแบบกลุ่ม (batch) และโต้ตอบ (Interactive)

ในการใช้งานมอดูลการตัดคำ อาจจะมีรูปแบบการใช้งานในแบบกลุ่ม กล่าวคือมีการเตรียมข้อมูลเป็นกลุ่มก้อน เช่น เป็นประโยคหรือเป็นย่อหน้า (paragraph) หรืออาจจะเป็นทั้งแฟ้ม (file) แล้วจึงค่อยส่งข้อมูลนั้นไปให้มอดูลการตัดคำทำการตัดคำอีกทีหนึ่ง

อีกรูปแบบหนึ่งจะเป็นการใช้งานในลักษณะโต้ตอบ คือเมื่อได้ข้อมูลมาหนึ่งตัวอักษร เช่น อาจจะได้มาจากการรับจากแป้นพิมพ์ (keybord) ก็จะทำการส่งไป

ให้มอดูลการตัดคำทันที มอดูลการตัดคำก็จะพยายามตัดคำให้ได้ จากข้อมูลเท่าที่มี ตัวเชื่อมโยงของมอดูลการตัดคำที่ออกแบบ จะต้องสามารถครอบคลุมและทำงานกับรูปแบบ ทั้ง 2 แบบนี้ได้อย่างมีประสิทธิภาพ



รูปที่ 3.4 แสดงปัญหาการใช้มอดูลการตัดคำของโปรแกรมประมวลผลคำกับโปรแกรมใช้งานอื่น

3.3 การออกแบบมอดูลการตัดคำ

3.3.1 โครงสร้างข้อมูล (data structure) และ พฤติกรรมของขั้นตอนวิธี (algorithm) การตัดคำ

ดังที่ได้กล่าวมาแล้วในหัวข้อ 2.1 ว่า ขั้นตอนวิธีการตัดคำนั้นจะมีอยู่ 2 วิธี คือใช้พจนานุกรม (dictionary) และใช้กฎ (rules) ลักษณะการทำงานของขั้นตอนวิธีทั้งสองจะต้องมีการกวาดตรวจ (scan) ไปในข้อความที่ต้องการตัดคำ เพื่อนำไปค้นหาเปรียบเทียบกับคำในพจนานุกรม หรือนำไปเข้ากฎต่างๆ เพื่อที่จะหาจุดตัดคำ

เพื่อเอื้ออำนวยต่อพฤติกรรมของขั้นตอนวิธีการตัดคำนี้ มอดูลการตัดคำจึงจำเป็นต้องมีโครงสร้างข้อมูลภายในที่เป็นบัฟเฟอร์ (buffer) สำหรับการกวาดตรวจของขั้นตอนวิธีการตัดคำ

3.3.2 ตัวเชื่อมโยง

การเชื่อมโยงกับมอดูลการตัดคำ จะใช้ข้อมูล 2 แบบคือ ข้อมูลที่ส่งไปซึ่งกำหนดให้เป็นพารามิเตอร์ และข้อมูลการส่งกลับ

รูปแบบการส่งพารามิเตอร์

เพื่อให้มอดูลการตัดคำสามารถนำไปใช้งานได้กับทุกๆโปรแกรมประยุกต์ (application program) จำเป็นต้องมีตัวเชื่อมโยงที่เป็นกลาง สามารถเข้าได้กับทุกๆ โปรแกรมประยุกต์ การที่จะทำเช่นนี้ได้ ตัวเชื่อมโยงจะต้องไม่ใช้ข้อมูลส่วนที่เป็นข้อมูลเฉพาะของโปรแกรมประยุกต์ใด จะต้องเป็นข้อมูลที่มีอยู่ในทุกๆโปรแกรมประยุกต์

ข้อมูลส่วนที่เป็นกลางที่สุดสำหรับทุกๆโปรแกรมประยุกต์ ก็คือตัวข้อความ (text) ภาษาไทยเอง และเพื่อให้ตัวเชื่อมโยงสามารถรองรับการตัดคำในแบบโต้ตอบ ซึ่งมีลักษณะการทำงานทีละตัวอักษร จึงกำหนดให้ตัวเชื่อมโยงของมอดูลการตัดคำรับข้อมูลครั้งละหนึ่งตัวอักษร และมอดูลการตัดคำจะส่งตำแหน่งของจุดตัดคำกลับไป สำหรับการตัดคำแบบกลุ่มนั้นก็สามารถใช้ตัวเชื่อมโยงในรูปแบบเดียวกันนี้ได้ โดยการรับข้อความมาเป็นกลุ่มแล้วทยอยส่งทีละตัวอักษรให้กับมอดูลการตัดคำ

ดังนั้น โปรแกรมประยุกต์ที่เรียกใช้มอดูลการตัดคำ จะเรียกใช้โดยส่งพารามิเตอร์เป็นตัวอักษร 1 ตัว ไปให้มอดูลการตัดคำ พารามิเตอร์นี้จะมีชนิด (type) เป็นแบบตัวอักษร (character) ขนาด 1 ไบต์

รูปแบบการส่งค่ากลับ

รูปแบบของการส่งค่าจุดตัดคำกลับไปให้มอดูลที่เรียก มอดูลการตัดคำนั้น เพื่อให้สอดคล้องกับการส่งตัวอักษรให้มอดูลการตัดคำ มอดูลการตัดคำจะส่งค่ากลับไปในรูปของออฟเซต (offset) ของตัวอักษรที่ส่งไปให้มอดูลการตัดคำ ค่าออฟเซตนี้จะเป็นตัวเลขจำนวนเต็ม (integer) ขนาด 2 ไบต์ โดยจะมีความหมายดังนี้ คือ

-n หมายถึง สามารถตัดคำได้ที่ตัวอักษรก่อนหน้าตัวอักษรปัจจุบัน n ตัว

0 หมายถึง ตัดคำได้ที่ตัวอักษรปัจจุบัน (ตัดหลังตัวอักษรปัจจุบัน)

1 หมายถึง ยังไม่สามารถตัดคำได้จากข้อมูลที่มี

ตัวอย่างเช่น ถ้าโปรแกรมต้องการหาจุดตัดคำของข้อความภาษาไทยว่า "กลวิธี" โปรแกรมจะเริ่มส่งตัวอักษรทีละตัวให้กับมอดูลการตัดคำ โดยเริ่มจาก "ก" เมื่อมอดูลการตัดคำรับตัวอักษร "ก" มาก็จะพบว่ายังไม่สามารถหาจุดตัดคำได้ ก็จะส่งค่า 1 กลับมา โปรแกรมก็จะทยอยส่ง "ล","ว","ิ","ธ","ี" ไปให้ เมื่อมอดูลการตัดคำรับตัวอักษร "ล" และ "ว" มาตามลำดับ ก็ยังคงตัดคำไม่ได้ ก็จะส่งค่า 1 กลับมาทั้งสองครั้ง และเมื่อรับตัวอักษร "ิ" มา มอดูลการตัดคำพบว่าสามารถหาจุดตัดคำได้ที่ ตัวอักษร "ล" มอดูลการตัดคำก็จะคำนวนค่าออฟเซตของตัวอักษร "ล" จากตัวอักษรปัจจุบันคือ "ิ" ซึ่งจะได้ค่าออฟเซตเป็น -2 กลับไปให้โปรแกรม หลังจากนั้นเมื่อมอดูลการตัดคำรับตัวอักษร "ธ" และ "ี" มา และหาจุดตัดคำได้ที่ตัว "ี" ซึ่งเป็นตัวอักษรปัจจุบัน ก็จะส่งค่า 0 กลับไป

ข้อจำกัดของค่าส่งกลับ

ค่าของออฟเซตที่มอดูลการตัดคำส่งกลับมานั้น จะมีค่าต่ำสุดได้ไม่เกิน -32768 เนื่องจากเป็นข้อจำกัดที่ขนาดของเลขจำนวนเต็ม และสำหรับโปรแกรมประยุกต์ จะต้องดูแลการใช้งานของบัฟเฟอร์ที่ใช้เก็บข้อความซึ่งต้องการส่งให้มอดูลการตัดคำรวมทั้งตัวชี้ (pointer) ของบัฟเฟอร์เอง เพื่อไม่ให้เกิดการทำงานผิดพลาด เนื่องจาก

การล้นของข้อมูล (data overflow)

แฟล็ก (flag) พิเศษ

ดังได้กล่าวแล้วในหัวข้อ 3.3.2 ว่าภายในมอดูลการตัดคำจะมีบัฟเฟอร์อยู่ เพื่อที่จะเก็บตัวอักษรตัวก่อนหน้าไว้ ในบางครั้งเราจำเป็นต้องมีการเปลี่ยนแปลงข้อมูลภายในบัฟเฟอร์ โดยเฉพาะในรูปแบบการตัดคำแบบโต้ตอบ เช่น การรับข้อมูลจากแป้นพิมพ์ ในบางครั้งอาจจะมีการพิมพ์ตัวอักษรถอยหลัง (backspace) เพื่อแก้ไขข้อมูล เราจำเป็นต้องมีการแก้ไขข้อมูลในบัฟเฟอร์ของมอดูลการตัดคำด้วย

นอกจากนี้การที่มอดูลการตัดคำมีบัฟเฟอร์ภายในของตัวเอง ข้อมูลตัวอักษรที่รับมาจะยังคงค้างอยู่ภายในบัฟเฟอร์ ถ้าตัวโปรแกรมใช้งานรับข้อมูลเข้ามาใหม่ ซึ่งเป็นข้อมูลคนละชุดกับข้อมูลเดิม แล้วส่งให้มอดูลการตัดคำ มอดูลการตัดคำก็จะไม่รับรู้ว่ามีการเปลี่ยนแปลงข้อมูลเป็นคนละชุดแล้ว ข้อมูลที่เข้ามาใหม่ก็จะมาต่อกับข้อมูลเดิมทำให้การหาจุดตัดคำเกิดข้อผิดพลาดได้

เพื่อให้ตัวเชื่อมโยงของมอดูลการตัดคำสามารถรองรับความต้องการการเปลี่ยนแปลงของบัฟเฟอร์ภายในได้ จึงกำหนดให้มีข้อมูลที่เป็นแฟล็กพิเศษ 2 ตัว คือ

ตัวอักษรถอยหลัง ซึ่งตรงกับรหัสแอสกี 8 เป็นการลบข้อมูลที่อยู่ภายในบัฟเฟอร์ตัวท้ายสุด

-1 เป็นแฟล็กสำหรับการลบล้าง (clear) บัฟเฟอร์ รวมทั้งสถานะภายในต่างๆ ของมอดูลการตัดคำ

3.4 ตัวเชื่อมโยงมอดูลการตัดคำในภาษาซี

จากการออกแบบมอดูลการตัดคำในข้างต้น เราสามารถนำมาเขียนเป็นฟังก์ชัน (function) ในภาษาซีได้ดังนี้

```
int  thaicut (c)
int  c;
```

โดย 'thaicut' เป็นชื่อของมอดูลการตัดคำและ 'c' เป็นตัวอักษรที่ส่งให้มอดูลการตัดคำ และมอดูลการตัดคำนี้จะส่งค่ากลับไปเป็นตัวเลข (integer) การที่ไม่กำหนดให้ 'c' มีชนิด (type) เป็น "char" เนื่องจากมีการส่งค่าตัวอักษรที่เป็นแฟล็กพิเศษ คือ -1 ด้วย

สำหรับโปรแกรมประยุกต์จะเรียกใช้มอดูลการตัดคำได้ดังนี้คือ

```
char c;
int offset;
. . .
offset = thaicut(c);
. . .
```

3.5 การใช้วิธีการเชื่อมโยงสำหรับภาษาอื่น

ข้อกำหนดของตัวเชื่อมโยงระบุเพียงว่า โปรแกรมที่ใช้มอดูลการตัดคำต้องส่งตัวอักษรไปให้มอดูลการตัดคำ ส่วนมอดูลการตัดคำจะส่งผลลัพธ์ของการตัดคำในลักษณะออฟเซตกลับมาให้โปรแกรมสำหรับแต่ละตัวอักษรที่ได้รับ ซึ่งลักษณะของการเชื่อมโยงนี้ จะเห็นว่าสามารถเขียนในภาษาอะไรก็ได้ ไม่ว่าจะเป็น โคบอล (COBOL), ฟอร์แทรน (FORTRAN), ซี (C), ปาสคาล (Pascal) หรือแม้แต่ แอสเซมบลี (ASSEMBLY)

ตัวอย่างเช่น ในภาษาฟอร์แทรน มอดูลการตัดคำจะขึ้นต้นด้วย

```
FUNCTION   INTEGER  THAICUT(X)
CHARACTER  X
```

การเรียกใช้มอดูลนี้จะเป็น

```
CHAR  C
. . .
IOFFSET  =  THAICUT(C)
```

016291

ดังนั้น จะเห็นได้ว่า ในระบบที่เป็นภาษาอื่น หากโปรแกรมประยุกต์และมอดูลการตัดคำเป็นภาษาเดียวกัน เพียงแต่ปรับการเรียกใช้มอดูลการตัดคำและการตอบสนองของมอดูลการตัดคำ ให้เป็นไปตามข้อกำหนดของตัวเชื่อมโยงนี้ (กล่าวคือ ส่งไปครั้งละหนึ่งตัวอักษร ได้รับออฟเซตกลับมา)

การปรับให้เข้ากับตัวเชื่อมโยงนี้ อาจจะยุ่งยากซับซ้อนอยู่บ้าง แต่เมื่อปรับแล้ว และตัวเชื่อมโยงนี้เป็นที่นิยมกันแพร่หลาย จะทำให้มอดูลการตัดคำที่ปรับแล้ว สามารถไปสวมประกอบใช้กับโปรแกรมประยุกต์ต่างๆได้อย่างกว้างขวางโดยไม่ต้องแก้ไข

3.6 การใช้ตัวเชื่อมโยงข้ามภาษา

ในกรณีที่มอดูลการตัดคำกับโปรแกรมประยุกต์เป็นคนละภาษา ถึงแม้ว่าจะมีการปรับตัวเชื่อมโยงให้เป็นไปตามที่กำหนดแล้วก็ตาม แต่การที่โปรแกรมประยุกต์จะเรียกใช้มอดูลการตัดคำได้นั้นต้องขึ้นกับตัวแปลภาษา (compiler) ของระบบว่าจะสามารถให้เรียกโปรแกรมย่อยที่มาจากคนละภาษาได้หรือไม่

ตัวอย่างเช่นในตัวแปลภาษา Turbo C ซึ่งเป็นตัวแปลภาษาซีสำหรับเครื่องไมโครคอมพิวเตอร์ โปรแกรมภาษาซีสามารถเรียกใช้โปรแกรมย่อยที่เขียนด้วยภาษาแอสเซมบลีได้ (Borland 1987) แต่ทั้งนี้จะต้องมีการปรับตัวโปรแกรมอยู่บ้าง เพราะลักษณะโครงสร้างของตัวโปรแกรมที่ผ่านการแปลแล้ว ในภาษาต่างๆจะไม่เหมือนกัน ซึ่งการเรียกข้ามภาษากันนี้ขึ้นกับความสามารถของตัวแปลภาษาของระบบ

3.7 องค์ประกอบข้างเคียงอื่นๆเกี่ยวข้องกับการตัดคำ

ในการออกแบบตัวเชื่อมโยงนี้ หลักการพื้นฐานที่สำคัญประการหนึ่ง คือ ได้กำหนดให้มอดูลการตัดคำต้องทำงานเฉพาะการตัดคำเท่านั้น กล่าวคือ มอดูลการตัดคำเมื่อรับข้อความ (ในลักษณะทีละตัวอักษร) จะพิจารณาว่าจุดใดเป็นตำแหน่งที่จะต้องตัดคำ ทั้งนี้โดยไม่รวมหน้าที่อื่นๆที่เกี่ยวข้อง เช่น การแทรกอักขระเพื่อแสดงตำแหน่งการแบ่งคำ เป็นต้น เหตุที่กำหนดเช่นนี้เพราะโปรแกรมประยุกต์บางประเภท เช่น โปรแกรมประมวลผลคำ จะต้องการทราบเพียงตำแหน่งของการตัดคำ เพื่อใช้ในการจัดหลังของปรรทัด โดยไม่ต้องการให้มีการแทรกอักขระใดๆเพิ่มเติมหลังจุดที่ตัดคำนั้น

ดังนั้น เป้าหมายของวิทยานิพนธ์นี้อีกประการหนึ่งคือ แยกองค์ประกอบและหน้าที่

ข้างเคียงเหล่านี้ออกจากหน้าที่แรก คือ การตัดคำ ดังนั้น มอดูลการตัดคำที่มีหน้าที่ข้างเคียงเหล่านี้ปะปนอยู่ ถ้าต้องการปรับให้เข้ากับลักษณะของตัวเชื่อมโยง จำเป็นต้องแยกส่วนหน้าที่ข้างเคียงต่างๆออกจากมอดูล ให้เหลือแต่หน้าที่การตัดคำแต่เพียงอย่างเดียวเท่านั้น มิฉะนั้นแล้วจะทำให้มอดูลการตัดคำมีลักษณะการเชื่อมต่อที่ไม่เหมาะสมต่อการนำไปใช้กับโปรแกรมประยุกต์ลักษณะต่างๆในวงกว้าง

แต่จากการที่มีโปรแกรมประยุกต์จำนวนมากที่ต้องอาศัยฟังก์ชันข้างเคียงนี้ ผู้วิจัยจึงได้กำหนดแนวทางและขั้นตอนวิธีที่เกี่ยวข้องกับการแทรกอักขระ เพื่อแสดงจุดแบ่งคำที่โปรแกรมประยุกต์สามารถเรียกใช้ได้เพื่อเสริมการตัดคำพื้นฐาน โดยจะได้นำเสนอส่วนนี้ในบทถัดไป (บทที่ 4 และ 5)

3.8 สรุป

ในบทนี้ได้กล่าวถึงวัตถุประสงค์หลักประการหนึ่งของวิทยานิพนธ์นี้ คือ การแยกหน้าที่การตัดคำออกเป็นเอกเทศ และกำหนดวิธีการเชื่อมต่อที่เหมาะสมระหว่างมอดูลการตัดคำกับโปรแกรมที่นำผลของการตัดคำไปใช้ ตัวเชื่อมโยงที่กำหนดขึ้นนี้ แม้ว่าจะกำหนดขึ้นมาโดยอาศัยภาษาซี แต่ก็สามารถนำหลักการไปใช้กับภาษาอื่นๆได้ ทั้งนี้จำต้องมีการปรับมอดูลการตัดคำและโปรแกรมประยุกต์ในสองส่วน คือ ให้เข้ากับตัวเชื่อมโยง และแยกหน้าที่ข้างเคียง เช่น การแทรกอักขระแบ่งคำ ออกจากมอดูลการตัดคำ ในบทที่ 4 และ บทที่ 5 จะได้กล่าวถึงวิธีการในการแทรกอักขระเพื่อบ่งบอกจุดที่มีการตัดคำ ในลักษณะที่สามารถจะใช้กับงานต่างๆได้ในวงกว้าง