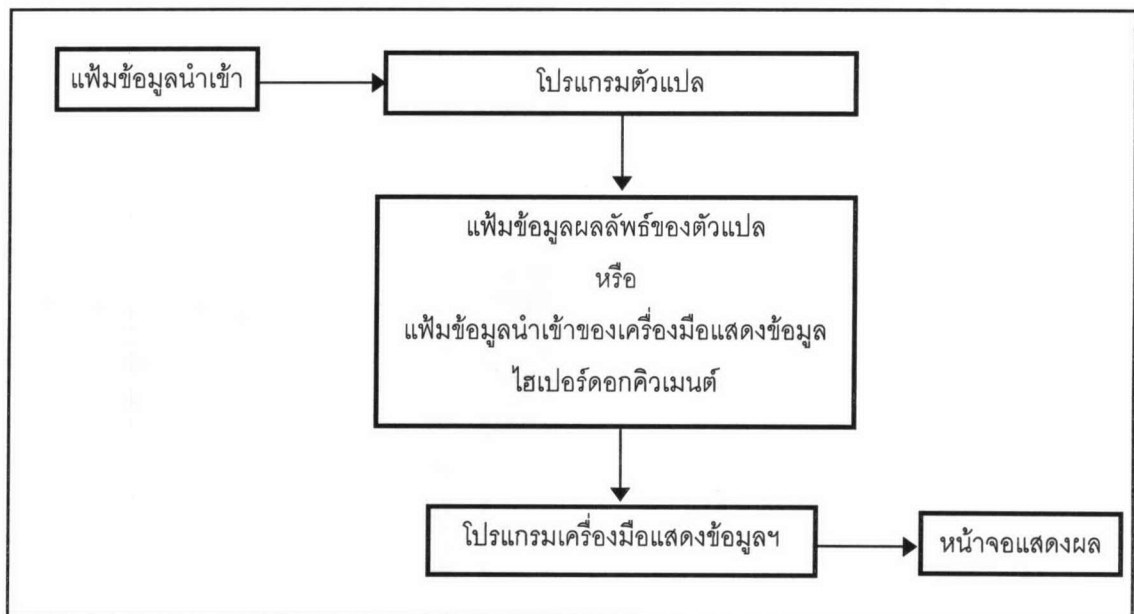


บทที่ 3

การออกแบบและพัฒนาระบบ

จากการวิเคราะห์ระบบ เพื่อพัฒนาโปรแกรมตัวแปลและเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ สามารถออกแบบระบบเป็น 2 ขั้นตอน คือ ขั้นตอนของโปรแกรมตัวแปล และขั้นตอนของโปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ ในแต่ละขั้นตอน จะประกอบด้วย ข้อมูลนำเข้า การประมวลผล และข้อมูลผลลัพธ์ โดยมีกระบวนการทำงาน ดังนี้



รูปที่ 3-1 แสดงกระบวนการทำงานของระบบ

เพิ่มข้อมูลนำเข้า เป็นเพิ่มข้อมูลในรูปแบบ RTF ที่จะต้องมีเนื้อหาตรงตามข้อกำหนดที่จะกล่าวในส่วนต่อไป จากนั้น โปรแกรมตัวแปล จะอ่านข้อมูลเข้ามาเพื่อค้นหาและแยกแยะข้อมูลส่วนที่เป็นการเชื่อมโยงแล้วจัดเก็บลงในเพิ่มข้อมูลผลลัพธ์ ในรูปแบบที่โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ จะสามารถอ่านได้ และเมื่อโปรแกรมเครื่องมือแสดงข้อมูลอ่านเพิ่มข้อมูลนี้เข้ามา โปรแกรมจะสามารถแสดงผลข้อมูลบนหน้าต่าง และมีความสามารถตามที่กำหนดในขอบเขตงาน

3.1 เพิ่มข้อมูลนำเข้า

เพิ่มข้อมูลนำเข้า ต้องอยู่ในรูปแบบของ RTF สร้างได้โดยใช้โปรแกรมที่มีความสามารถจัดเก็บข้อมูลเอกสารในรูปแบบ RTF ได้ เช่น ไมโครซอฟเวิร์ด เป็นต้น เนื้อหาภายในเพิ่มข้อมูลจะแบ่งเป็นหัวข้อ (Topic)

เรียงต่อกันไป โดยแต่ละหัวข้อจะคั่นด้วยรหัสการขึ้นหน้าใหม่ (hard page break หรือใช้รหัส RTF “\par”) เนื้อหาภายในแต่ละหัวข้อ จะมีชื่อหัวข้อ 1 ชื่อเสมอซึ่งจะต้องไม่ซ้ำกับชื่อหัวข้ออื่นในแฟ้มข้อมูลนำเข้า ชื่อหัวข้อนี้ใช้ในการค้นหาข้อมูลของหัวข้อที่ต้องการ หัวข้ออาจมีหรือไม่มีชื่อเรื่อง (Title) ประจำหัวข้อก็ได้ ถ้ามีจะมีได้เพียง 1 ชื่อเรื่องเท่านั้น ชื่อเรื่องมักเป็นชื่อที่สื่อความหมายกับผู้ใช้ ในขณะที่ชื่อหัวข้อจะเป็นชื่อสั้น ๆ กะทัดรัดสำหรับการอ้างอิงในโปรแกรม ดังนั้น ชื่อเรื่องจะใช้ในการแสดงให้ผู้เลือกใช้เลือกหัวข้อที่ต้องการ นอกจากนี้ แต่ละหัวข้อยังอาจมีหรือไม่มีคำหลัก (Keyword) ที่ใช้สำหรับการค้นหาและเข้าถึงหัวข้อที่ต้องการ ใน 1 หัวข้อมีคำหลักได้ไม่จำกัดจำนวน คำหลักของแต่ละหัวข้ออาจซ้ำกันได้

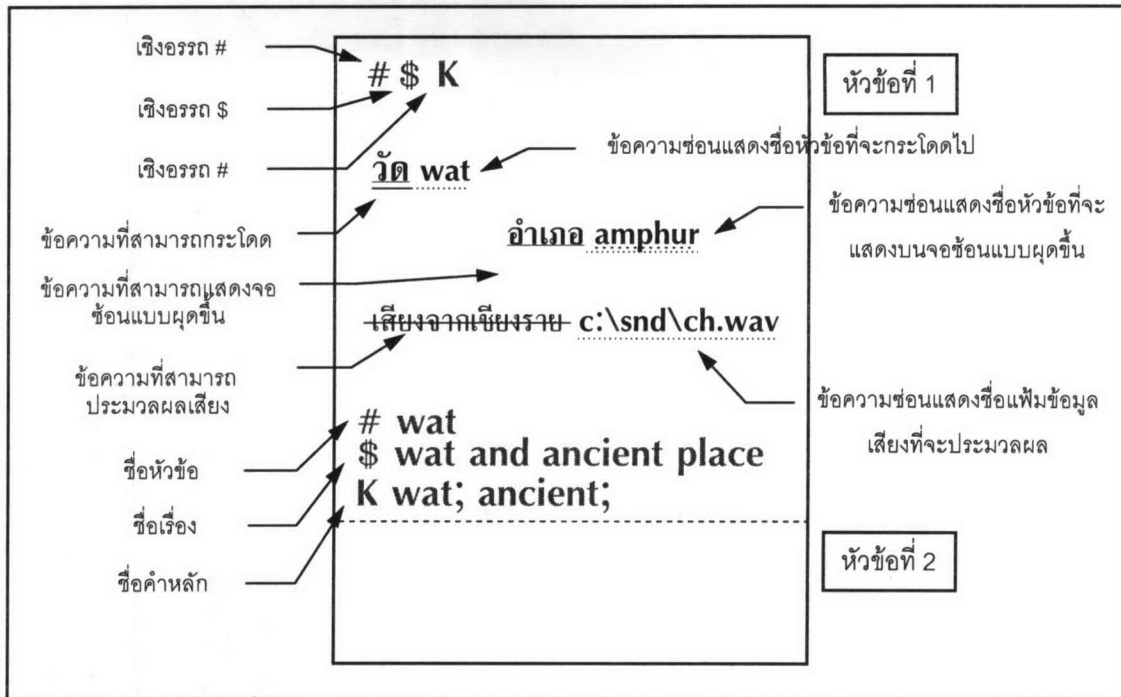
ข้อกำหนดสำหรับชื่อหัวข้อ ชื่อเรื่อง และคำหลัก คือ จะต้องอยู่ที่คอลัมน์แรกของแต่ละ หัวข้อเรียงต่อเนื่องกันไม่เว้นวรรค แล้วจึงเป็นข้อมูลของหัวข้อนั้น การกำหนดทำได้โดยกำหนดเป็นเชิงอรรถ (Footnote) ดังนี้

- # สำหรับ ชื่อหัวข้อ โดยพิมพ์ชื่อหัวข้อเป็นข้อความในเชิงอรรถ ไม่มีเว้นวรรค มีความไม่เกิน 64 ตัวอักษร
- \$ สำหรับ ชื่อเรื่อง โดยพิมพ์ชื่อเรื่องเป็นข้อความในเชิงอรรถ เว้นวรรคได้ มีความไม่เกิน 128 ตัวอักษร
- K สำหรับ คำหลัก โดยพิมพ์ชื่อคำหลักเป็นข้อความในเชิงอรรถ แล้วตามท้ายด้วยเครื่องหมาย ; ถ้ามีคำหลักหลายคำให้พิมพ์ต่อเนื่องกันไปจนหมด คำหลักเว้นวรรคได้ และมีความไม่เกิน 128 ตัวอักษร

ภายในเนื้อหาของแต่ละหัวข้อ สามารถกำหนดการเชื่อมโยง และการประมวลผลเสียง ได้ใน 3 รูปแบบ โดยใช้รูปแบบการขีดเส้นของตัวอักษร และ ข้อความซ่อน (hidden text) ดังนี้

- **ขีดเส้นใต้คู่ (double underline)** หมายถึง ข้อความนั้นจะเชื่อมโยงไปยังหัวข้ออื่นด้วยการกระโดด (jump) และเมื่อหมดข้อความนี้แล้วให้ต่อท้ายด้วยข้อความซ่อนซึ่งระบุชื่อหัวข้อที่ต้องการการกระโดดไป
- **ขีดเส้นใต้เดี่ยว (single underline)** หมายถึง ข้อความนั้นจะเชื่อมโยงไปยังหัวข้ออื่นในรูปแบบของจอซ้อนแบบผุดขึ้น (popup window) และเมื่อหมดข้อความนี้แล้วให้ต่อท้ายด้วยข้อความซ่อนซึ่งระบุชื่อหัวข้อที่ต้องการแสดงในจอซ้อนแบบผุดขึ้น
- **ขีดเส้นผ่านข้อความ (strike-thru)** หมายถึง ข้อความนั้นจะประมวลผลเสียง (sound execution) และเมื่อหมดข้อความนี้แล้วให้ต่อท้ายด้วยข้อความซ่อนซึ่งระบุชื่อแฟ้มข้อมูลรูปแบบ .wav

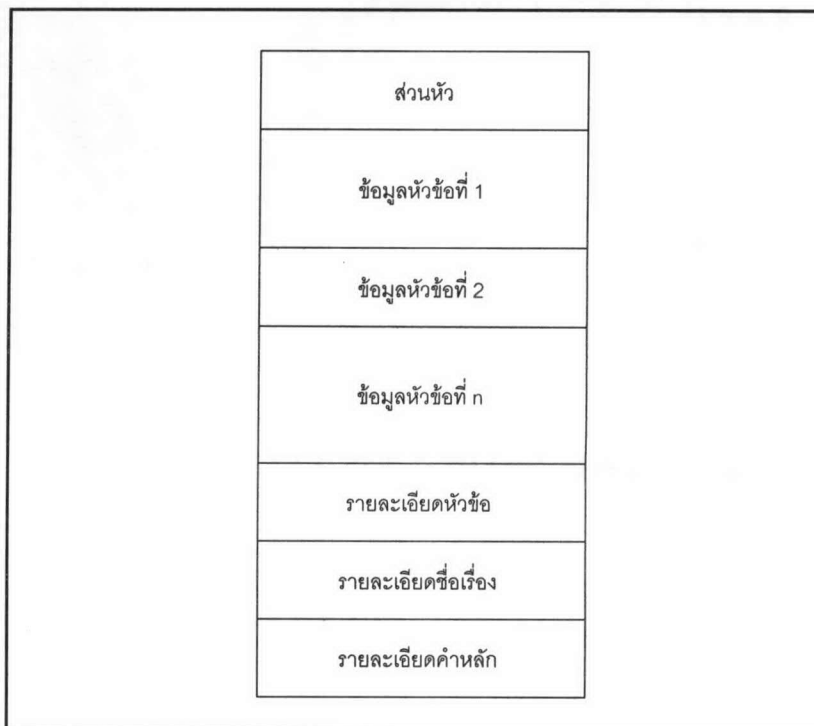
นอกจากนี้ ข้อมูลในแฟ้มข้อมูลสามารถใช้รูปแบบของฟอนต์ต่าง ๆ ได้ ได้แก่ ชื่อฟอนต์ ขนาด สี ตัวหนา ตัวเอียง และสามารถใส่รูปภาพในรูปแบบบิตแมพ (.bmp) ทั้งนี้ ถ้าแฟ้มข้อมูลมีรูปภาพ และ/หรือ เสียง ต้องนำแฟ้มข้อมูลรูปภาพ (.bmp) และแฟ้มข้อมูลเสียง (.wav) ไปใช้กับเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ด้วยเสมอในสารบบ (directory) เดียวกับที่กำหนดในแฟ้มข้อมูลนำเข้า



รูปที่ 3-2 ตัวอย่างเพิ่มข้อมูลนำเข้า

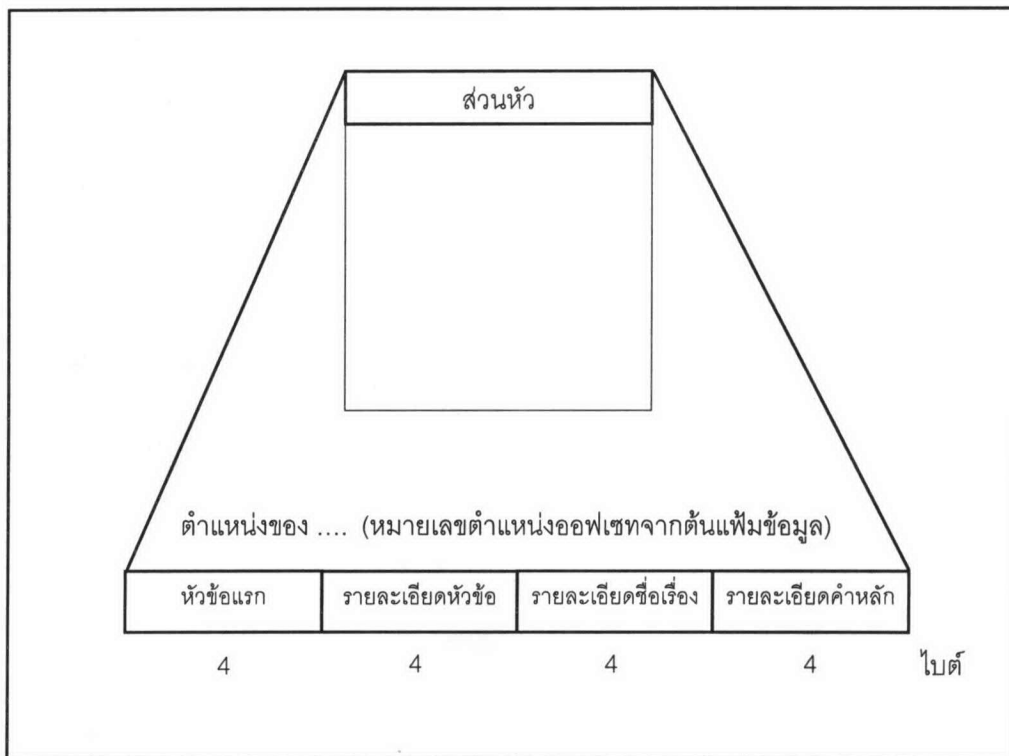
3.2 เพิ่มข้อมูลผลลัพธ์

โปรแกรมตัวแปลจะเขียนเพิ่มข้อมูลผลลัพธ์ โดยให้ชื่อเดียวกับเพิ่มข้อมูลนำเข้า และให้นามสกุล .hdv (HyperDocument Viewer file format) ที่สรวบเดียวกันกับเพิ่มข้อมูลนำเข้า



รูปที่ 3-3 โครงสร้างเพิ่มข้อมูลผลลัพธ์

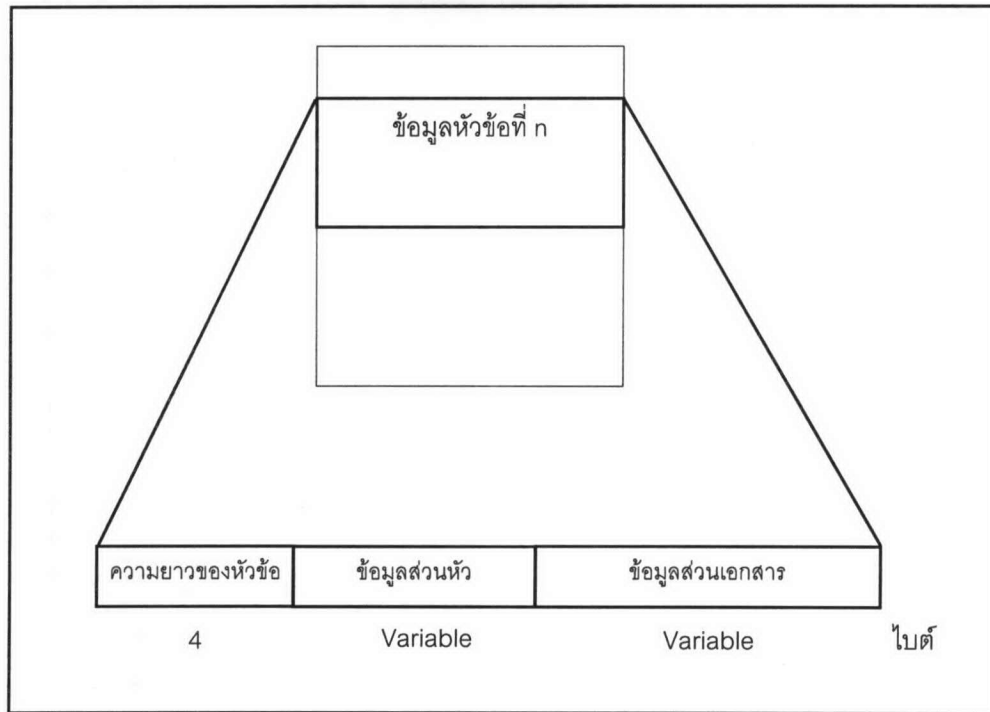
1. ส่วนหัว ประกอบด้วย 16 ไบต์ โดยแบ่งเป็น 4 ส่วน ๆ ละ 4 ไบต์ ในแต่ละส่วนเก็บหมายเลขตำแหน่งที่ชี้ไปยังข้อมูลของหัวข้อแรก ข้อมูลรายละเอียดหัวข้อ ข้อมูลรายละเอียดชื่อเรื่อง และข้อมูลรายละเอียดคำหลัก ตามลำดับ



รูปที่ 3-4 โครงสร้างส่วนหัวของแฟ้มข้อมูลผลลัพธ์

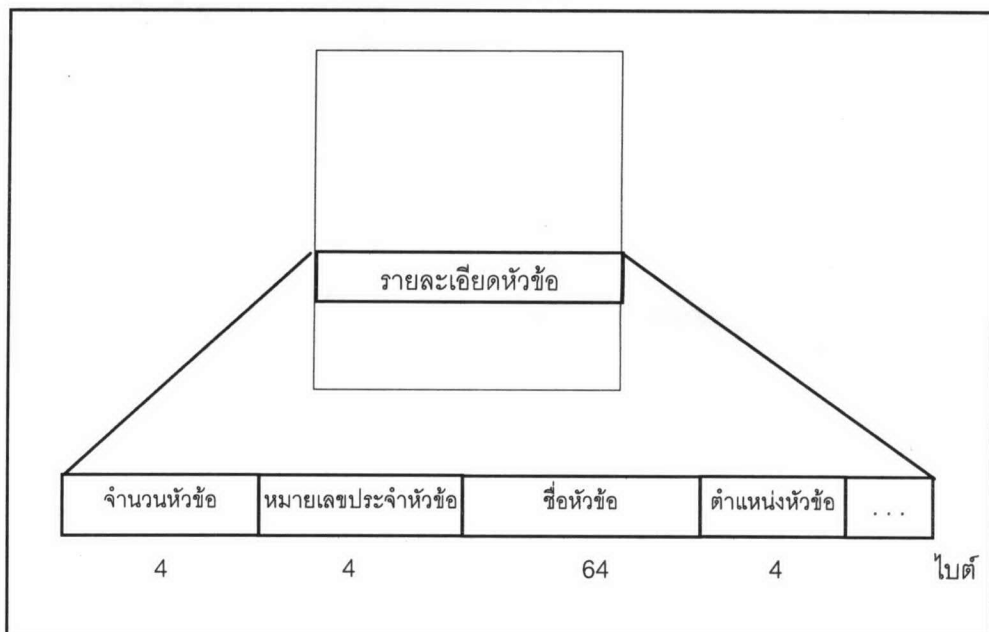
การใช้เนื้อที่เก็บ 4 ไบต์ ทำให้สามารถเก็บตำแหน่งได้สูงสุดถึง 2^{32} หรือ 4,294,967,295 ซึ่งหมายถึงแฟ้มข้อมูลผลลัพธ์สามารถมีขนาดใหญ่ที่สุดได้ 4,294,967,295 ไบต์

2. ข้อมูลหัวข้อที่ k เป็นพื้นที่ที่ใช้เก็บข้อมูลของแต่ละหัวข้อเริ่มจากหัวข้อที่ 1 ไปจนครบหัวข้อที่มี โดยที่ 4 ไบต์แรกของส่วนนี้จะเก็บความยาวของข้อมูลในหัวข้อ ซึ่งจะทำให้สามารถใช้เนื้อที่ในการเก็บข้อมูลตามจำนวนข้อมูลที่มีอยู่จริง ทั้งนี้ ข้อมูลของหัวข้อ จะประกอบด้วยส่วนที่เป็นข้อมูลส่วนหัว (RTF Header) และข้อมูลส่วนเอกสาร (RTF Document) เพื่อให้โปรแกรมเครื่องมือการแสดงผลข้อมูลไฮเปอร์ดอกคิวเมนต์สามารถอ่านข้อมูลทั้งสองส่วนไปใช้ในการกำหนดสภาพแวดล้อมและใช้ในการแสดงผลข้อมูลของหัวข้อ



รูปที่ 3-5 โครงสร้างส่วนข้อมูลของแฟ้มข้อมูลผลลัพธ์

3. รายละเอียดหัวข้อ ส่วนนี้จะเก็บจำนวนหัวข้อ และข้อมูลเกี่ยวกับหัวข้อที่มีในระบบ รวมถึงตำแหน่งของหัวข้อ โดยตำแหน่งที่ 1-4 จะเก็บจำนวนหัวข้อ จากนั้นจะเป็นโครงสร้างของแต่ละหัวข้อ คือ หมายเลขประจำหัวข้อ ชื่อหัวข้อ และตำแหน่งเริ่มต้นของหัวข้อ ซึ่งส่วนที่เป็นโครงสร้างของหัวข้อจะเรียงต่อกันไปจนครบจำนวนหัวข้อ



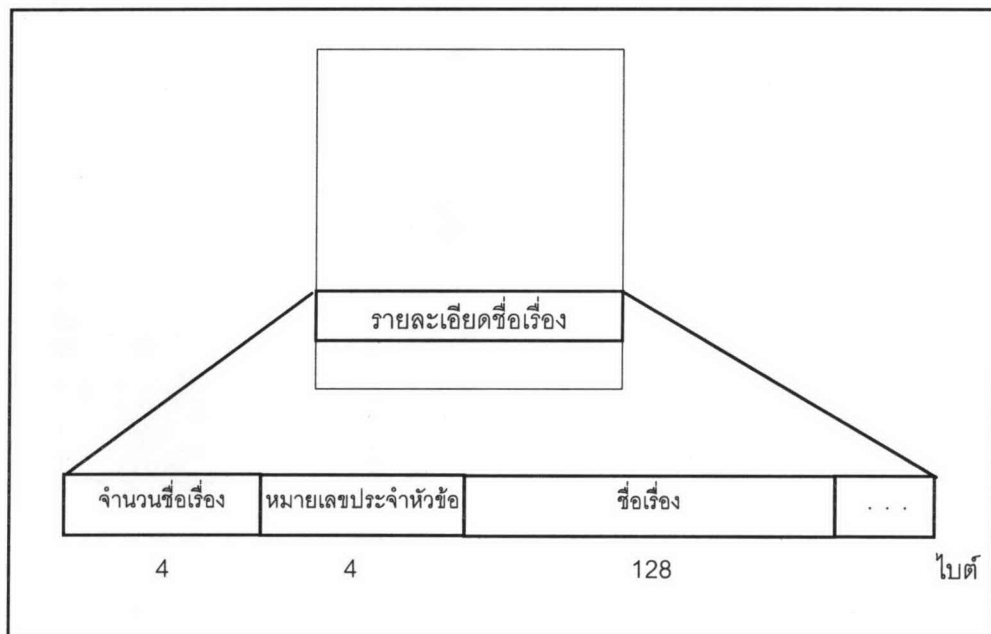
รูปที่ 3-6 โครงสร้างรายละเอียดหัวข้อของแฟ้มข้อมูลผลลัพธ์

หมายเลขประจำหัวข้อ โปรแกรมตัวแปลจะกำหนดเรียงตามลำดับให้หัวข้อที่พบในแฟ้มข้อมูลนำเข้า ใช้เนื้อที่ในการเก็บ 4 ไบต์ ทำให้สามารถมีหัวข้อได้ถึง 4,294,967,295 หัวข้อ

ชื่อหัวข้อ ใช้เนื้อที่ในการเก็บขนาด 64 ไบต์เสมอ แม้ว่าชื่อจะยาวนานน้อยกว่า 64 ตัวอักษรก็ตาม

ตำแหน่ง เป็นตำแหน่งเริ่มต้นของหัวข้อนั้นในแฟ้มข้อมูลผลลัพธ์ เพื่อให้โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ด็อกคิวเมนต์ สามารถเริ่มอ่านข้อมูลที่ตำแหน่งนี้ได้โดยตรง

4. รายละเอียดชื่อเรื่อง ส่วนนี้จะเก็บจำนวนชื่อเรื่อง และข้อมูลเกี่ยวกับชื่อเรื่องที่มีในระบบ บางหัวข้ออาจไม่มีชื่อเรื่อง ทำให้ชื่อเรื่องอาจมีจำนวนน้อยกว่าหรือเท่ากับจำนวนหัวข้อ โดยตำแหน่งที่ 1-4 จะเก็บจำนวนชื่อเรื่อง จากนั้นจะเป็นโครงสร้างของแต่ละชื่อเรื่อง คือ หมายเลขประจำหัวข้อของชื่อเรื่อง และชื่อเรื่อง ส่วนที่เป็นโครงสร้างของชื่อเรื่องจะเรียงต่อกันไปจนครบจำนวนชื่อเรื่อง

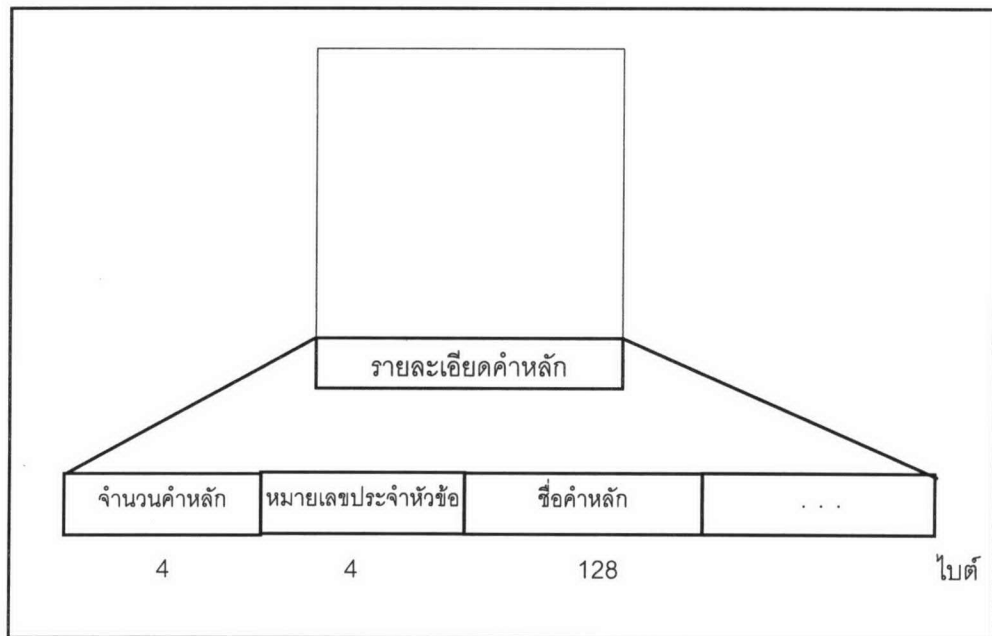


รูปที่ 3-7 โครงสร้างรายละเอียดชื่อเรื่องของแฟ้มข้อมูลผลลัพธ์

หมายเลขประจำหัวข้อ เป็นหมายเลขของหัวข้อที่เป็นของชื่อเรื่องนั้น เพื่อให้โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ด็อกคิวเมนต์ สามารถทราบหมายเลขหัวข้อ และหาตำแหน่งของหัวข้อได้ในที่สุด

ชื่อเรื่อง ใช้เนื้อที่ในการเก็บขนาด 128 ไบต์เสมอ แม้ว่าชื่อจะยาวนานน้อยกว่า 128 ตัวอักษรก็ตาม

5. รายละเอียดคำหลัก ส่วนนี้จะเก็บจำนวนคำหลัก และข้อมูลเกี่ยวกับคำหลักที่มีในระบบ โดยตำแหน่งที่ 1-4 จะเก็บจำนวนคำหลัก จากนั้นจะเป็นโครงสร้างของแต่ละคำหลัก คือ หมายเลขประจำหัวข้อของคำหลัก และคำหลัก ส่วนที่เป็นโครงสร้างของคำหลักจะเรียงต่อกันไปจนครบจำนวนคำหลัก



รูปที่ 3-8 โครงสร้างรายละเอียดคำหลักของแฟ้มข้อมูลผลลัพธ์

หมายเลขประจำหัวข้อ เป็นหมายเลขของ หัวข้อที่เป็นของคำหลัก เพื่อให้โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ สามารถทราบหมายเลขหัวข้อ และหาตำแหน่งข้อมูลของหัวข้อได้ในที่สุด

ชื่อคำหลัก ใช้เนื้อที่ในการเก็บขนาด 128 ไบต์เสมอ แม้ว่าชื่อจะยาวน้อยกว่า 128 ตัวอักษรก็ตาม

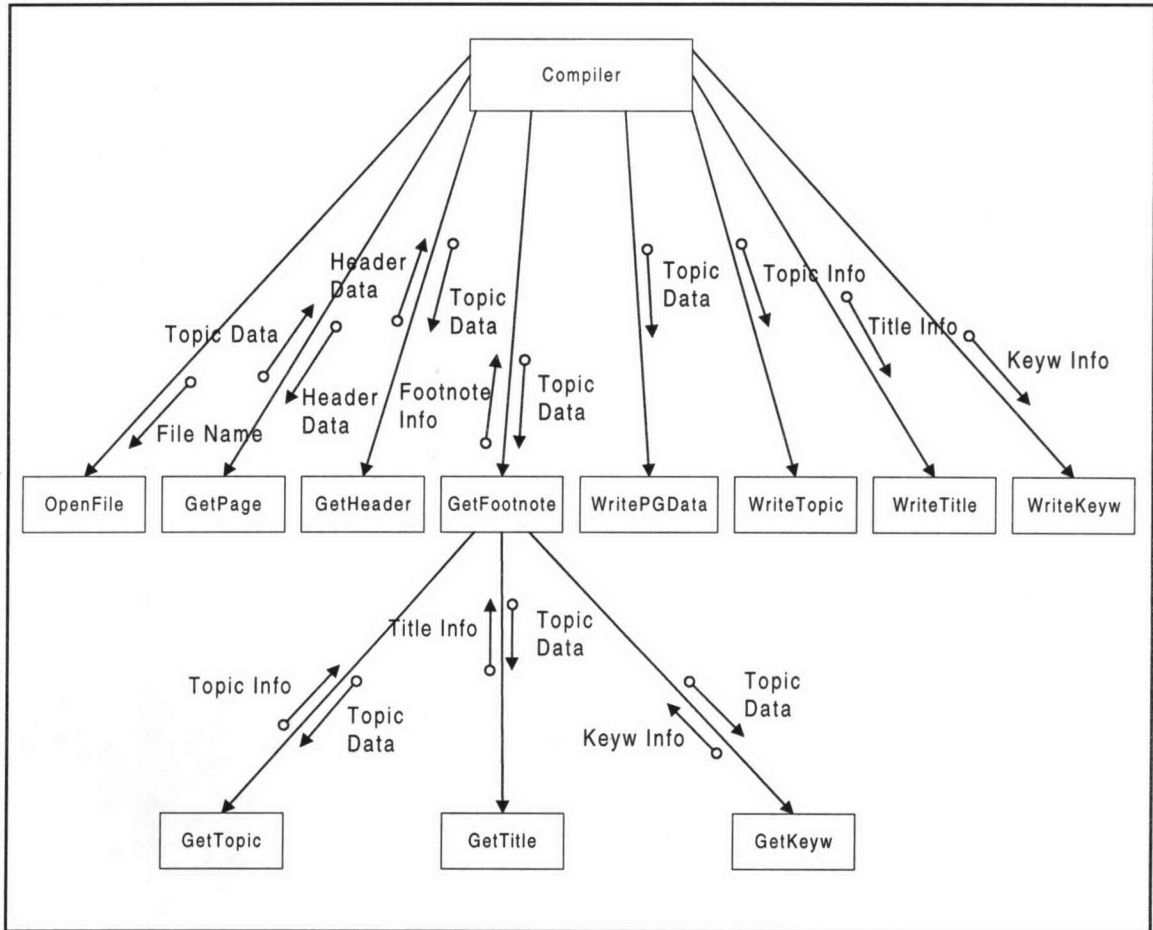
3.3 โปรแกรมตัวแปล

ตัวแปลมีหน้าที่อ่านแฟ้มข้อมูลนำเข้า และแยกข้อมูลที่เป็นเชิงอรรถ # \$ และ K ซึ่งได้แก่ ชื่อหัวข้อ ชื่อเรื่อง และคำหลักของแต่ละหัวข้อ เพื่อเก็บข้อมูลเหล่านี้พร้อมกับตำแหน่งของแต่ละหัวข้อ

ตำแหน่ง หมายถึงตำแหน่งที่นับเริ่มต้น 0 (ศูนย์) ที่จุดเริ่มต้นของแฟ้มข้อมูลนำเข้าในหน่วยความจำสำรอง และตำแหน่ง 1 ที่ไบต์แรกของแฟ้มข้อมูลนำเข้า เมื่อโปรแกรมตัวแปลได้ข้อมูลเหล่านี้แล้ว จะเขียนลงในแฟ้มข้อมูลผลลัพธ์ตามรูปแบบที่กำหนดในหัวข้อต่อไป

โปรแกรมตัวแปล จะต้องอ่านแฟ้มข้อมูลรูปแบบ RTF ซึ่งประกอบด้วย คำควบคุม (control word) ที่มีความหมายต่างกันไป โปรแกรมตัวแปลต้องสามารถเข้าใจและแยกแยะข้อมูลตามที่คำควบคุมกำกับอยู่

การออกแบบโปรแกรม สามารถอธิบายโดยใช้ Structure Chart และรายละเอียดการทำงานของแต่ละโมดูล ดังนี้



รูปที่ 3-9 Structure Chart ของโปรแกรมตัวแปล

1. เปิดเพิ่มข้อมูลนำเข้า (OpenFile) เพื่อเป็นข้อมูลในการทำงานของโปรแกรมตัวแปล
2. อ่านข้อมูลของแต่ละหัวข้อ (GetPage) อ่านข้อมูลขึ้นมาไว้ในตัวแปร ในที่นี้เรียกว่า ตัวแปรข้อมูล คราวละ 1 หัวข้อ โดยการอ่านครั้งแรก จะประกอบด้วย header และหัวข้อที่ 1 และสำหรับการอ่านครั้งถัดไปจะมีเพียงหัวข้อเท่านั้น
3. เก็บข้อมูลส่วนหัว (GetHeader) อ่านข้อมูลจากตัวแปรข้อมูล และตรวจสอบ RTF Control word เฉพาะที่เป็น header และเก็บไว้ในตัวแปรอีกชุดหนึ่ง ในที่นี้เรียกว่า ตัวแปรส่วนหัว

4. เก็บข้อมูลรายละเอียดหัวข้อ (GetTopic) อ่านข้อมูลจากตัวแปรข้อมูล ส่วนที่เหลือจากส่วนหัว โดยโปรแกรมตัวแปลจะทำงานนี้ ก็ต่อเมื่อพบเชิงอรรถที่เป็นเครื่องหมาย # โดยจะเก็บหมายเลขประจำหัวข้อ และชื่อหัวข้อ ไว้ในตัวแปรอีกชุดหนึ่ง ในที่นี้เรียกว่า ตัวแปรหัวข้อ

5. เก็บข้อมูลรายละเอียดชื่อเรื่อง (GetTitle) อ่านข้อมูลจากตัวแปรข้อมูล ส่วนที่เหลือจากเก็บข้อมูล รายละเอียดหัวข้อ โดยโปรแกรมตัวแปลจะทำงานนี้ ก็ต่อเมื่อพบเชิงอรรถที่เป็นเครื่องหมาย \$ โดยจะเก็บหมายเลขประจำหัวข้อ และชื่อเรื่อง ไว้ในตัวแปรอีกชุดหนึ่ง ในที่นี้เรียกว่า ตัวแปรชื่อเรื่อง

6. เก็บข้อมูลรายละเอียดคำหลัก (GetKeyw) อ่านข้อมูลจากตัวแปรข้อมูล ส่วนที่เหลือจากเก็บ ข้อมูลรายละเอียดชื่อเรื่อง โดยโปรแกรมตัวแปลจะทำงานนี้ ก็ต่อเมื่อพบเชิงอรรถที่เป็นเครื่องหมาย K โดยจะ เก็บหมายเลขประจำหัวข้อ และชื่อคำหลัก ไว้ในตัวแปรอีกชุดหนึ่ง ในที่นี้เรียกว่า ตัวแปรคำหลัก

7. เก็บข้อมูลของหัวข้อ และเขียนลงในแฟ้มข้อมูลผลลัพธ์ (WritePGData) อ่านข้อมูลจากตัวแปร ข้อมูล ส่วนที่เหลือจากเก็บข้อมูลรายละเอียดคำหลัก แล้วเขียนลงในแฟ้มข้อมูลผลลัพธ์ โดยคำนวณตำแหน่งที่ เริ่มเขียนของหัวข้อ โดย m คือลำดับของหัวข้อที่กำลังจะเขียน ดังนี้

$$\text{ตำแหน่งเริ่มต้น} = 16 + \text{ความยาวของหัวข้อที่ } 1 + \dots + \text{ความยาวของหัวข้อที่ } (m-1) + 4$$

การเขียนจะเขียนข้อมูลจากตัวแปรส่วนหัวก่อน แล้วจึงอ่านข้อมูลจากตัวแปรข้อมูลที่เหลือ เมื่อเขียน จนหมดข้อมูลจากตัวแปรข้อมูลแล้ว จะทราบความยาวของหัวข้อนั้น ให้นำเอาความยาวบวกด้วย 4 แล้วเขียน ลงในตำแหน่งเริ่มต้นลบด้วย 4 จากนั้นเก็บตำแหน่งนี้ไว้ในตัวแปรหัวข้อด้วย

8. เขียนข้อมูลรายละเอียดหัวข้อลงแฟ้มข้อมูลผลลัพธ์ (WriteTopic) เมื่อเขียนข้อมูลของหัวข้อจน ครบทุกหัวข้อแล้ว จะได้ตัวแปรหัวข้อ ตัวแปรชื่อเรื่อง และ ตัวแปรคำหลักของแต่ละหัวข้อ ในขั้นนี้ให้เขียนจำนวน หัวข้อ และข้อมูลจากตัวแปรหัวข้อของแต่ละหัวข้อเรียงต่อกันไป โดยตัวแปรหัวข้อของแต่ละหัวข้อ ประกอบด้วย หมายเลขประจำหัวข้อ ชื่อหัวข้อ และ ตำแหน่งที่เก็บข้อมูลหัวข้อ

9. เขียนข้อมูลรายละเอียดชื่อเรื่องลงแฟ้มข้อมูลผลลัพธ์ (WriteTitle) เขียนจำนวนชื่อเรื่อง และ ข้อมูลจากตัวแปรชื่อเรื่องเรียงต่อกันไป โดยตัวแปรชื่อเรื่องของแต่ละหัวข้อจะประกอบด้วย หมายเลขประจำหัว ข้อ และ ชื่อเรื่องลงแฟ้มข้อมูลผลลัพธ์

10. เขียนข้อมูลรายละเอียดคำหลัก (WriteKeyw) เขียนจำนวนคำหลัก และข้อมูลจากตัวแปรคำ หลักเรียงต่อกันไป โดยตัวแปรคำหลักแต่ละชุด จะประกอบด้วย หมายเลขประจำหัวข้อ และ ชื่อคำหลัก

3.4 โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์

ในส่วนนี้เป็นโปรแกรมที่อ่านแฟ้มข้อมูลผลลัพธ์ที่ผ่านการแปลจากตัวแปลแล้ว โดยโปรแกรมมีความสามารถในการแสดงข้อมูล ดังนี้

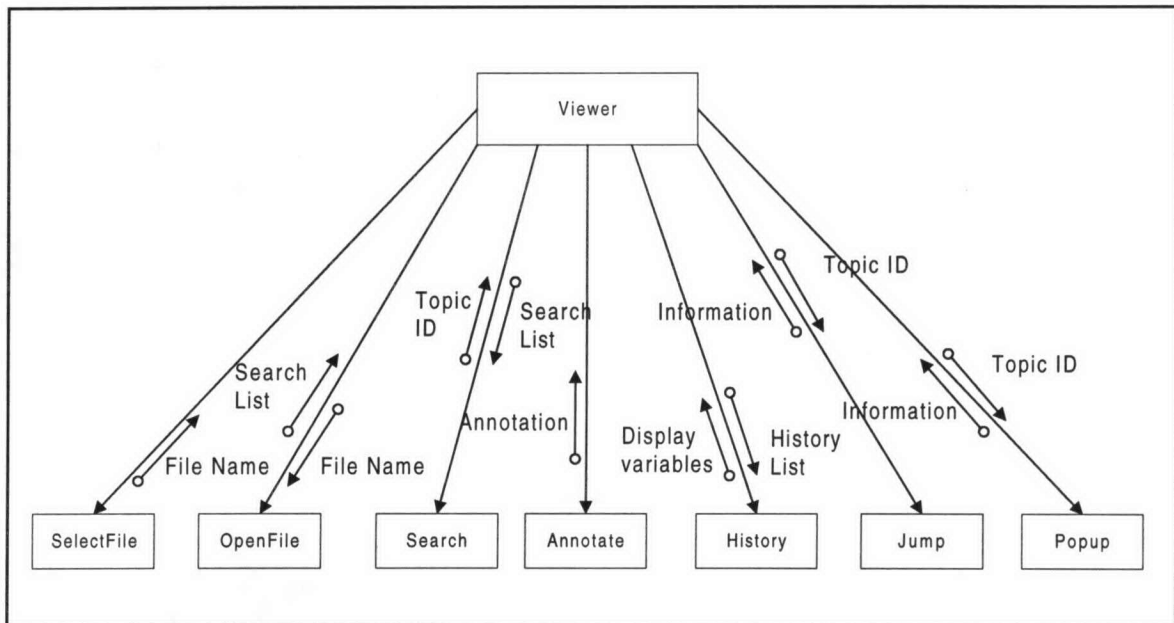
1. สามารถแสดงข้อมูลตัวอักษร ด้วยรูปแบบ ขนาด และสีที่ต่างกันได้
2. สามารถแสดงรูปภาพชนิดบิตแมพ (.bmp) ที่สร้างภายใต้วินโดว และประมวลผลเสียง (.wav) ได้
3. สามารถกระโดด (jump) ไปยังตำแหน่งข้อมูลใด ๆ และแสดงบนหน้าตาต่างเดียวกัน
4. สามารถเจาะลึกข้อมูลที่ต้องการโดยแสดงบนจอซ้อนแบบผุดขึ้น
5. สามารถค้นหาคำหรือข้อมูลที่ต้องการจากการค้นหาคำหลัก (keyword search)
6. สามารถค้นหาข้อมูลที่ต้องการจากหัวข้อ (topic) ที่ปรากฏในสารบัญ (content)
7. สามารถให้ผู้ใช้เขียนหรือเรียกแสดงข้อความเตือนความจำ (annotation) ให้กับข้อมูลที่ต้องการ
8. สามารถจดจำตำแหน่งที่ทำงานผ่านมาแล้วได้ และผู้ใช้สามารถย้อนกลับไปได้โดยไม่ต้องเรียงตามลำดับ

ในการออกแบบส่วนเชื่อมประสานผู้ใช้ของระบบ จะเป็นเมนูที่ด้านบนของหน้าต่าง โดยเมนูมีหัวข้อหลักและหัวข้อย่อย ให้ผู้ใช้เลื่อนเมาส์ไปกดบนหัวข้อที่ต้องการทำงาน

แฟ้มข้อมูล	เครื่องมือ	ไปยัง	คำอธิบาย
เปิด	ค้นหา ...	สารบัญ	เกี่ยวกับโปรแกรม
ออกจากการทำงาน	จดบันทึก	History ... ไปข้างหน้า ย้อนกลับ	

รูปที่ 3-10 เมนูหลักและเมนูย่อยของโปรแกรมแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์

การออกแบบโปรแกรม สามารถอธิบายโดยใช้ Structure Chart และรายละเอียดการทำงานของแต่ละโมดูล ดังนี้



รูปที่ 3-11 Structure Chart ของโปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์

1. เลือกเพิ่มข้อมูลนำเข้า (SelectFile) ทำหน้าที่เปิดหน้าต่างที่แสดงรายชื่อโดรพี ชื่อสารบบและชื่อเพิ่มข้อมูล เพื่อให้ผู้ใช้เลือกเพิ่มข้อมูลที่มีนามสกุล .hdv ซึ่งเป็นผลลัพธ์จากการทำงานของโปรแกรมตัวแปล
2. เปิดเพิ่มข้อมูลนำเข้า (OpenFile) ทำหน้าที่อ่านข้อมูลจากเพิ่มข้อมูลที่มีนามสกุล .hdv ที่ผู้ใช้เลือกเพื่อเก็บข้อมูลการเชื่อมโยง และแสดงข้อมูลของหัวข้อแรก โปรแกรม OpenFile มีขั้นตอนการทำงาน ดังนี้
 - 2.1 เปิดเพิ่มข้อมูลที่เลือก ถ้า เปิดได้ก็จะทำข้อ 2.2 ถ้าเปิดไม่ได้ให้จบการทำงาน
 - 2.2 กำหนดค่าเริ่มต้นของระบบ ดังนี้
 - 2.2.1 ลำดับที่ของ history ให้มีค่าเป็น 0
 - 2.2.2 ฟอนต์ในอาร์เรย์ที่ 0 ให้มีค่าเป็น "AngsanaUPC" กรณีที่ไม่มีฟอนต์ตามที่ระบุในข้อมูลให้ใช้ฟอนต์นี้แทน
 - 2.2.3 สีในอาร์เรย์ที่ 0 ให้มีค่าเป็นสีเขียว กรณีที่ไม่มีสีตามที่ระบุในข้อมูลให้ใช้สีนี้แทน
 - 2.3 อ่านข้อมูลจากเพิ่มข้อมูลเกี่ยวกับชื่อและตำแหน่งของหัวข้อ ดังนี้
 - 2.3.1 อ่านค่าที่ตำแหน่งที่ 4-7 จะได้ตำแหน่งเริ่มต้นของรายละเอียดเกี่ยวกับหัวข้อ สมมติว่าเท่ากับ xPos
 - 2.3.2 อ่านค่าที่ตำแหน่งที่ xPos ถึง (xPos+4) จะได้จำนวนของหัวข้อที่มีในระบบ เก็บไว้ในตัวแปรชื่อ totalTopic

- 2.3.3 อ่านค่าที่ตำแหน่งถัดไป 4 ไบต์ จะได้หมายเลขประจำหัวข้อ เก็บไว้ในตัวแปรชื่อ `vtopic(1).id`
- 2.3.4 อ่านค่าที่ตำแหน่งถัดไป 64 ไบต์ จะได้ชื่อหัวข้อ เก็บไว้ในตัวแปรชื่อ `vtopic(1).tName`
- 2.3.5 อ่านค่าที่ตำแหน่งถัดไป 4 ไบต์ จะได้ตำแหน่งเริ่มต้นของข้อมูลของหัวข้อ เก็บไว้ในตัวแปรชื่อ `vtopic(1).PosPg`
- 2.3.6 อ่านค่าข้อมูลตามข้อ 2.3.3 - 2.3.5 จนครบตามจำนวนหัวข้อ โดยหมายเลขในอาร์เรย์ให้เพิ่มคราวละ 1
- 2.4 อ่านข้อมูลจากแฟ้มข้อมูลเกี่ยวกับชื่อเรื่อง ดังนี้
 - 2.4.1 อ่านค่าที่ตำแหน่งที่ 8-11 จะได้ตำแหน่งเริ่มต้นของชื่อเรื่อง สมมติว่าเท่ากับ `xPos`
 - 2.4.2 อ่านค่าที่ตำแหน่งที่ `xPos` ถึง `(xPos+4)` จะได้จำนวนของชื่อเรื่องที่มีในระบบ เก็บไว้ในตัวแปรชื่อ `totalTitle`
 - 2.4.3 อ่านค่าที่ตำแหน่งถัดไป 4 ไบต์ จะได้หมายเลขประจำหัวข้อของ ชื่อเรื่อง เก็บไว้ในตัวแปรชื่อ `vtitle(1).id`
 - 2.4.4 อ่านค่าที่ตำแหน่งถัดไป 128 ไบต์ จะได้ชื่อเรื่อง เก็บไว้ในตัวแปรชื่อ `vtitle(1).tName`
 - 2.4.5 อ่านค่าข้อมูลตามข้อ 2.4.3 - 2.3.4 จนครบตามจำนวนชื่อเรื่อง โดยหมายเลขในอาร์เรย์ให้เพิ่มคราวละ 1
- 2.5 อ่านข้อมูลจากแฟ้มข้อมูลเกี่ยวกับคำหลัก ดังนี้
 - 2.5.1 อ่านค่าที่ตำแหน่งที่ 12 -15 จะได้ตำแหน่งเริ่มต้นของคำหลัก สมมติว่าเท่ากับ `xPos`
 - 2.5.2 อ่านค่าที่ตำแหน่งที่ `xPos` ถึง `(xPos+4)` จะได้จำนวนของคำหลักที่มีในระบบ เก็บไว้ในตัวแปรชื่อ `totalKeyW`
 - 2.5.3 อ่านค่าที่ตำแหน่งถัดไป 4 ไบต์ จะได้หมายเลขประจำหัวข้อของคำหลัก เก็บไว้ในตัวแปรชื่อ `vkeyw(1).id`
 - 2.5.4 อ่านค่าที่ตำแหน่งถัดไป 128 ไบต์ จะได้ชื่อเรื่อง เก็บไว้ในตัวแปรชื่อ `vkeyw(1).kName`
 - 2.5.5 อ่านค่าข้อมูลตามข้อ 2.5.3 - 2.5.4 จนครบตามจำนวนคำหลัก โดยหมายเลขในอาร์เรย์ให้เพิ่มคราวละ 1
- 2.6 สร้างรายชื่อคำหลักเก็บไว้ในตัวแปร โดยต้องเลือกเฉพาะคำหลักที่ไม่ซ้ำกันเท่านั้น ไว้ในตัวแปรชื่อ `keywList(n)` โดย `n` เป็นลำดับที่ของคำหลักในรายชื่อคำหลัก เริ่มต้นที่ 0 และเก็บจำนวนคำหลักไว้ที่ตัวแปรชื่อ `keywListno`
- 2.7 แสดงข้อมูลของหัวข้อแรกบนหน้าต่าง โดยการเรียกใช้โปรแกรมการกระโดด (Jump) ซึ่งจะกล่าวถึงต่อไป

3. ค้นหาหัวข้อ (Search) ทำหน้าที่แสดงรายชื่อคำหลัก ให้ผู้ใช้เลือก และกระโดดไปยังหัวข้อที่ผู้ใช้เลือก โปรแกรม Search มีขั้นตอนการทำงาน ดังนี้

- 3.1 เปิดหน้าต่างใหม่สำหรับเตรียมแสดงรายชื่อคำหลัก
- 3.2 เพิ่มชื่อคำหลักที่เก็บอยู่ในตัวแปร `keywList(n)` ให้กับ รายชื่อ คำหลัก จนครบตามจำนวนที่เก็บในตัวแปร `keywListNo`
- 3.3 เมื่อผู้ใช้เลือกคำหลัก ต้องตรวจสอบว่าคำหลักนั้นเป็นของหัวข้อเดียว หรือหลายหัวข้อ
 - 3.3.1 กรณีเป็นของหัวข้อเดียว ให้เรียกโปรแกรม Jump ไปยังหัวข้อที่ระบุในหมายเลขประจำ หัวข้อของคำหลักนั้น
 - 3.3.2 กรณีเป็นของหลายหัวข้อ ให้เรียกหน้าต่างแสดงรายชื่อชื่อเรื่อง ของคำหลักนั้น เพื่อให้ผู้ใช้เลือกชื่อเรื่อง แล้วจึงเรียกโปรแกรม Jump ไปยังหัวข้อที่ระบุในหมายเลขประจำหัวข้อของชื่อเรื่องนั้น

4. บันทึกข้อความเตือนความจำ (Annotate) ทำหน้าที่แสดงหน้าต่างให้ผู้ใช้พิมพ์ข้อความ หรือแก้ไขข้อความเดิม ที่ต้องการบันทึกลงในหัวข้อที่แสดงอยู่บนหน้าต่าง และโปรแกรมจะบันทึกข้อความนั้นไว้ จนกว่าผู้ใช้จะลบออก ข้อความที่บันทึกของแต่ละหัวข้อจะเก็บลงในแฟ้มข้อมูลในราก `\hyperdoc\` มีชื่อว่า `tN` โดยที่ `N` คือหมายเลขประจำหัวข้อ โปรแกรม Annotate มีขั้นตอนการทำงาน ดังนี้

- 4.1 เปิดหน้าต่างใหม่สำหรับให้ผู้ใช้บันทึก แก้ไข หรือ ลบข้อความ
- 4.2 ผู้ใช้สามารถพิมพ์ หรือแก้ไขข้อความได้ ข้อความจะเป็นข้อความแบบบรรทัดเดียว
- 4.3 เมื่อผู้ใช้บันทึก หรือแก้ไขข้อความแล้ว ให้ตรวจสอบความยาวของข้อความ ดังนี้
 - 4.2.1 ถ้าความยาวเท่ากับ 0 ให้ลบแฟ้มข้อมูลที่เก็บข้อความ
 - 4.2.2 ถ้าความยาวมากกว่า 0 ให้เขียนข้อความนั้นลงแฟ้มข้อมูลของหัวข้อนั้น
- 4.4 แสดงข้อความที่ด้านบนของข้อมูลหัวข้อนั้น

5. เรียกดูหัวข้อที่เคยเรียกดูแล้ว (History และ Goto) ทำหน้าที่ให้ผู้ใช้เรียกดูข้อมูลของหัวข้อที่เคยเรียกดูแล้ว ซึ่งจะทำงานได้เร็วกว่าการกระโดดไปใหม่

- 5.1 การทำงานของโปรแกรม History มีขั้นตอน ดังนี้
 - 5.1.1 เปิดหน้าต่างใหม่สำหรับเตรียมแสดงรายชื่อหัวข้อที่เคยเรียกดูแล้ว
 - 5.1.2 สร้างรายชื่อหัวข้อที่เคยเรียกดูแล้ว จากตัวแปรชื่อ `history(N).topicName` โดยที่ `N` คือลำดับที่ของหัวข้อในประวัติ เริ่มที่ 1 สำหรับจำนวนหัวข้อในประวัติจะเก็บอยู่ในตัวแปรชื่อ `historyLast`
 - 5.1.3 เมื่อผู้ใช้เลือกชื่อหัวข้อที่ต้องการดูข้อมูล ให้ตรวจสอบว่าเป็นหัวข้อเดียวกับที่แสดงอยู่ขณะนั้นหรือไม่ ถ้าใช่ โปรแกรมจะไม่ทำอะไร ถ้าไม่ใช่ ให้อ่านค่าข้อมูลที่เก็บในชุดตัวแปรที่เกี่ยวข้องกับประวัติ แล้วแสดงข้อมูลของหัวข้อนั้น โดยใช้ข้อมูลจากตัวแปร ไม่ต้องอ่านจากแฟ้มข้อมูลใหม่

5.2 การทำงานของโปรแกรม Goto มีขั้นตอน ดังนี้

- 5.2.1 กรณี ไปหน้าสารบัญ ให้อ่านค่าจากชุดตัวแปรที่เกี่ยวกับประวัติชุดที่ 1 แล้วแสดงข้อมูลของหัวข้อแรกบนหน้าต่าง
- 5.2.2 กรณี ไปข้างหน้า ให้ตรวจสอบว่าหัวข้อที่แสดงในขณะนั้นเป็นหัวข้อสุดท้ายในประวัติหรือไม่ ถ้าใช่ โปรแกรมไม่ทำงานต่อ ถ้าไม่ใช่ ให้อ่านค่าจากชุดตัวแปรที่เกี่ยวกับประวัติชุดถัดไป แล้วแสดงข้อมูลของหัวข้อนั้นบนหน้าต่าง
- 5.2.3 กรณี ย้อนกลับ ให้ตรวจสอบว่าหัวข้อที่แสดงในขณะนั้นเป็นหัวข้อแรกหรือไม่ ถ้าใช่ โปรแกรมไม่ทำงานต่อ ถ้าไม่ใช่ ให้อ่านค่าจากชุดตัวแปรที่เกี่ยวกับประวัติชุดก่อนหน้า แล้วแสดงข้อมูลของหัวข้อนั้นบนหน้าต่าง

6. การกระโดด (Jump) ทำหน้าที่กระโดดไปยังหัวข้อที่เลือก เพื่อแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ของหัวข้อนั้นบนหน้าต่างของระบบ โปรแกรม Jump มีขั้นตอนการทำงาน ดังนี้

- 6.1 ตรวจสอบว่าหัวข้อที่ต้องการกระโดดไป เป็นหัวข้อเดียวกับที่แสดงอยู่ในขณะนี้หรือไม่ ถ้าใช่ โปรแกรมไม่ทำงานต่อ ถ้าไม่ใช่ ทำข้อถัดไป
- 6.2 อ่านค่าจากแฟ้มข้อมูลที่ตำแหน่งตามที่ระบุในตัวแปรชื่อ vtopic(N).PosPg โดยที่ N คือหมายเลขประจำหัวข้อ ให้อ่าน 4 ไบต์ จะได้ความยาวของหัวข้อนั้น หน่วยเป็นไบต์ ให้เก็บในตัวแปรชื่อ topicLength
- 6.3 อ่านข้อมูลของหัวข้อนั้นจากแฟ้มข้อมูลตามความยาวที่อ่านได้ ขึ้นมาเก็บไว้ในหน่วยความจำ ข้อมูลนี้จะเป็นรูปแบบของ RTF ประกอบด้วยส่วนหัว และส่วนเอกสาร
- 6.4 ทำการประมวลผลข้อมูลของหัวข้อจากหน่วยความจำ ในส่วนหัว และส่วนเอกสาร เพื่อเก็บคุณลักษณะต่าง ๆ ที่ต้องใช้ในการแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ โดยมีหลักการประมวลผล เป็นดังนี้
 - 6.4.1 การประมวลผล 1 ครั้งจะอ่านข้อมูลจากตำแหน่งแรก หรือตำแหน่งที่อ่านเสร็จครั้งก่อน เมื่อได้ข้อมูล 1 ชุดจะส่งข้อมูลชุดนั้นออกมาให้โปรแกรมทำงานต่อ
 - 6.4.2 การประมวลผลจะแยกประเภทข้อมูลเป็น 3 ประเภท คือ คำควบคุม อักษรภาษาอังกฤษ และอักษรภาษาไทย
 - 6.4.3 ชุดข้อมูลที่เป็น คำควบคุม จะประกอบด้วย ชื่อคำควบคุม และพารามิเตอร์
 - 6.4.4 ชุดข้อมูลที่เป็น ภาษาอังกฤษ จะประกอบด้วย ข้อความภาษาอังกฤษแต่ละคำ รวมอักษรว่างท้ายคำ 1 ตัว
 - 6.4.5 ชุดข้อความที่เป็น ภาษาไทย จะประกอบด้วย ข้อความภาษาไทยแต่ละประโยค รวมตัวอักษรว่างท้ายคำ 1 ตัว
- 6.5 การประมวลผลในส่วนหัว ให้เก็บคุณลักษณะ ดังนี้
 - 6.5.1 ฟอนต์ที่ใช้ในกรณีหาฟอนต์ที่ต้องการไม่พบ เก็บหมายเลขประจำฟอนต์ไว้ในตัวแปรชื่อ DefFont

- 6.5.2 รายการเกี่ยวกับฟอนต์ เก็บชื่อฟอนต์ในตัวแปรชื่อ $f(N)$ โดยที่ N คือ หมายเลขประจำฟอนต์
- 6.5.3 รายการเกี่ยวกับสี เก็บสีไว้ในตัวแปรชื่อ $c(N)$ โดยที่ N คือ หมายเลขประจำสี
- 6.6 การประมวลผลในส่วนเอกสาร ให้เก็บคุณลักษณะ ดังนี้
 - 6.6.1 ความกว้างและความยาวของข้อมูล ที่ต้องการแสดงบนหน้าต่าง
 - 6.6.2 ระยะเว้นว่างด้านซ้าย และระยะเว้นว่างด้านบน
 - 6.6.3 รูปแบบตัวอักษรสำหรับการแสดงบนหน้าต่าง ได้แก่ รูปแบบปกติ ตัวหนา และตัวเอียง แล้วเก็บตำแหน่งที่จะแสดงบนหน้าต่าง
 - 6.6.4 รูปแบบตัวอักษรที่เป็นพื้นที่สำคัญ ได้แก่
 - 6.6.4.1 ชิดเส้นใต้คู่ แสดงว่าเป็นข้อความสำหรับการกระโดด ให้เก็บข้อความที่ชิดเส้นใต้คู่จนครบทั้งหมด แล้วเก็บตำแหน่งของพื้นที่บริเวณที่จะแสดงบนหน้าต่าง เพื่อเปลี่ยนตัวซึ่งเป็นสัญลักษณ์ว่าเป็นพื้นที่สำคัญ นอกจากนี้ให้เปลี่ยนรูปแบบตัวอักษรเป็น รูปแบบปกติ ชิดเส้นใต้เดี่ยว และให้สีเป็นสีเขียว
 - 6.6.4.2 ชิดเส้นใต้เดี่ยว แสดงว่าเป็นข้อความสำหรับแสดงจอซ้อนแบบผุดขึ้น ให้เก็บข้อความที่ชิดเส้นใต้เดี่ยวจนครบทั้งหมด แล้วเก็บตำแหน่งของพื้นที่บริเวณที่จะแสดงบนหน้าต่าง เพื่อเปลี่ยนตัวซึ่งเป็นสัญลักษณ์ว่าเป็นพื้นที่สำคัญ นอกจากนี้ให้เปลี่ยนรูปแบบตัวอักษรเป็น รูปแบบปกติ ชิดเส้นใต้เดี่ยว และให้สีเป็นสีน้ำเงิน
 - 6.6.4.3 ชิดเส้นกลางข้อความ แสดงว่าเป็นข้อความสำหรับประมวลผลเสียง ให้เก็บข้อความที่ชิดเส้นกลางข้อความจนครบทั้งหมด แล้วเก็บตำแหน่งของพื้นที่บริเวณที่จะแสดงบนหน้าต่าง เพื่อเปลี่ยนตัวซึ่งเป็น สัญลักษณ์ว่าเป็นพื้นที่สำคัญ นอกจากนี้ให้เปลี่ยนรูปแบบตัวอักษรเป็น ตัวเอียง ชิดเส้นใต้เดี่ยวและให้สีเป็นสีเขียว
 - 6.6.5 รูปภาพ ให้เก็บพารไอยังแฟ้มข้อมูลบิตแมพ รวมทั้งตำแหน่งที่จะแสดงบนหน้าต่าง และยาวของพื้นที่ที่ต้องใช้
- 6.7 แสดงผลข้อมูลที่เป็นข้อความ และ รูปภาพ ตามคุณลักษณะและ ตำแหน่งที่เก็บไว้
- 6.8 ปรับค่าของสกรอลบาร์ทั้งแนวดิ่ง และแนวนอนให้สอดคล้องกับความกว้างและยาวของข้อมูลที่แสดง
- 6.9 เก็บหมายเลขประจำหัวข้อ และชื่อหัวข้อลงในชุดตัวแปรที่เกี่ยวกับประวัติ

7. การแสดงจอซ้อนแบบผุดขึ้น (Popup) ทำหน้าที่แสดงข้อมูลของหัวข้อที่เลือกบนจอซ้อนแบบผุดขึ้น และจอซ้อนจะปิดไปเมื่อมีการคลิกเมาส์บนพื้นที่ใด ๆ บนหน้าต่างของระบบ โปรแกรม Popup มีขั้นตอนการทำงาน ดังนี้

- 7.1 อ่านค่าจากแฟ้มข้อมูลที่ตำแหน่งตามที่ระบุในตัวแปรชื่อ vtopic(N).PosPg โดยที่ N คือหมายเลขประจำหัวข้อให้อ่าน 4 ไบต์ จะได้ความยาวของหัวข้อนั้น หน่วยเป็นไบต์ ให้เก็บในตัวแปรชื่อ topicLength
- 7.2 อ่านข้อมูลของหัวข้อนั้นจากแฟ้มข้อมูลตามความยาวที่อ่านได้ ขึ้นมาเก็บไว้ในหน่วยความจำ ข้อมูลนี้จะเป็นรูปแบบของ RTF ประกอบด้วยส่วนหัว และส่วนเอกสาร
- 7.3 ประมวลผลข้อมูลด้วยหลักการเดียวกับการกระโดด
- 7.4 กำหนดให้จอซ้อนแบบผุดขึ้นแสดงบนหน้าต่างของระบบ โดยตำแหน่งด้านซ้ายของหน้าต่างเป็น 100 และตำแหน่งด้านบนอยู่ที่ตำแหน่งที่เมาส์คลิก
- 7.5 ถ้ายค่าตัวแปรที่ได้จากการประมวลผลให้กับชุดตัวแปรของการแสดงข้อมูลบนจอซ้อนแบบผุดขึ้น
- 7.6 แสดงข้อมูลบนจอซ้อนแบบผุดขึ้น
- 7.7 อ่านค่าของชุดตัวแปรของข้อมูลที่เกี่ยวข้องประวัติของหัวข้อปัจจุบันที่แสดงบนหน้าต่างของระบบ เพื่อสามารถแสดงข้อมูลบนหน้าต่างของระบบได้ตามปกติ

3.5 การพัฒนาโปรแกรม

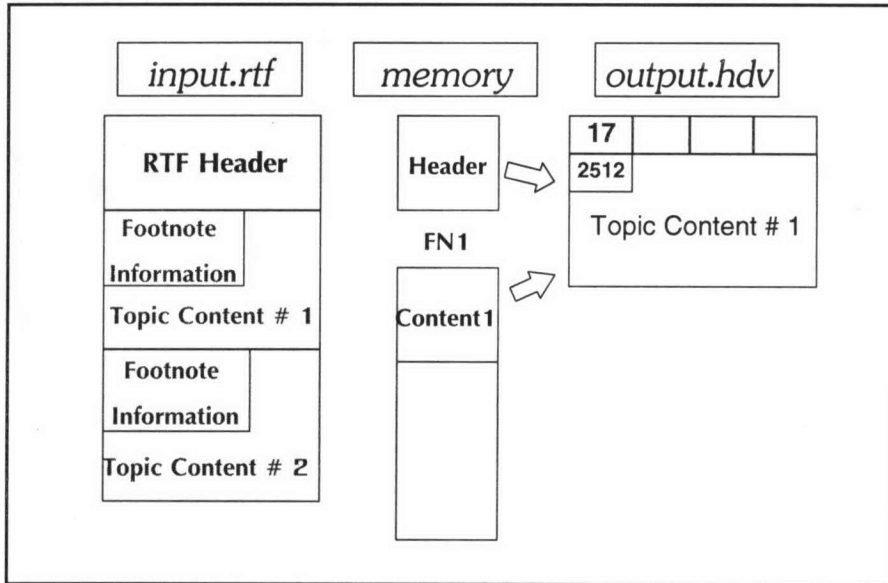
การพัฒนาโปรแกรม แบ่งเป็น 2 ส่วนเช่นเดียวกับการออกแบบ คือ การพัฒนาโปรแกรมตัวแปล และการพัฒนาโปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์

โปรแกรมตัวแปล พัฒนาด้วยภาษา Visual Basic 4.0 แล้วจัดทำเป็นแฟ้มประมวลผล (Executable File) เพื่อให้ผู้ใช้สามารถเรียกใช้โปรแกรมได้โดยง่าย

โปรแกรมเครื่องมือแสดงข้อมูลไฮเปอร์ดอกคิวเมนต์ พัฒนาด้วย Visual Basic 4.0 เช่นกัน โดยส่วนที่เป็นการประมวลผลข้อมูล RTF พัฒนาด้วย Visual C++ เพื่อผลเรื่องความเร็วในการประมวลผล นอกจากนี้โปรแกรมส่วนนี้ จะจัดทำเป็น OLE Server โดยมีฟังก์ชัน 2 ฟังก์ชัน คือ การเปิดแฟ้มข้อมูล และการกระโดด ให้นักพัฒนาโปรแกรมสามารถเรียกไปใช้งานได้

3.5.1 การพัฒนาโปรแกรมตัวแปล มีขั้นตอนดังนี้

- 3.5.1.1 อ่านข้อมูลเข้าสู่หน่วยความจำหลัก ได้แก่
 - ข้อมูลส่วนหัวของแฟ้มข้อมูลนำเข้า
 - ข้อมูลหัวข้อที่ 1 ประกอบด้วย ข้อมูลเชิงอรรถ และข้อมูลหัวข้อที่เหลือ
- 3.5.1.2 เขียนข้อมูลลงในแฟ้มข้อมูลผลลัพธ์ ดังนี้
 - ตำแหน่งเริ่มต้นของหัวข้อแรก ซึ่งเป็น 17 เสมอ ลงในไบต์ที่ 1-4
 - ข้อมูลส่วนหัวของแฟ้มข้อมูลนำเข้า
 - ข้อมูลหัวข้อที่ 1 เฉพาะส่วนข้อมูลหัวข้อที่เหลือ



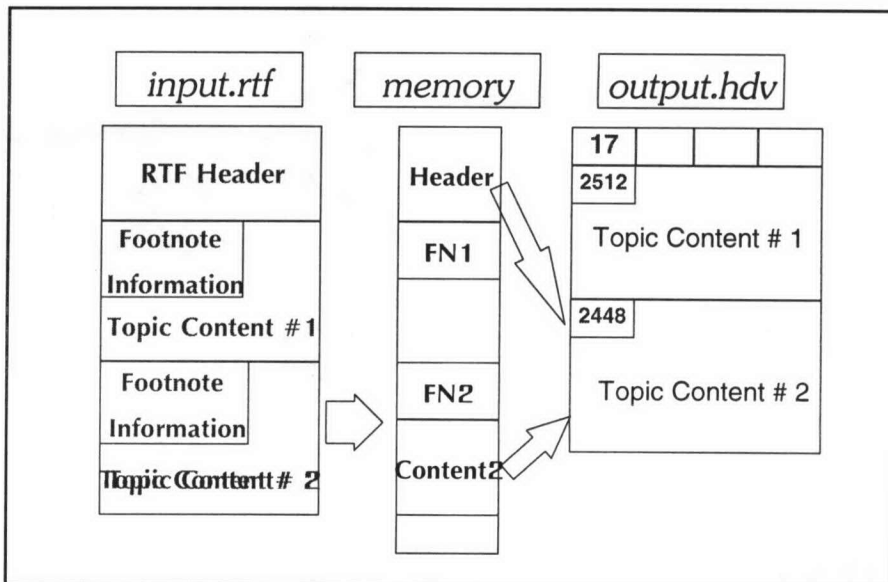
รูปที่ 3-12 แสดงการทำงานของ 3.5.1.1 และ 3.5.1.2

3.5.1.3 อ่านข้อมูลเข้าสู่หน่วยความจำหลัก ได้แก่

- ข้อมูลหัวข้อถัดไป ประกอบด้วย ข้อมูลเชิงอรรถ และข้อมูลหัวข้อที่เหลือ

3.5.1.4 เขียนข้อมูลลงในแฟ้มข้อมูลผลลัพธ์ ดังนี้

- ข้อมูลส่วนหัวของแฟ้มข้อมูลนำเข้า ที่อ่านจากข้อ 3.5.1.1
- ข้อมูลหัวข้อ เฉพาะส่วนข้อมูลหัวข้อที่เหลือ

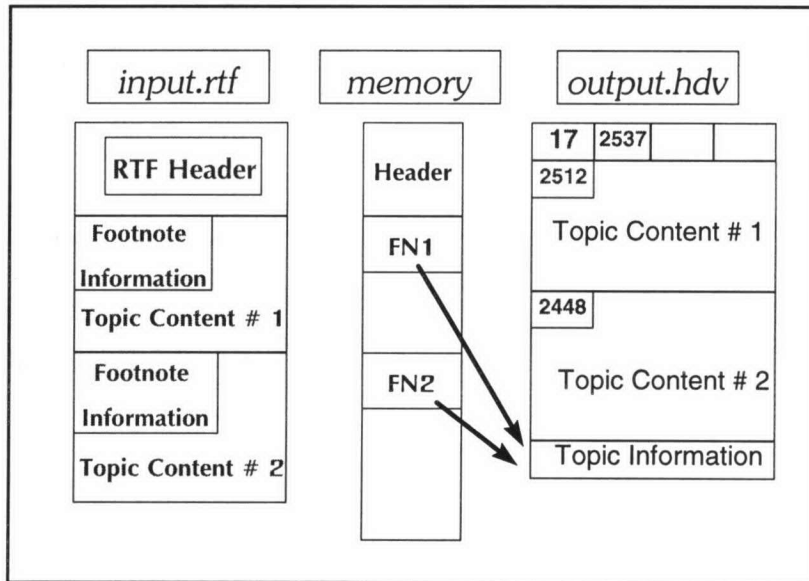


รูปที่ 3-13 แสดงการทำงานของ 3.5.1.3 และ 3.5.1.4

3.5.1.5 ทำซ้ำข้อ 3.5.1.3 และ 3.5.1.4 จนกระทั่งครบทุกหัวข้อ

3.5.1.6 เขียนข้อมูลลงในแฟ้มข้อมูลผลลัพธ์ ดังนี้

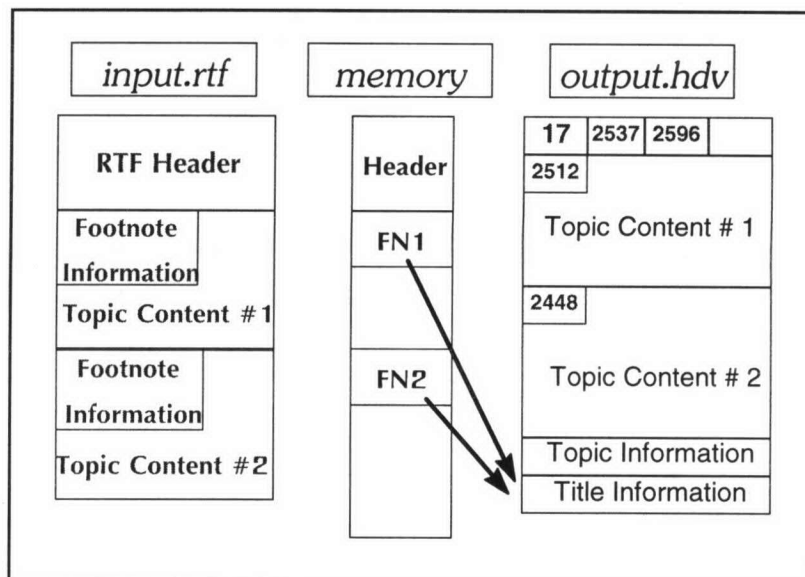
- หมายเลขตำแหน่งเริ่มต้นของรายละเอียดหัวข้อ
- ข้อมูลเชิงอรรถของแต่ละหัวข้อ ที่เก็บไว้ในหน่วยความจำหลัก เฉพาะส่วนที่เป็นรายละเอียดหัวข้อ



รูปที่ 3-14 แสดงการทำงานของ 3.5.1.6

3.5.1.7 เขียนข้อมูลลงในแฟ้มข้อมูลผลลัพธ์ ดังนี้

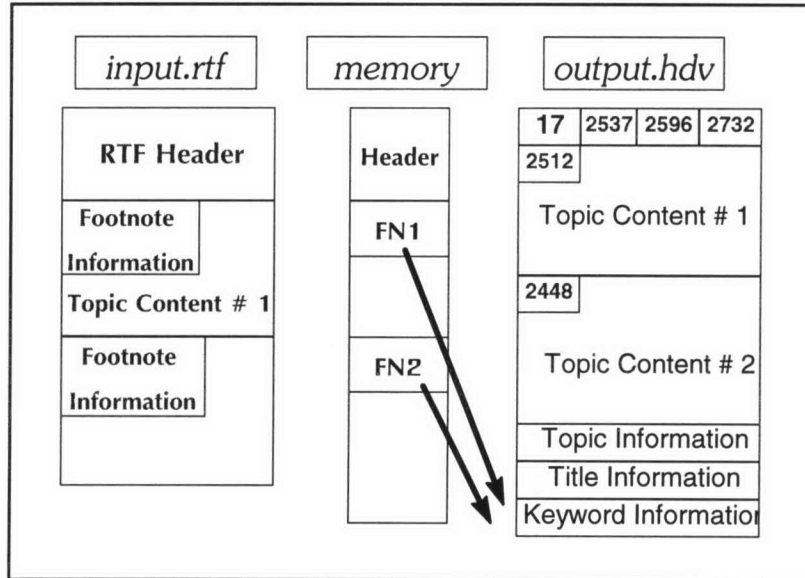
- หมายเลขตำแหน่งเริ่มต้นของรายละเอียดชื่อเรื่อง
- ข้อมูลเชิงอรรถของแต่ละหัวข้อ ที่เก็บไว้ในหน่วยความจำหลัก เฉพาะส่วนที่เป็นรายละเอียดชื่อเรื่อง



รูปที่ 3-15 แสดงการทำงานของ 3.5.1.7

3.5.1.8 เขียนข้อมูลลงในแฟ้มข้อมูลผลลัพธ์ ดังนี้

- หมายเลขตำแหน่งเริ่มต้นของรายละเอียดคำหลัก
- ข้อมูลเชิงอรรถของแต่ละหัวข้อ ที่เก็บไว้ในหน่วยความจำหลัก เฉพาะส่วนที่เป็นรายละเอียดคำหลัก

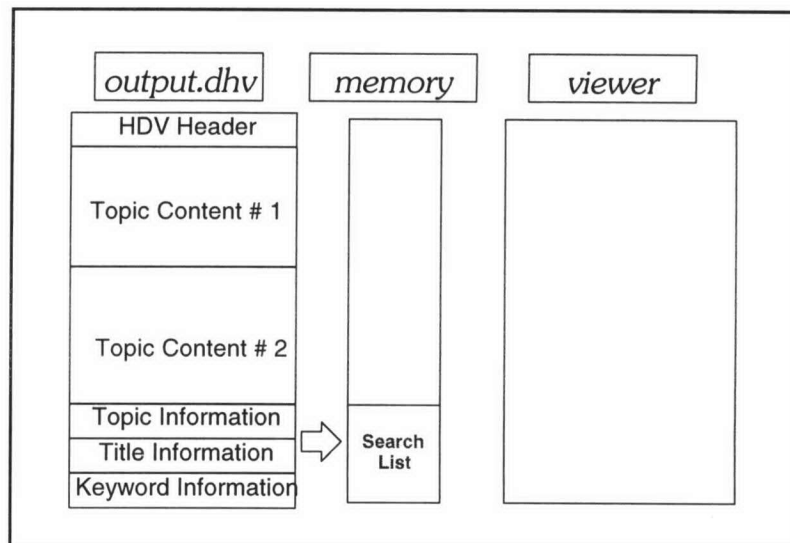


รูปที่ 3-16 แสดงการทำงานของ 3.5.1.8

3.5.2 การพัฒนาโปรแกรมเครื่องมือแสดงผลข้อมูลไฮเปอร์ดอกคิวเมนต์ มีขั้นตอนดังนี้

3.5.2.1 อ่านข้อมูลเข้าสู่หน่วยความจำหลัก ได้แก่

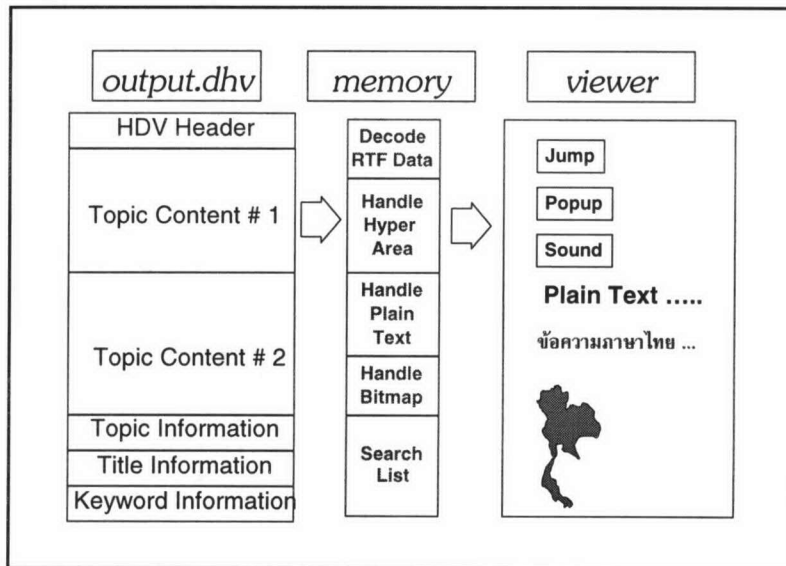
- ข้อมูลรายละเอียดหัวข้อ รายละเอียดชื่อเรื่อง และรายละเอียดคำหลัก
- ข้อมูลหัวข้อที่ 1 โดยแยกประเภทเป็น พื้นที่สำคัญ (hyper area) พื้นที่ชุดอักขระ (plain text) และ พื้นที่ภาพบิตแมพ



รูปที่ 3-17 แสดงการทำงานของ 3.5.2.1

3.5.2.2 แสดงผลบนหน้าต่างแสดงผล ดังนี้

- พื้นที่สำคัญ สำหรับการกระโดด แสดงจอซ้อนแบบผุดขึ้น และประมวลผลเสียง
- ชุดตัวอักษร
- รูปภาพบิตแมพ



รูปที่ 3-18 แสดงการทำงานของ 3.5.2.2

3.5.2.3 อ่านข้อมูลหัวข้อที่ต้องการเข้าสู่หน่วยความจำหลัก ทั้งนี้ หัวข้อที่ต้องการจะได้อ่านด้วยวิธีการ ดังนี้

- ค้นหาจากคำหลัก
- การเลือกพื้นที่สำคัญสำหรับการกระโดด
- การเลือกพื้นที่สำคัญสำหรับจอซ้อนแบบผุดขึ้น
- การเลือกจากชื่อเรื่องที่เคยเรียกดูแล้ว

จากนั้น แสดงผลตามข้อ 3.5.2.2