กรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้างในองค์กร

นายนัทธี ศรีหาจักษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

COLLABORATIVE INFORMATION FRAMEWORK FOR STRUCTURED DOCUMENT RETRIEVAL IN ORGANIZATION

Mr. Nutthee Srihajak

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

หัวข้อวิทยานิพนธ์	กรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้างในองค์กร
โดย	นายนัทธี ศรีหาจักษ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์)

.....กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต)

นัทธี ศรีหาจักษ์ : กรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้างในองค์กร. (COLLABORATIVE INFORMATION FRAMEWORK FOR STRUCTURED DOCUMENT RETRIEVAL IN ORGANIZATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ญาใจ ลิ่มปิยะกรณ์, 71 หน้า.

การค้นหากลุ่มเอกสารที่มีลักษณะสัมพันธ์กันของบริบทเป็นสิ่งที่ท้าทาย เนื่องจากเป็นการยากที่จะประเมินได้ว่าเอกสารที่ได้มานั้นมีเนื้อหาที่ถูกต้อง เหมาะสมและตรงตามความต้องการของผู้ใช้ งานวิจัยนี้จึงได้นำเสนอกรอบงานสารสนเทศควบรวม เพื่อรวบรวมสาระสำคัญที่น่าสนใจและเหมาะสมจากเอกสารที่ได้จากการค้นคืน ซึ่งเป็นเอกสารมีโครงสร้างในรูปแบบเอกซ์เอ็มแอล แนวทางที่นำเสนอประกอบด้วย 2 ส่วนหลัก คือ ส่วนการค้นคืนสารสนเทศจากเอกสาร และส่วนการนำเสนอสารสนเทศ โดยส่วนการค้นคืนสารสนเทศจากเอกสารมีโครงสร้าง ทำหน้าที่แยกส่วน รวบรวมและพิจารณาบริบทในเอกสารเพื่อสกัดสาระสำคัญที่เหมาะสมและตรงตามความต้องการของผู้ใช้งานด้วยเทคนิคการสืบค้นข้อมูลเอกซ์เอ็มแอล ซึ่งใช้ภาษาเอกซ์คิวรีและวิธีการแท็กข้อมูลด้วยคำศัพท์ควบคุมที่ประกอบด้วยคำสำคัญและคำที่มีความหมายใกล้เคียง เพื่อจัดทำเป็นดัชนีด้วยภาษาเอกซ์พาธ ชุดข้อมูลผลลัพธ์จากการสืบค้นจะถูกนำมาหาความสัมพันธ์ของบริบทด้วยเทคนิควิธีการจัดกลุ่มโดยใช้อัลกอริทึมเค-มีนส์ และตัววัดทีเอฟ-ไอดีเอฟ เพื่อบอกความเกี่ยวข้องของเอกสารผลลัพธ์จากการค้นคืน ต่อจากนั้น ส่วนการนำเสนอสารสนเทศจะทำการเรียงลำดับและจัดรูปแบบสารสนเทศตามที่กำหนดไว้ก่อนหน้าด้วยภาษาเอกซ์เอสแอลทีเพื่อแปลงข้อมูลเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอล ผลลัพธ์การค้นคืนสารสนเทศจากการทดลองในงานวิจัยนี้ถูกประเมินด้วยค่าพรีซิชัน รีคอล และค่าเอฟ ได้ค่าเฉลี่ยที่ 83% 84% และ 83% ตามลำดับ ซึ่งอยู่ในระดับดีปานกลาง

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

5470955621 : MAJOR COMPUTER SCIENCE
KEYWORDS: COLLABORATIVE INFORMATION / DOCUMENT DECOMPOSITION / STRUCTURED DOCUMENTS RETRIEVAL

NUTTHEE SRIHAJAK: COLLABORATIVE INFORMATION FRAMEWORK FOR STRUCTURED DOCUMENT RETRIEVAL IN ORGANIZATION. ADVISOR: ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D., 71 pp.

Searching for a cluster of documents with context relevance is challenging as it is difficult to assess whether those documents contain relevant contents and satisfy the user needs. This research therefore presents a Collaborative Information Framework for retrieving the proper and interesting contents from the structured documents in XML format. The proposed approach consists of two main components, which are the part of document information retrieval, and the part of information presentation. The document information retrieval component is in charge of document decomposition, and collection of the proper contexts satisfying user needs with the XML searching technique. The XQuery language and the method of index tagging by XPath language using controlled vocabularies composed of keywords and synonyms. The set of documents resulting from searching will then be clustered by k-Means algorithm, and the measure of TF-IDF for examining the context relevance. Next, the information presentation component will re-order and re-format the obtained information based on the predefined templates using XSLT language to transform XML data to HTML. The results of information retrieval from the experiment in this study, evaluated with the values of Precision, Recall, and F-measure, yield the averages of 83%, 84%, and 83 %, respectively that can be rated moderate.

Department: Computer Engineering Student's Signature .....

Field of Study: Computer Science Advisor's Signature .....

Academic Year: 2013

กิตติกรรมประกาศ

ผู้วิจัยขอขอบพระคุณ รองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งเป็นผู้เสียสละเวลาในการให้คำปรึกษา คำแนะนำ แนะแนวทางในการวิจัยอย่างดีตลอดมาจนเสร็จสมบูรณ์ ชี้ให้เห็นถึงการปัญหาด้วยตัวเอง เสริมสร้างและหล่อหลอมคุณลักษณะของมหาบัณฑิตที่ดี

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ประกอบด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล ประธานกรรมการ และอาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต ผู้ทรงคุณวุฒิและกรรมการ ที่เป็นผู้ให้คำแนะนำ ข้อคิดเห็น ข้อเสนอแนะ แนวทางในการพัฒนาและตรวจสอบงานวิจัยนี้

ขอขอบพระคุณครอบครัวทุกคนที่คอยให้กำลังใจ คอยสนับสนุนมาโดยตลอด จนงานวิจัยนี้สำเร็จลุล่วงด้วยดีและขอขอบคุณพี่ๆ เพื่อนๆ น้องๆ นิสิตสาขาวิทยาศาสตร์คอมพิวเตอร์ทุกคน ที่ให้กำลังใจให้การสนับสนุนและความช่วยเหลือในด้านต่างๆ และท่านอื่นๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนช่วยให้วิทยานิพนธ์ของข้าพเจ้าสำเร็จไปได้ด้วยดี

สุดท้ายนี้ขอขอบพระคุณคณาจารย์ทุกท่านตั้งแต่อดีตจนถึงปัจจุบันที่ได้ประสิทธิ์ประสาทความรู้แก่ผู้วิจัย ทำให้สามารถนำความรู้ที่ได้สั่งสมมาเพื่อใช้เป็นพื้นฐานในการดำเนินงานวิจัยจนสำเร็จเป็นวิทยานิพนธ์นี้

สารบัญ

หน้า

หน้า

หน้า

สารบัญตาราง

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การพัฒนาการทำงานในองค์กรให้มีประสิทธิภาพนั้น ปัจจัยที่สำคัญส่วนหนึ่งคือการพัฒนาบุคลากรให้มีความรู้ความสามารถ ดังนั้น กลยุทธ์ในการจัดการความรู้ในองค์กรจึงถูกนำมาใช้เพื่อเพิ่มศักยภาพของบุคลากรอย่างยั่งยืน และเพิ่มเติมด้วยความรู้ที่หลากหลายทั้งภายในและภายนอกองค์กร ปัจจุบันมีองค์ความรู้ที่เรียกว่าการค้นคืนสารสนเทศ (Information Retrieval) [8] เป็นการค้นคืนโดยรวมเอาข้อมูล (Data) สารสนเทศ (Information) และการบริการ (Services) จากแหล่งต่างๆมาใช้งานได้อย่างรวดเร็วและมีประสิทธิภาพ ในการจัดการองค์ความรู้ (Knowledge Management) [16] เพื่อให้บุคลากรมีการเรียนรู้ด้วยตนเองแบบอีเลิร์นนิ่ง (e-Learning) และนำความรู้ภายในและภายนอกองค์มาใช้ให้เกิดประโยชน์สูงสุด งานวิจัยนี้จึงมีแนวคิดใหม่ที่จะทำสารสนเทศควบรวม ด้วยการพัฒนาระบบเพื่อให้ผู้ใช้งานสามารถใช้ข้อมูลสารสนเทศที่สัมพันธ์กับสิ่งที่ทำการสืบค้นจากเอกสารประเภทต่างๆด้วยกรอบงานสารสนเทศควบรวมจากการประมวลผลคำสืบค้นจากดัชนีคำศัพท์ที่กำหนดไว้เฉพาะสำหรับหน่วยงานหรือองค์กรเพื่อใช้งานในการสืบค้นด้วยภาษาเอกซ์คิวรี (XQuery) [12] และจะใช้ภาษาเอกซ์พาธ (XPath) [9] มาร่วมในการเข้าถึงโหนดต่างๆในเอกสารเอกซ์เอ็มแอล (XML documents) [9, 10, 11] สำหรับงานวิจัยนี้จะนำเสนอการออกแบบและพัฒนาการรวบรวมข้อมูลสารสนเทศที่เกี่ยวข้องและเป็นประโยชน์ต่อผู้ใช้งานโดยอาศัยการจัดเรียงลำดับข้อมูลสารสนเทศที่น่าสนใจและเหมาะสมจากเอกสารที่เป็นผลลัพธ์จากการค้นคืน ซึ่งมีกลไกในการจัดเก็บและเรียงลำดับเนื้อหาสาระสำคัญในเอกสารที่เกี่ยวข้องและเหมาะสมเพื่อแสดงผลข้อมูลให้กับผู้ใช้งานโดยใช้ความรู้ทางด้านภาษาเอกซ์เอสแอลที [18] ที่เป็นมาตรฐานเอกซ์เอ็มแอล

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอการออกแบบและพัฒนาระบบในการรวบรวมสารสนเทศจากแหล่งข้อมูลหรือสารสนเทศที่ถูกจัดเก็บเป็นเอกสารที่มีโครงสร้าง โดยใช้กลไกการประมวลผลดึงข้อมูลหรือสารสนเทศที่มีสาระสำคัญและเหมาะสมของเนื้อหาที่เกี่ยวข้องกับผู้ใช้งาน และทำการแสดงผลลัพธ์สารสนเทศที่ได้ในรูปแบบที่กำหนดไว้ ทำให้ผู้ใช้งานสามารถนำสารสนเทศที่ได้ไปใช้งานอย่างมีประสิทธิภาพและตรงกับความต้องการ

1.3 ขอบเขตของการวิจัย

1. เอกสารที่ใช้ในการพัฒนาระบบการค้นคืนสารสนเทศจะเป็นเอกสารประเภทโครงสร้างเอกซ์เอ็มแอลสามารถใช้ร่วมกันได้กับมาตรฐานของโอเพนเอกซ์เอ็มแอล (OpenXML) [15] และสามารถสืบค้นด้วยภาษาเอกซ์คิวรี
2. ระบบการสืบค้นเอกสารจะใช้ภายในองค์กรที่ทำการทดสอบระบบ รวมถึงการวัดผลประเมิน ซึ่งเป็นเครื่องมือร่วมในการพัฒนาระบบการจัดการเอกสาร (Documents Management)
3. เลือกเครื่องมือที่เหมาะสมมาใช้ในการออกแบบและพัฒนาระบบทั้ง 2 กระบวนการคือ การค้นคืนเอกสาร และการนำเสนอสารสนเทศ
4. การประเมินผลการทดสอบจะใช้วิธีการพิจารณาจำนวนผลลัพธ์ที่ได้จากการสืบค้นด้วยเอกซ์คิวรี เป็นจำนวนของพาธหรือจำนวนของข้อความ และใช้หลักการทำงานของการจัดกลุ่มข้อความโดยการรวมกันของอัลกอริทึม K-means และเทคนิควิธีวัด TF-IDF [4] เพื่อกำหนดค่าน้ำหนักของคำสืบค้นเพื่อหาความสัมพันธ์ของเนื้อความ โดยพิจารณาจากอัตรา Recall และ Precision ของทฤษฎีการค้นคืนสารสนเทศ

1.4 ข้อตกลงเบื้องต้น

1. งานวิจัยนี้มีขอบเขตสภาพแวดล้อมการทำงานระบบการจัดการเอกสารภายในองค์กร
2. ศัพท์ควบคุมที่นำมาใช้ร่วมในงานวิจัยนี้เป็นส่วนหนึ่งของการทดลองของหน่วยภายนอก
3. เอกสารต้นฉบับที่ใช้ในการทดลองของงานวิจัยนี้จะเป็นไมโครซอฟต์เวิร์ด (.docx) เวอร์ชัน 2007 หรือล่าสุดเท่านั้น

1.5 ข้อจำกัดของการวิจัย

1. เอกสารต้นฉบับที่ใช้ในการทดสอบระบบจะเป็นเอกสารภาษาอังกฤษ
2. ข้อมูลที่เป็นผลลัพธ์ที่นำมาใช้แสดงผลจะเป็นข้อความเท่านั้น

1.6 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำระบบสารสนเทศควบรวมมาใช้ร่วมกับระบบการเรียนรู้ด้วยตนเอง โดยที่มีการนำข้อมูลที่น่าสนใจและเกี่ยวข้องโดยตรงต่อผู้ใช้มาแสดงแบบอัตโนมัติ ทำให้เกิดการพัฒนาประสิทธิภาพการเรียนรู้ด้วยตนเองเมื่อนำมาใช้ในองค์กรจะทำให้เกิดการพัฒนาบุคลากรได้อย่างมีประสิทธิภาพและต่อเนื่อง

1.7 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีพื้นฐานของขั้นตอนการประมวลผลการค้นคืนสารสนเทศและการแยกส่วนประกอบจากเอกสารที่มีโครงสร้าง
2. ศึกษาวิธีการในการออกแบบและพัฒนาเอกสารที่มีโครงสร้าง และเลือกเครื่องมือที่เหมาะสมเพื่อใช้ในการดำเนินการวิจัย
3. ออกแบบและพัฒนารูปแบบของเอกสารเพื่อใช้ในการจัดเก็บสารสนเทศที่ได้จากการค้นคืนสาระสำคัญ และเหมาะสมจากเอกสารที่เป็นแบบโครงสร้าง
4. ออกแบบและพัฒนากระบวนการที่ใช้เป็นกลไกประมวลผลสารสนเทศจากเอกสารที่สืบค้นได้ ประกอบด้วย 2 กระบวนการ คือ
    4.1 กลไกการดึงสารสนเทศจากเอกสาร
    4.2 กลไกการแสดงผลลัพธ์จากข้อ 4.1
5. วิเคราะห์ผลลัพธ์จากวิธีการที่นำเสนอ
6. สรุปผลและเรียบเรียงวิทยานิพนธ์
7. ตีพิมพ์ผลงานวิชาการ
8. จัดทำวิทยานิพนธ์

1.8 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ลำดับลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิจัยในหัวข้อเรื่อง “กรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้าง” โดย นัทธี ศรีหาจักษ์ และ ญาใจ ลิ่มปิยะกรณ์ ในวารสารรามคำแหง ฉบับวิศวกรรมศาสตร์ (Ramkhamhaeng Journal of Engineering) ปีที่ 8 ฉบับที่ 1 เดือนพฤษภาคม 2557

1.9 ลำดับลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท ดังนี้ บทที่ 1 บทนำ กล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของงานวิจัย ประโยชน์ที่คาดว่าจะได้รับและผลงานตีพิมพ์ บทที่ 2 กล่าวถึง ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับงานวิจัย บทที่ 3 กล่าวถึง วิธีดำเนินการวิจัย บทที่ 4 กล่าวถึง การออกแบบและพัฒนาระบบตามแนวทางการวิจัย บทที่ 5 กล่าวถึงวิธีการประเมินและวัดผลการทดลอง และบทที่ 6 สรุปผลการวิจัย ข้อเสนอแนะ และแนวทางสำหรับการวิจัยต่อ

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 โมเดลการค้นคืนสารสนเทศ (Information Retrieval Models) [8]

การค้นคืนสารสนเทศมาจากทฤษฎีพื้นฐานทางคณิตศาสตร์ที่มีการพัฒนาในอดีตจนปัจจุบัน โดยใช้รูปแบบที่แสดงในเชิงตรรกะ(Logical View) เพื่อจำลองตัวเอกสาร คลังข้อมูลเอกสาร สารสนเทศที่ผู้ใช้ต้องการหรือคำที่ต้องการค้นหา และจะดำเนินการเปรียบเทียบเพื่อหาผลลัพธ์โดยใช้หลักการจัดกลุ่ม โมเดลการค้นคืนสารสนเทศสามารถเขียนเป็นรูปภาพหรือแสดงเป็นสัญลักษณ์ และสามารถเขียนเป็นสมการทางคณิตศาสตร์เพื่อแสดงกระบวนการปฏิบัติการทางกายภาพ ตัวอย่างเช่น ให้ *D* เป็นสัญลักษณ์ของคลังเอกสาร และ d_j เป็นเอกสารฉบับที่ *j* ในคลังเอกสาร และ *q* เป็นคำที่ใช้ค้นหา กำหนดให้ M เป็นฟังก์ชันของการเปรียบเทียบระหว่าง *q* กับ d_j จะเป็น M(*q*, d_j)

วิทยาการค้นคืนสารสนเทศสามารถนำทฤษฎีทางคณิตศาสตร์มาใช้ในการสร้างโมเดลในรูปแบบต่างๆ ได้ 3 แบบหลักดังนี้

- ทฤษฎีเซต (Set Theory)
- เมตริกซ์พีชคณิต (Matrix Algebra)
- ทฤษฎีความน่าจะเป็น (Probability Theory)

ทฤษฎีเซตจะเป็นโมเดลของเอกสารประกอบด้วยคำต่างๆที่อยู่ภายใน เพื่อความสะดวกในการสืบค้นจะใช้การสร้างดัชนีของคำเพื่อจับคู่เปรียบเทียบกับคำสืบค้น ผลลัพธ์ที่ได้ประกอบด้วย 2 ค่าคือ เหมือนกับไม่เหมือน คำสืบค้นสามารถเขียนเทอมปฏิบัติการ AND, OR และ NOT เพื่อใช้ในการเปรียบเทียบโดยผลลัพธ์ที่ได้จะเป็นเซตของเอกสาร และทำการรวมเซตด้วยบูลีนโอเปอร์เรชัน จึงเรียกว่า Classical Boolean Model (CBM) ต่อมามีการพัฒนาปรับปรุง CMB ด้วยวิธีการเมตริกซ์พีชคณิตโดยใช้ Term-Document Matrix เพื่อสร้างคำดัชนี ซึ่งในแต่ละหน่วยของเมตริกซ์จะกำหนดค่าน้ำหนักขึ้นอยู่กับจำนวนความถี่ของคำ (Term Frequency) ที่ปรากฏในเอกสาร คำสืบค้นสามารถเขียนเป็นเวกเตอร์ของเทอมที่สอดคล้องกับคำตามแนวตั้งของ Term-Document Matrix โดยคำที่ปรากฏจะถูกกำหนดค่าน้ำหนักขึ้นอยู่กับคำสำคัญของคำนั้นๆ ส่วนคำที่ไม่ปรากฏจะถูกกำหนดค่าเป็น 0 หลักการเปรียบเทียบระหว่างคำสืบค้นกับเอกสารในแต่ละชุดจะให้ค่าเป็น $\cos\theta$ โดย θ เป็นมุมระหว่างเวกเตอร์ทั้งสอง ถ้าทำมุมได้ 0° ค่า $\cos\theta$ จะเข้าใกล้ 1 ถ้าทำมุมได้ 90° ค่า $\cos\theta$ จะเข้าใกล้ 0 ปฏิบัติการทางคณิตศาสตร์ใช้วิธีการ Inner Product ของเวกเตอร์ทำให้สามารถจัดอันดับ (Ranking) ของเอกสารที่เป็นคำตอบตามค่า $\cos\theta$ โดยพิจารณาจากค่าที่สูงก่อนแล้วลดลำดับลง โมเดลนี้เรียกว่าปริภูมิเวกเตอร์ (VSM: Vector Space Model) งานวิจัยนี้ใช้โมเดลดังกล่าวมาเป็นพื้นฐานในการจัดลำดับบริบทที่มีความสัมพันธ์และเกี่ยวข้องกับสิ่งที่ผู้ใช้ต้องการค้นหา ส่วนทฤษฎีความน่าจะเป็นมีความแตกต่างจากทั้งสองโมเดลดังกล่าว โดยใช้วิธีการเปรียบเทียบระหว่างคำสืบค้นกับเอกสารด้วยค่าความน่าจะเป็น 2 ค่าคือ ค่าความน่าจะเป็นที่เอกสารจะเกี่ยวข้องกับคำสืบค้น แทนด้วย $P(\mathbf{R} \mid q, \mathbf{d_j})$ ส่วนอีกค่าคือ ค่าความน่าจะเป็นที่เอกสารจะไม่เกี่ยวข้องกับคำ

สืบค้น แทนด้วย $P(\overline{\mathbf{R}}|\ q, \mathbf{d_j})$ ดังนั้นค่าความสอดคล้องจึงกำหนดเป็น $P(\mathbf{R}\ |\ q, \mathbf{d_j}) / P(\overline{\mathbf{R}}|\ q, \mathbf{d_j})$ รายละเอียดยังมีการคำนวณที่ซับซ้อนจึงไม่เป็นที่นิยมในวิทยาการค้นคืนสารสนเทศ

2.1.2 โมเดลปริภูมิเวกเตอร์ (Vector Space Model) [8, 3]

การสร้างคลังเอกสารในรูปเมตริกซ์ของ Term-Document และกำหนดค่าคำสืบค้นกับตัวเอกสารให้อยู่ในรูปเวกเตอร์ $\overrightarrow{d_i} = (w_{i,1}, w_{i,2} \dots w_{i,j})$ การวัดค่าความเหมือนกันจะใช้วิธีการวัดระยะห่างระหว่างคู่เวกเตอร์ด้วยการทำ Inner Product ซึ่งสามารถนำผลลัพธ์ที่ได้มาจัดลำดับของความสอดคล้องกัน คำในปริภูมิเวกเตอร์จะมีค่าสอดคล้องกับความถี่ที่ปรากฏในเอกสาร ค่าดังกล่าวจะเป็นน้ำหนักของคำ แทนด้วย $w_{i,j}$ กำหนดให้ $w_{i,j}$ ประกอบด้วย 2 ค่าคือ tf เป็นความถี่ของคำที่ปรากฏ (Term Frequency) และ idf เป็นค่าผกผันจำนวนความถี่ของเอกสารที่มีคำนั้นปรากฏ (Inverse Document Frequency) ดังนั้น $w_{i,j} = tf_{i,j} \times idf_i$

เมื่อพิจารณาค่า tf จะพบว่าแม้จะเป็นค่าความถี่ของการปรากฏ แต่ในความเป็นจริงแล้วจำนวนคำในแต่ละเอกสารมีไม่เท่ากัน ดังนั้นเพื่อให้สามารถนำมาเปรียบเทียบกันโดยที่ไม่ขึ้นกับขนาดของจำนวนคำในเอกสาร จึงกำหนด tf เป็นนอร์มอไลซ์ฟอร์ม ดังนี้

$$tf_{i,j} = \frac{f_{i,j}}{max_k f_{k,j}}$$

โดย $f_{i,j}$ เป็นข้อมูลดิบของความถี่ของคำ t_i ในเอกสาร j

นอกจากวิธีการเลือกค่าสูงสุดของความถี่เป็นตัวหาร มีอีกวิธีการคือใช้ผลรวมของความถี่ทั้งหมดเป็นตัวหาร แทนด้วย $\sum_k f_{k,j}$

การคำนวณ *idf* เนื่องจากจะขึ้นอยู่กับค่าจำนวนความถี่ของเอกสารที่มีคำนั้นปรากฏและมีลักษณะเป็นค่าผกผัน ดังนั้นจะใช้ *log* มาช่วยเพื่อไม่ให้เกิดค่าแปลกแยกในการคำนวณ ดังนี้

$$idf_i = \log \frac{N}{n_i}$$

โดย N เป็นจำนวนเอกสารทั้งหมดในระบบ

n_i เป็นจำนวนเอกสารทั้งหมดที่ปรากฏของคำ i

$\log$ เป็น $\log$ ฐาน *2* หรือฐาน $\mathbf{e}$ แทนด้วย $\ln$

เอกสารในระบบทั้งหมดแทนด้วยเมตริกซ์ *D* ของเอกสารที่ 1 ถึง *N* ดังนี้

$$D = [d_1 \quad d_2 \quad d_3 \cdots d_j \cdots d_N]$$

$$= \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,1} \\ w_{2,1} & w_{2,2} & & w_{2,N} \\ \vdots & & w_{i,j} & \vdots \\ \vdots & & & \vdots \\ w_{M,1} & w_{M,2} & & w_{M,N} \end{bmatrix}$$

D คือ เมตริกซ์ขนาด *M* × *N* ส่วนค่า $w_{i,j}$ จะหมายถึงความสำคัญของคำที่ t_i ในเอกสาร *j*

เมื่อเอกสารใดที่ไม่มีคำ t_i ปรากฏ จะได้ค่า $w_{i,j}$ เท่ากับ 0 ซึ่งทำให้เกิดค่าที่ไม่เป็นเท่ากับ 0 จำนวนมาก แต่คำที่ไม่เท่ากับ 0 แสดงถึงความสำคัญของคำนั้นในเอกสาร เมื่อถูกนำไปเปรียบเทียบกับคำนั้นในเอกสารอื่น และเปรียบเทียบกับคำอื่นในเอกสารเดียวกัน

วิธีการกำหนดค่าน้ำหนักของคำดังกล่าวมีชื่อเรียกเป็นทางการว่าวิธี *tf-idf* ส่วนการคำนวณหาค่า *tf* และ *idf* มีความแตกต่างกันในรายละเอียด

$$Term\ Weight = w_i = tf_i * \log(\frac{D}{df_i}) \qquad (2.1)$$

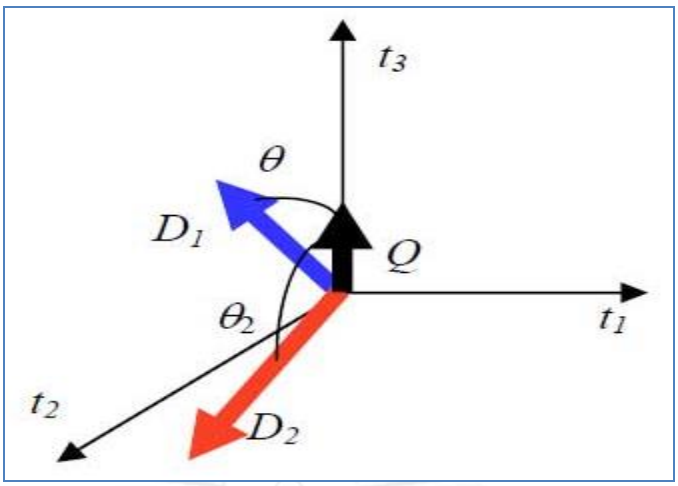
การคำนวณค่าความเหมือน (Similarity) ถูกนำมาพิจารณาโดยการหา Inner Product ดังนี้

$$SIM(q, d_j) = \frac{q \cdot d_j}{\|q\| \|d_j\|}$$

$$= \frac{\sum_{i=1}^{M} (w_{i,q} \times w_{i,j})}{\sqrt{\sum_{i=1}^{M} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{M} w_{i,j}^2}} \qquad (2.2)$$

ให้ θ เป็นมุมระหว่างเวกเตอร์ q ทำกับเวกเตอร์ d_j จากสมการ (2.1) แทนด้วย

$$SIM(q, d_j) = \cos\theta \qquad (2.3)$$



ภาพที่ 2.1 ปริภูมิเวกเตอร์ 3 มิติ t1, t2 และ t3

นอกจากนี้ศาสตราจารย์ Salton และ Buckly ยังแนะนำการให้น้ำหนักของ $w_{i,q}$ ดังนี้

$$w_{i,q} = \left(0.5 + \frac{0.5f_{i,q}}{\max_k f_{k,q}}\right) \times \log\frac{N}{n_i}$$

Robertson และ Spark Jones แนะนำในการคำนวณคำดังนี้

$$w_{i,j} = \frac{(u+1)\cdot tf_{i,j}\cdot idf_j}{\mu\cdot\left((1-\beta)+\beta\cdot ndl_i\right)+tf_{i,j}}$$

โดย $ndl_i = \frac{doc\ len_i}{avg\ len}$

μ เป็นตัวปรับค่า ขึ้นอยู่กับการความถี่ของจำนวนคำ โดยทั่วไปจะมีค่า $\mu \simeq 2$

β เป็นตัวปรับค่าอีกประเภทหนึ่งที่มีค่าอยู่ในช่วง $0 \leq \beta \leq 1$ ขึ้นอยู่กับขนาดความยาวของเอกสาร โดยที่

$\beta = 0$ กรณีเป็นเอกสารขนาดยาวเพราะครอบคลุมในหลายหัวข้อ

$\beta = 1$ กรณีเป็นเอกสารขนาดยาวเพราะเกิดการซ้ำๆกัน

โดยทั่วไปจำเลือก $\beta = 0.75$

Topic-based Vector Space Model (TVSM) นำเสนอโดย Becker และ Kuropka ซึ่งมีแนวคิดให้คำในเอกสารที่อิสระต่อกัน เพื่อความยืดหยุ่นในการกำหนดความคล้ายกันของคำ (Term Similarity) ซึ่งในแต่ละแกนในปริภูมิเวกเตอร์จะใช้ชื่อเรื่องพื้นฐาน (Fundamental Topic) แทนคำ และเป็นเวกเตอร์ใน *k Dimensional Space R* จะพิจารณาเฉพาะแกนบวกเท่านั้น โดยไม่ใช้แกนลบ

$R = \Re_{\geq 0}^{k}$ ด้วย $k \in N_{\geq 1}$

ในแต่ละแกน R เรียกว่า Fundamental Topic มีคุณสมบัติเป็น Orthogonal และ Independent ต่อกัน

ให้คำ $t_i \in T$ โดย T เป็นเซตของทุกคำและเวกเตอร์ t_i อยู่ใน R โดยกำหนดน้ำหนักมีค่าใน [0,1] และทิศทางของเวกเตอร์แต่ละคำแสดงถึงความสัมพันธ์ของคำต่อกันโดยอิงตามชื่อเรื่องพื้นฐานดังกล่าว จะได้สมการดังนี้

$$\boldsymbol{t_i} = \left(t_{i,1}, t_{i,2}, \cdots t_{i,k}\right)$$

$$|t_i| = \sqrt{t_{i,1}^2 + t_{i,2}^2 + \cdots t_{i,k}^2} \quad \in [0,1]$$

ให้ d_j เป็นเอกสารที่ j ในคลังเอกสาร D แทนด้วย $d_j \in D$ เวกเตอร์ของเอกสาร $d_j \in R$ และเมื่อทำนอร์มอลไลเซชันโดยให้ขนาดความยาวเอกสารเท่ากับ 1 แทนด้วยสมการดังนี้

$$\forall d_j \in D: d_j = \frac{1}{\delta_j}\delta_j \Longrightarrow |d_j| = 1$$

โดยที่ $\delta_j = \sum_{t_i \in T} w_{d_j,t_i} t_i$ เมื่อ w_{d_j,t_i} เป็นน้ำหนักของชื่อเรื่องพื้นฐาน t_i และเอกสาร d_j

การหาความคล้ายกันของเอกสาร d_i และ d_j จะใช้หลักการคำนวณ Cosine ของมุมที่เวกเตอร์ทั้งสองเอกสารกระทำต่อกัน แทนด้วยสมการดังนี้

$$Sim(d_i, d_j) = d_i \cdot d_j$$

$$Sim(d_i, d_j) = |d_i||d_j| \cos \alpha$$

$$= \cos \alpha$$

จากที่ TVSM กำหนดให้ใช้แกนบวกทั้งหมดใน R ดังนั้นเพื่อให้การทำมุมระหว่างเอกสารใดๆ ต้องไม่เกิน 90° หรือ $\forall\alpha \in [0^{\circ}, 90^{\circ}]$ จะได้ว่า $\forall\alpha \in [0,1]$ เมื่อค่าความคล้ายกันเท่ากับ 0 แสดงว่าเอกสารทั้งสองไม่เหมือนกัน และถ้าค่าความคล้ายกันเท่ากับ 1 แสดงว่าเอกสารทั้งสองเหมือนกัน 100% โดยทำมุม 0° ต่อกันและสามารถคำนวณหาความเหมือนอื่นๆเป็นเปอร์เซ็นต์ได้จากค่ามุมองศาที่กระทำต่อกัน จากสมการ δ และ t ที่ประกอบเป็นเอกสารสามารถนำมาคำนวณได้ดังนี้

$$Sim(d_i, d_j) = d_i \cdot d_j$$

$$Sim(d_i, d_j) = \frac{1}{|\delta_i|}\delta_i \cdot \frac{1}{|\delta_j|}\delta_j$$

$$Sim(d_i, d_j) = \frac{1}{|\delta_i||\delta_j|}\delta_i \cdot \delta_j$$

$$Sim(d_i, d_j) = \frac{1}{|\delta_i||\delta_j|}\sum_{t_k \in T} w_{d_i,t_k} t_k \cdot \sum_{t_1 \in T} w_{d_j,t_1} t_1$$

$$Sim(d_i, d_j) = \frac{1}{|\delta_i||\delta_j|}\sum_{t_k \in T}\sum_{t_1 \in T} w_{d_i,t_k}\, w_{d_j,t_1} t_k t_1$$

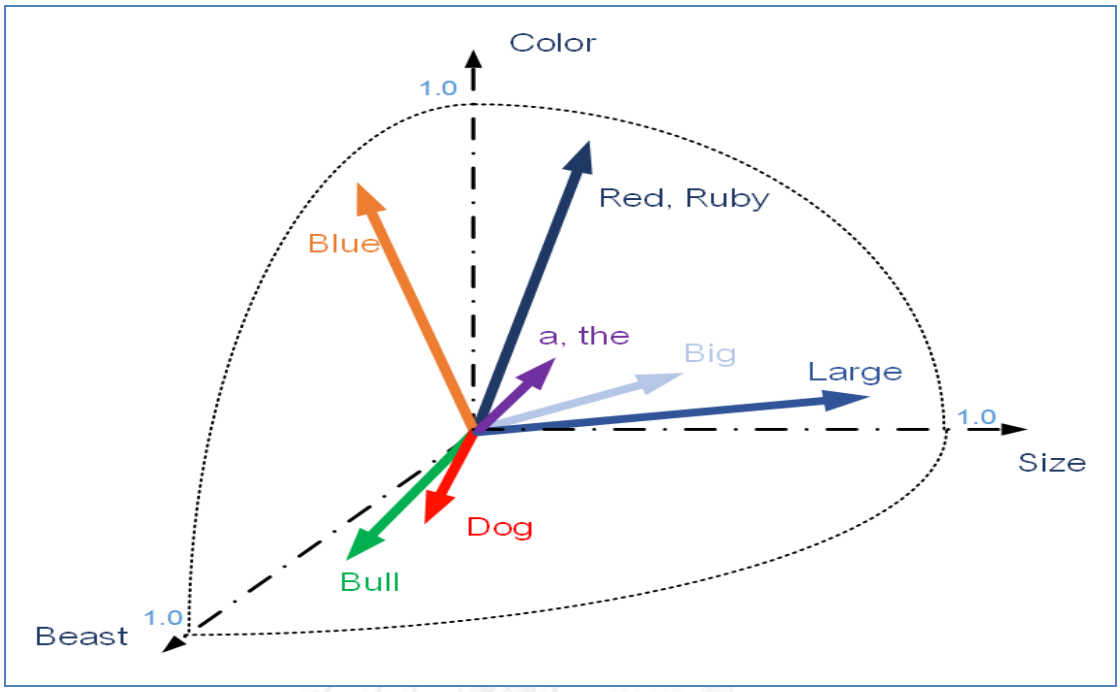
เมื่อพิจารณาค่าของเทอม $|\delta_i|$ และ $|\delta_j|$ คำนวณได้ดังนี้

$$|\delta_i| = \left|\sum_{t_k \in T} w_{d_i,t_k} t_k\right|$$

$$|\delta_i| = \sqrt{\left|\sum_{t_k \in T} w_{d_i,t_k} t_k\right|^2}$$

$$|\delta_i| = \sqrt{\left(\sum_{t_k \in T} w_{d_i,t_k} t_k\right)^2}$$

$$|\delta_i| = \sqrt{\sum_{t_k \in T}\sum_{t_1} w_{d_i,t_k} t_k t_1}$$



ภาพที่ 2.2 ปริภูมิเวกเตอร์ TSVM 3 แกนมิติของชื่อเรื่องที่ใช้ค้นหา [8]

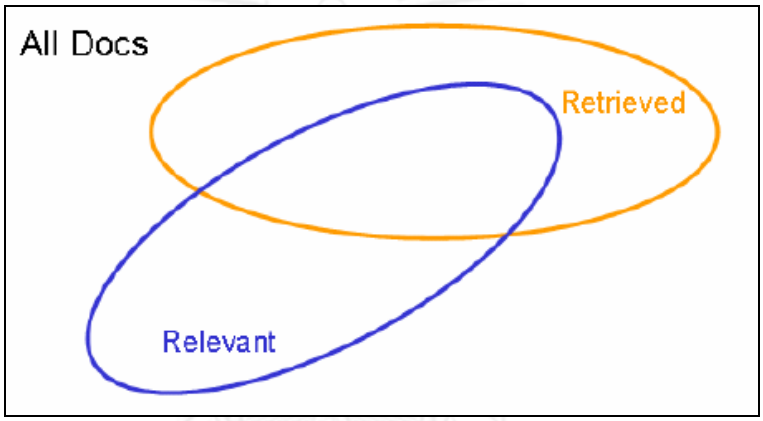
จากภาพที่ 2.2 เป็นตัวอย่างข้อมูลปริภูมิเวกเตอร์ 3 แกนของ TVSM ในแต่ละแกนแทนชื่อเรื่องพื้นฐานของ Size, Color และ Beast ซึ่ง TSVM สามารถรวมคำในลักษณะต่างๆได้ เช่น Synonym, Inflection, Composition, Hyponym และ Meronym ดังนี้

- **Synonym** เมื่อ 2 คำมีความหมายเหมือนกัน ดังนั้นมุมของเวกเตอร์จะมีค่าเข้าใกล้ 0° ซึ่งเป็นแนวคิดของ Thesaurus ที่นำมาใช้ใน TVSM
- **Stemming** เมื่อ 2 คำมาจากรากศัพท์เดียวกัน เช่น Compute, Computing มุมของเวกเตอร์จะต้องเป็น 0° หรือเข้าใกล้ 0 เป็นแนวคิด Stemming ที่นำมาใช้ใน TVSM
- **Hyponym** หมายถึงความเป็นชนิดของ (a kind of) มีความหมายตรงข้ามกับ Hypernym เมื่อ 2 คำที่มาจากชนิดเดียวกัน ย่อมทำให้เวกเตอร์มุมที่กระทำระหว่างทั้ง 2 คำมีค่าเข้าใกล้ 0
- Meronym หมายถึงความเป็นส่วนหนึ่งของ (a part of) มีความหมายตรงข้ามกับ Holonym เมื่อ 2 คำเป็นส่วนหนึ่งของสิ่งเดียวกัน ย่อมทำให้เวกเตอร์มุมที่กระทำระหว่างทั้ง 2 คำมีค่าเข้าใกล้ 0

2.1.3 การวัดผลการประเมินการค้นคืน (Retrieval Evaluation Measure) [8, 13, 14]

การคำนวณค่า Recall และค่า Precision

การวัดค่า Recall และค่า Precision ทำได้ยากทั้งในทางทฤษฎีและทางปฏิบัติ ปัญหาในการกำหนดค่า Recall และค่า Precision คือการอธิบายความหมายของความเกี่ยวข้อง และมีนิยามเกี่ยวกับความเกี่ยวข้องที่เป็นไปได้ว่า “ความเกี่ยวข้องนั้นเป็นความสอดคล้องระหว่างคำสืบค้นกับเอกสาร กล่าวได้ว่า ปริมาณที่เอกสารต่างๆ จะครอบคลุมเอกสารที่เหมาะสมหรือเกี่ยวเนื่องกับคำสืบค้น” [13]



ภาพที่ 2.3 ความสัมพันธ์กันระหว่างเอกสารที่ถูกสืบค้นกับเอกสารที่เกี่ยวข้องจากจำนวนของเอกสารทั้งหมด [14]

ค่า Recall จะถูกกำหนดให้เป็นอัตราส่วนของเอกสารที่เกี่ยวข้องที่ถูกดึงออกมาจากจำนวนเอกสารที่เกี่ยวข้องทั้งหมด ในขณะที่ค่า Precision เป็นอัตราส่วนของเอกสารที่เกี่ยวข้องถูกดึงออกจากจำนวนเอกสารที่ถูกดึงออกมาทั้งหมด จากภาพที่ 2.3 สามารถแทนด้วยสมการของค่า Recall กับค่า Precision ได้ดังนี้

$$\text{Recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$$

$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|}$$

$$\text{Precision} = \frac{\text{number of relevant items retrieved}}{\text{number of items retrieved}}$$

$$\text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

ในระบบการค้นคืนข้อมูลแบบดั้งเดิมคำสืบค้นจะถูกแสดงเป็น Boolean Combinations ของดัชนีที่ใช้ในการค้นหา เอกสารที่ถูกดึงออกมาประกอบด้วยคำสำคัญในเอกสาร โดยเอกสารที่ได้นั้นจะมีคำสำคัญระบุในคำสืบค้นรวมกัน ซึ่งในแต่ละคำสืบค้นจะทำให้เกิดผลลัพธ์ทั้งเอกสารที่เกี่ยวข้องและเอกสารที่ไม่เกี่ยวข้อง ดังนั้นในแต่ละคำสืบค้นจะเกิดตัวเลขของค่า Precision และค่า Recall เป็นคู่ของตัวเลขระหว่าง **RECALL** และ **PRECISION** สามารถนำมาเปรียบเทียบสำหรับการค้นหาครั้งที่ **i** และ **j** เมื่อใดก็ตามที่

RECALLi < **RECALLj**

และ **PRECISIONi** < **PRECISIONj**

ผลลัพธ์ของการค้นหาครั้งที่ **j** จะถูกพิจารณาว่า ดีกว่าการค้นหาครั้งที่ **i** แต่จะมีปัญหาเกิดขึ้นเมื่อ

RECALLi < **RECALLj**

และ **PRECISIONi** > **PRECISIONj**

หรือในทางตรงข้ามกันจะได้

RECALLi > **RECALLj**

และ **PRECISIONi** < **PRECISIONj**

ในกรณีดังกล่าวการพิจารณาว่าในการค้นหาครั้งใดจะดีกว่ากันนั้นจะขึ้นอยู่กับผู้ใช้งาน โดยผู้ใช้จะต้องกำหนดว่าสิ่งที่ผู้ใช้งานสนใจเป็นค่า Recall หรือค่า Precision และประเมินความสำคัญของความแตกต่างระหว่างค่าของ Recall และ Precision ในระบบค้นคืน สารสนเทศที่เป็นมาตรฐานนั้น ค่า Recall จะแปรผกผันค่า Precision ทำให้เกิดการพิจารณาค่าที่สมดุลกันระหว่างค่าทั้งสองเรียกว่า **“The Weighted Harmonic Mean of Precision and Recall”** [14] หรือค่าเอฟ (F-Measure) ดังสมการ

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

$$F = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

โดยที่

$$\beta^2 = \frac{1-\alpha}{\alpha}$$

เมื่อ $\alpha \in [0,1]$ และดังนั้น $\beta^2 \in [0,\infty]$ โดยทั่วไปเพื่อทำให้เกิดความสมดุลของสมการค่าเอฟ เพื่อกำหนดค่าน้ำหนักของ Precision และ Recall กำหนดให้ $\alpha = {}^{1}/_{2}$ หรือ $\beta = 1$ และสามารถแสดงเป็นสูตรอย่างง่ายดังนี้

$$F_{\beta=1} = \frac{2PR}{P+R} \qquad (2.4)$$

นอกจากการพิจารณาค่าน้ำหนักดังกล่าวยังสามารถกำหนดค่า $\beta < 1$ เมื่อต้องการเน้นผลลัพธ์ค่า Precision หรือกำหนดค่า $\beta > 1$ เมื่อต้องการเน้นผลลัพธ์ค่า Recall ซึ่งอาจจะกำหนดเป็นค่า $\beta = 3$ หรือ $\beta = 5$ โดยพิจารณาตามความเหมาะสม

2.1.4 เอกซ์เอ็มแอล (XML : The Extensible Markup Language) [9, 10, 11, 20]

เอกซ์เอ็มแอล เป็นภาษามาร์กอัปที่ออกแบบมาเพื่อให้สามารถนิยามความหมายของข้อมูล หรือที่เรียกว่า นิยามข้อมูล (Data Definition) โดยผู้ใช้งานสามารถสร้างแท็กขึ้นเองได้ ซึ่งแท็กที่สร้างขึ้นเองนั้นจะเป็นมาตรฐานที่ผู้ใช้กำหนดไว้ และเป็นเทคโนโลยีที่ไม่ขึ้นกับแพลตฟอร์มใดๆ รองรับการใช้งานจากภาษาคอมพิวเตอร์ที่หลากหลาย เช่น ASP, VB, PHP, JavaScript, MS .Net และ Java เป็นต้น เนื่องจากแท็กที่ผู้ใช้สร้างขึ้นไม่ได้ทำหน้าที่เพื่อแสดงข้อมูลเท่านั้นแต่ทำหน้าที่ระบุขอบเขตของข้อมูล นอกจากนี้เอกซ์เอ็มแอลสามารถนำมาใช้งานเป็นมาตรฐานร่วมกันทางด้านข้อมูลเอกสาร ทำให้กำจัดปัญหาการใช้งานร่วมกันระหว่างฐานข้อมูลกับเอกสาร

เอกสารเอกซ์เอ็มแอลจะมีการอธิบายข้อมูลในส่วนรูปแบบที่เป็นลำดับชั้น (Hierarchy) เหมือนต้นไม้ (Tree) ทำให้สามารถสร้างโปรแกรมประยุกต์เพื่อทำการประมวลผลเอกสารและต่อประสาน (Binding) กับเอกสารเอชทีเอ็มแอล (HTML) ในการแสดงผลบนเว็บบราวเซอร์ได้อย่างอิสระ ผู้ที่ทำหน้าที่รับผิดชอบและกำหนดมาตรฐานของเอกซ์เอ็มแอล คือ World Wide Web Consortium (W3C)

เอกซ์เอ็มแอลประกอบด้วยส่วนหลัก คือ

- ตัวเอกสารเอกซ์เอ็มแอลซึ่งเป็นโครงสร้างทางตรรกะ (Logical Structure) อธิบายคุณลักษณะต่างๆ ของข้อมูลที่บรรจุอยู่ในเอกสารในลักษณะโครงสร้างลำดับชั้น ดังในภาพที่ 2.4
- การกำหนดกฎและรายละเอียดของเนื้อหาเอกสารหรือที่เรียกว่า ดีทีดี (DTD: Document Type Declaration) และเอกซ์เอ็มแอลสกีมา (XML Schema) ทำหน้าที่ในการกำหนดไวยากรณ์ของเอกสารเอกซ์เอ็มแอลที่มีรูปแบบถูกต้อง (Well-formed document) ดังในภาพที่ 2.5

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>

<shiporder orderid="889923"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

ภาพที่ 2.4 ตัวอย่างข้อมูลในไฟล์ XML ชื่อ "shiporder.xml" [10]

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
            <xs:element name="country" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="note" type="xs:string" minOccurs="0"/>
            <xs:element name="quantity" type="xs:positiveInteger"/>
            <xs:element name="price" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>
```

ภาพที่ 2.5 ตัวอย่าง XML Schema ชื่อ "shiporder.xsd" [10]

มาตรฐานของ XML

- ดีทีดีและเอกซ์เอ็มแอลสคีมา เป็นมาตรฐานที่ใช้ในการกำหนด หรืออธิบายโครงสร้างของเอกสาร เช่น การกำหนดแท็กว่าควรมีอะไรบ้าง หรือ การกำหนดแอตทริบิวต์ (Attribute) ที่ต้องมีอะไรบ้าง เป็นต้น
- เอกซ์เอสแอลที (XSLT: eXtensible Stylesheet Language Transformations) เป็นมาตรฐานที่เกี่ยวข้องกับการเปลี่ยนแปลงเอกสารไปเป็นเอกสารชนิดอื่น ๆ XSLT เป็นองค์ประกอบหนึ่งของเอกซ์เอสแอล ซึ่งประกอบด้วยส่วนต่างๆ ได้แก่
  - XSLT ใช้ในการเปลี่ยนเอกสารไปเป็นรูปแบบอื่นๆ
  - XPath ใช้ในการเข้าถึง อีลิเมนต์ (element) ในเอกสารเอกซ์เอ็มแอล
  - XSL-FO ใช้ในการกำหนดรูปแบบการแสดงผล
- เอกซ์พาธ (XPath) เป็นมาตรฐานที่ใช้ในการระบุตำแหน่งต่าง ๆ ของเอกสารเอกซ์เอ็มแอล โดยใช้ในการเข้าถึงโหนดจุดต่อ (Node) ของเอกสารเอกซ์เอ็มแอลนั้น
- เอกซ์คิวรี (XQuery) เป็นภาษาที่ใช้ในการสืบค้นข้อมูลในเอกสารเอกซ์เอ็มแอล โดยเป็นการมองเอกสารเอกซ์เอ็มแอลเป็นเหมือนฐานข้อมูลตัวหนึ่ง โดยบทบาทของเอกซ์คิวรี ที่มีต่อเอกสารเอกซ์เอ็มแอลเหมือนกับบทบาทของภาษาเอสคิวแอล ที่มีต่อฐานข้อมูล ในการประมวลผลฝั่งเซิร์ฟเวอร์
  - DOM และ SAX เป็นวิธีการสำรวจข้อมูลในเอกสารเอกซ์เอ็มแอล

กฎไวยากรณ์ของเอกซ์เอ็มแอล

- เอกสารเอกซ์เอ็มแอลจะมี root element ได้เพียงหนึ่งเดียวเท่านั้น
- ชื่อแท็กเปิด และแท็กปิดจะเหมือนกันเพียงแต่แท็กปิดจะมีเครื่องหมาย "/" นำหน้า
- ไม่สามารถให้มีการ ซ้อนเหลื่อมกันของแท็ก (Overlap) คือแท็กเปิดก่อนต้องปิดหลังเสมอ
- ชื่อแท็กมีคุณสมบัติ case-sensitive คือ ตัวพิมพ์เล็ก-ตัวพิมพ์ใหญ่ ถือว่าเป็นคนละแท็กกัน
- สำหรับแท็กว่าง สามารถเขียนได้ 2 แบบ คือ<tagName></tagName>และ <tagName />
- ค่าข้อมูลของแอตทริบิวต์ต้องอยู่ในเครื่องหมาย Double Quote หรือ Single Quote เท่านั้น
- ภาษาเอกซ์เอ็มแอลมีอักขระที่สงวนไว้ 5 ตัว คือ <, >, &, ", ‘ จึงต้องใช้ชุดตัวอักษรพิเศษแทนอักขระเหล่านี้
- การตั้งชื่อแท็กนั้น ตัวอักษร 3 ตัวแรกห้ามเป็นเอกซ์เอ็มแอลนำหน้า

โครงสร้างเอกสารเอกซ์เอ็มแอล

- Prolog แบ่งออกเป็น 2 ส่วนย่อย คือ xml declaration และ document type declaration
- Body เป็นส่วนของเนื้อหาเอกสาร คือส่วนของ root element
- Epilog มี 2 ประเภท คือ comment และ pi

2.1.5 ภาษาเอกซ์คิวรี (XQuery : XML Query Language) [10, 12]

เอกซ์คิวรีเป็นภาษาสืบค้นสำหรับเอกสารเอกซ์เอ็มแอล ซึ่งได้รับอิทธิพลมาจากภาษาสืบค้นสำหรับเอกซ์เอ็มแอล ในรุ่นแรกๆ การออกแบบภาษาโดยยึดหลักทำให้สั้นกระชับและสามารถทำความเข้าใจได้ง่าย ใช้งานร่วมกันทั้งแบบฐานข้อมูลและเอกสาร และมีการพัฒนาร่วมกันกับมาตรฐานอื่นๆที่เกี่ยวข้อง ได้แก่ เอกซ์พาธ (XPath), เอกซ์คิวแอล (XQL), เอสคิวแอล (SQL) โอคิวแอล (OQL) เป็นต้น เอกซ์คิวรีเป็นภาษาที่ขยายต่อจากเอกซ์พาธ ดังนั้นจึงใช้แนวคิดของนิพจน์พาธ ในการระบุไปยังข้อมูลตำแหน่งต่างๆ ในเอกสารที่กำลังสนใจ นิพจน์พาธของเอกซ์คิวรีจะอ้างอิงมาจากมาตรฐานของเอกซ์พาธเสริมด้วยคำสั่งหรือฟังก์ชันบางส่วนที่ขอยืมมาจากเอกซ์คิวแอล นิพจน์พาธเป็นวิธีที่สะดวกและมีประสิทธิภาพมากในการระบุไปยังข้อมูลที่สนใจในโครงสร้างที่ซับซ้อนของเอกสาร ตัวอย่าง นิพจน์พาธ เช่น document("bib.xml")/bib/book หมายถึง element ที่ชื่อ book ทุกๆ อันที่อยู่ภายใต้ root element ที่ชื่อ bib ของเอกสาร bib.xml

โครงสร้างการทำงานของเอกซ์คิวรีจะประกอบไปด้วยลำดับของประโยค **FOR - LET - WHERE - RETURN** ซึ่งได้รับอิทธิพลมาจากภาษาเอสคิวแอล ที่เป็นลำดับ**ประโยค SELECT - FROM - WHERE,** โดยประโยค FOR จะระบุไปยังข้อมูลที่กำลังสนใจโดยอาศัยนิพจน์พาธและนำข้อมูลนั้นมากำหนดให้กับตัวแปร ในแต่ละรอบของการทำงาน, ประโยค LET จะใช้ในการกำหนดค่าให้กับตัวแปรเพื่อช่วยเสริมการทำงานกับประโยค FOR, ประโยค WHERE จะตรวจสอบเงื่อนไขของข้อมูลที่เก็บอยู่ใน ตัวแปรที่ได้มาจากในส่วนของประโยค FOR, LET ที่อยู่ข้างต้นและส่งผลลัพธ์ที่ตรงตามเงื่อนไขให้กับส่วนประโยค RETURN เพื่อสร้างผลลัพธ์ที่ต้องการ

ตัวอย่างการใช้งานเอกซ์คิวรีเพื่อเรียกค้นข้อมูล ดังนี้

```
<bib>
 {
        for $b in doc("http://bstore1.example.com/bib.xml")/bib/book
         where $b/publisher = "Addison-Wesley" and $b/@year > 1991
         return
        <book year="{ $b/@year }">
                { $b/title }
        </book>
 }
</bib>
```

ภาษาเอกซ์คิวรีประกอบด้วยการใช้งานประโยคคำสั่งต่างๆ เช่น การเรียกใช้งานฟังก์ชัน การนิยามฟังก์ชันจะเป็นแบบเดียวกันกับที่มีในภาษาโอคิวแอล ซึ่งเอกซ์คิวรีนั้นสามารถนับได้ว่าเป็นภาษาสืบค้นสำหรับเอกซ์เอ็มแอลที่มีประสิทธิภาพมาก ที่ผ่านมาเอกซ์คิวรีนั้นถูกพัฒนามาในแนวทางของฐานข้อมูล จึงทำให้สามารถจัดการกับข้อมูลที่อยู่ในรูปของฐานข้อมูลได้ดี แต่สำหรับข้อมูลที่อยู่ในรูปของเอกสารนั้นยังมีข้อจำกัดอยู่บางประการ

จากตัวอย่าง เป็นการหารายชื่อหนังสือและปีที่พิมพ์ โดยที่หนังสือเล่มนั้นๆ ต้องพิมพ์โดยสำนักพิมพ์ Addison-Wesley หลังจากปี 1991, ประโยค FOR จะกำหนดค่าของแต่ละ element ที่ชื่อ **book** ที่คืนมาจากนิพจน์พาธ **document("bib.xml")/bib/book** มาเก็บไว้ในตัวแปร **$b,** ต่อจากนั้นประโยค WHERE จะตรวจสอบเงื่อนไขของค่าที่เก็บอยู่ใน ตัวแปร **$b** ว่าเป็นไปตามที่ต้องการ ในที่นี้คือมีค่าของ element publisher เป็น **"Addison-Wesley"** และค่าของ attribute **year** มากกว่า **1991** (เครื่องหมาย **@** แสดงถึงการเป็นแอตทริบิวต์), หลังจากนั้นประโยค RETURN ก็จะอาศัยค่าตัวแปร $b เพื่ออ้างอิงถึง ค่าของ attribute **year** และ element **title** ในการสร้างผลลัพธ์ที่ต้องการ

2.1.6 ภาษาเอกซ์พาธ (XPath : XML Path Language) [9, 10]

เป็นภาษาสืบค้นสำหรับระบุตำแหน่งของส่วนต่างๆในเอกสารเอกซ์เอ็มแอล ซึ่งถูกออกแบบมาให้ใช้กับเอกซ์เอสแอลที และเอกซ์พอยต์เตอร์ (XPointer) เอกซ์พาธมีลักษณะเป็นภาษาประกาศที่ระบุตำแหน่งของโหนดต่างๆ เช่น อีลิเมนต์โหนดและแอตทริบิวต์ในเอกสารเอกซ์เอ็มแอล และเป็นภาษาที่คล้ายกับภาษาเอสคิวแอล (SQL) ในด้านที่สามารถนำไปใช้ดึงข้อมูลเฉพาะบางข้อมูลออกมาได้ แต่แตกต่างกันที่เอสคิวแอลใช้กับดึงข้อมูลจากฐานข้อมูลแต่เอกซ์พาทใช้ดึงข้อมูลจากเอกสารเอกซ์เอ็มแอล

<u>ดาตาโมเดลของเอกซ์พาธ</u>

การระบุตำแหน่งของโหนดต่างๆในเอกสารเอกซ์เอ็มแอลโดยการใช้โมเดลโครงสร้างต้นไม้ (Tree Structure Model) ประกอบด้วยโหนดต่างๆ ดังนี้

- รูทโหนด (root nodes)
- อีลิเมนต์โหนด (element nodes)
- เท็กซ์โหนด (text nodes)
- แอตทริบิวต์โหนด (attribute nodes)
- เนมสเปซโหนด (space nodes)
- โหนดในการประมวลคำสั่ง (processing instruction nodes)
- คอมเมนต์โหนด (comment node)

นิพจน์ของเอกซ์พาธ

นิพจน์ของเอกซ์พาธประกอบด้วย โลเคชันพาธ (Location path) การเรียกฟังก์ชัน (Function calls) ชุดของโหนด (Node-set) ค่าบูลีน (Booleans) ตัวเลข (Numbers) และสายอักขระ (String) โดยส่วนที่ใช้งานบ่อยและสำคัญที่สุดคือ โลเคชันพาธ

โลเคชันพาธ

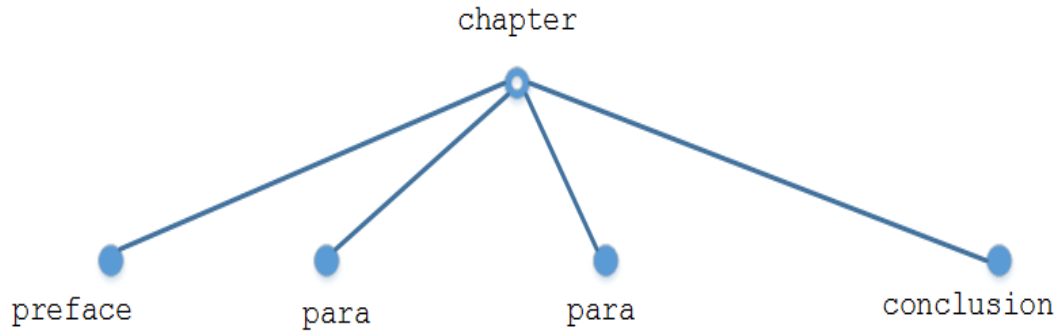
ชุดของโหนดเอกสารเอกซ์เอ็มแอลที่แต่ละโลเคชันพาธประกอบด้วย โลเคชันสเต็ป (Location step) ซึ่งในแต่ละโลเคชันสเต็ปจะประกอบด้วย เอซิส (Asis) การตรวจสอบโหนดทดสอบ (Node test) และส่วนของเงื่อนไขที่โหนดนั้นจะต้องดำเนินการตาม (Predicates) โดยที่จะมีหรือไม่มีส่วนนี้ก็ได้ แต่ละโลเคชันสเต็ปจะถูกประมวลผลตามโหนดบริบท (Context node) ซึ่งจะหมายถึงโหนดที่ถูกประมวลผลเสร็จในขณะนั้น โดยใช้สัญลักษณ์ :: เป็นตัวแยกระหว่างเอซิสกับการตรวจสอบโหนด และใช้สัญลักษณ์ [ ] ครอบส่วนที่เป็นเงื่อนไข ตัวอย่างแสดงการเขียนโลเคชันสเต็ปดังนี้

```
axis::node test[ predicate ]
```

แทนด้วย

```
child::para[position()=1]
```

โดยที่ **child** เป็นชื่อของเอซิส (the name of the axis) **para** เป็นสิ่งที่ตรวจสอบว่าโหนดนั้นมีชื่อว่า “para” หรือไม่ node test และ position=1 เป็นเงื่อนไขที่จะกำหนดตำแหน่งของโหนดเป็นตำแหน่งแรก โดยสรุปหมายถึง โลเคชันพาธที่เลือกโหนดลูกของโหนดบริบท โดยที่โหนดลูกจะต้องเป็นชื่อ “para” ซึ่งโหนด **para** จะต้องเป็นโหนดแรก ดังแสดงในภาพที่ 2.6



ภาพที่ 2.6 ตัวอย่างองค์ประกอบโครงสร้างเอกสารเอกซ์เอ็มแอลเพื่อใช้ประมวลผลด้วยเอกซ์พาธ [9]

จากภาพที่ 2.6 สมมติว่าโหนดบริบทคือโหนด **chapter** ถ้าระบุเอกซ์พาธเป็น child::para[position()=1] แต่ในความเป็นจริงแล้วโหนดที่เป็นโหนดลูกจะมี 4 โหนดคือ **preface, para, para** และ **conclusion** ดังนั้นเมื่อระบุเอกซ์พาธเป็น child::para จะได้ 2 โหนดเท่านั้นที่เป็นคำตอบ เพราะโหนด **preface** กับ โหนด **conclusion** จะไม่ตรงกับเงื่อนไข

ในตารางที่ 2.1 เป็นการแสดงรูปแบบนิพจน์เอกซ์พาธที่มีการใช้งานแบบย่อและรูปแบบเต็มเพื่อใช้ในการเข้าถึงโหนดข้อมูลเอกซ์เอ็มแอล

ตารางที่ 2.1 ไวยากรณ์แบบย่อและแบบเต็มของภาษาเอกซ์พาธ [9]

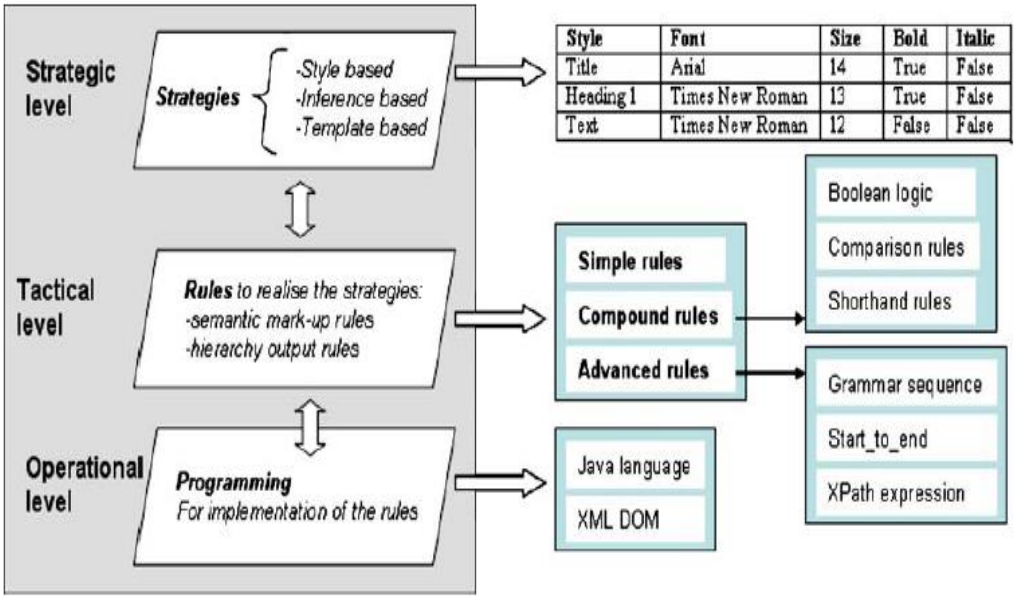
รูปแบบย่อ	รูปแบบเต็ม	ความหมาย
nodeName	child::*nodeName*	เลือกโหนดลูกที่ชื่อ *nodeName*
@attributeName	attribute::*attributeName*	เลือกแอตทริบิวต์โหนดชื่อ *attributeName*
//	/descendant-or-self::node()/	เลือกโหนดหลานของโหนดบริบทและโหนดบริบทเอง
.	self::node	เลือกโหนดบริบทตัวเอง
..	parent::node()	เลือกโหนดพ่อแม่ของโหนดบริบท

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 An automatic mark-up approach for structured document retrieval in engineering design [1]

เป็นงานวิจัยที่เกี่ยวกับการจัดทำเอกสารอัตโนมัติด้วยการค้นคืนเอกสารที่เป็นโครงสร้าง ในการออกแบบทางวิศวกรรม ซึ่งเป็นงานที่นำเสนอความรู้ทางด้านวิศวกรรมเพื่อใช้ในการทำมาร์กอัป (Mark-up) เอกสารเอกซ์เอ็มแอลโดยการกำหนดแท็กเพื่อระบุไว้ในโครงสร้างเอกสารสารสนเทศ ในงานวิจัยนี้จะใช้กับเอกสารที่อยู่ในงานทางด้านวิศวกรรม หลักการทำงานประกอบด้วยโมเดล 3 ระดับ เพื่อทำให้เกิดการค้นหาข้อมูลในแบบซีแมนทิคมาร์กอัป ประสบความสำเร็จ โดยพิจารณาจากกลุ่มของโครงสร้างข้อมูลเอกสารที่ถูกแยกส่วนประกอบ (Document Decomposition Schema) ดังนี้

- ระดับกลยุทธ์ (Strategic Level) คือ การกำหนดให้มีประเภทเอกสารเป็นแบบภาพกราฟิก โดยมีพื้นฐานมาจากสิ่งต่างๆ เช่น รูปแบบ (Styles)
- ระดับกลวิธี (Tactical Level) คือ การนิยามกฎที่จะใช้ในแบบซีแมนทิคมาร์กอัป เพื่อให้สอดคล้องกับคุณลักษณะของเอกสาร
- ระดับปฏิบัติการ (Operation Level) คือ สามารถทำการประมวลผลกฎมาร์กอัปที่นิยามไว้และนำไปใช้ได้



ภาพที่ 2.7 โมเดลสามระดับของมาร์กอัปอัตโนมัติ [1]

จากภาพที่ 2.7 เป็นการสรุปแนวคิดของงานวิจัยดังกล่าวเพื่อใช้ในการพัฒนาและการนำไปใช้ของกฎมาร์กอัป ที่ระดับสูงสุดเป็นการพิจารณาจัดทำกลยุทธ์ของกฎมาร์กอัปที่มาจากสิ่งที่เป็นพื้นฐานของเอกสาร เช่น รูปแบบ (Style), แผ่นแบบ (Template) และการอนุมาน (Inference) จากภาพที่แสดงรูปแบบบางส่วนที่อยู่ในเอกสารเวิร์ด เช่น ชื่อเรื่อง (Title) จะเป็นแบบอักขระ Arial ขนาด 14 pt. เป็นแบบตัวหนา (Bold) และจะทำการแปลงเอกสารเวิร์ดเป็นพรีพรอเซสเอกซ์เอ็มแอล (ppXML) โดยจะมีเก็บรูปแบบของข้อมูลเป็นแบบคุณลักษณะ/ค่า (attribute/value) ระดับกลวิธีเป็นการพัฒนากฎมาร์กอัปจากกลุ่มของกลยุทธ์ต่างในระดับสูงก่อนหน้านี้ โดยนิยามกฎไว้เป็นแบบซีแมนทิคมาร์กอัป ประกอบด้วย 3 แบบดังนี้

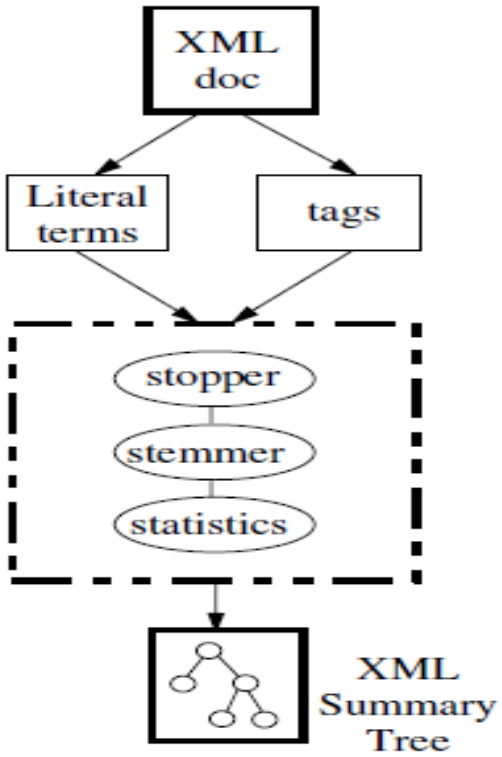
- กฎพื้นฐานอย่างง่าย (Simple rules) คือ การใช้เพียงกฎเดียว
- กฎแบบผสม (Compound rules) คือ การใช้กฎร่วมกันมากกว่า 1 กฎ เช่น เงื่อนไขบูลีน (AND, OR, NOT) การเปรียบเทียบ (equals, less than, greater than, true, false)
- กฎแบบขั้นสูง (Advanced rules) คือ การใช้กฎในแบบเป็นจัดลำดับ (Sequences) เช่น จัดลำดับแกรมม่า (grammar sequence), จุดเริ่ม-จุดสุดท้าย (Start-to-end) และนิพจน์เอกซ์พาธ (XPath expression)

ผลลัพธ์ที่ได้จากงานวิจัยนี้มี 3 ส่วนดังนี้

- รวบรวมคำศัพท์และดีทีดี เพื่อใช้แทนเนื้อหาของเอกสารด้านวิศวกรรมที่ถูกแยกส่วนประกอบเอกสาร
- สามารถทำมาร์กอัปจากองค์ประกอบเอกสารทั้งในระดับจุลภาค (Micro-level) จากประโยคไปสู่คำและระดับมหภาค (Macro-level) จากย่อหน้า (Paragraph) ไปสู่เซคชั่นย่อยหรือเซคชั่นหลัก
- สามารถแปลความหมายขององค์ประกอบที่เป็นตรรกะของเนื้อหาจากสิ่งต่างๆ เช่น บทนำ (Introduction) ความรู้พื้นฐาน (Background) และข้อสรุป (Conclusion)

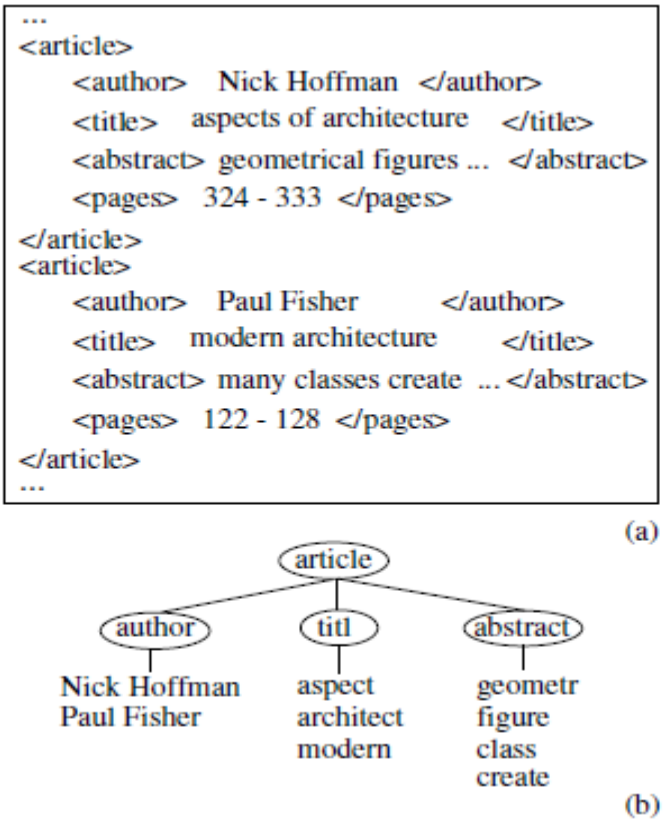
2.2.2 Structured Information Retrieval in XML documents [2]

งานวิจัยนี้นำเสนอโครงสร้างดัชนีของพจนานุกรม (Lexicography Index) เพื่อนำไปใช้ในการประเมินผลการสืบค้นที่เกี่ยวกับนิพจน์พาธ (Path Expression) และงานวิจัยนี้ยังกล่าวถึงการจัดอันดับสกีมาที่ขึ้นอยู่กับการกระจายและโครงสร้างเอกสาร โครงสร้างดัชนีจะรวมอยู่ในไฟล์ที่ถูกทำย้อนกลับ (Inverted File) ด้วยลำดับของดัชนีพาธเพื่อนำมาใช้ในการประเมินผลบูลีนคิวรี (Boolean Queries) ในการค้นหาจากโครงสร้างของเนื้อหาเอกสารเอกซ์เอ็มแอล เมื่อความคล้ายกันหรือการจัดลำดับอาจจะเกิดจากคำสำคัญที่ใช้ในการค้นหา



ภาพที่ 2.8 กระบวนการทำนอร์มอไลเซชันเอกสารเอกซ์เอ็มแอล [2]

ภาพที่ 2.8 แสดงขั้นตอนการระบุสิ่งที่จะถูกทำเป็นดัชนีของเอกสารเอกซ์เอ็มแอลด้วยอัลกอริทึมนอร์มอไลเซชัน โดยทำการกำจัดสิ่งที่ไม่สำคัญออก เช่น Stop Word ซึ่งจะได้คำเหล่านี้จากกลุ่มคำที่ถูกบันทึกไว้ เมื่อพบว่าไม่ใช่ Stop Word จะถูกนำไปบันทึกเก็บไว้ด้วยอัลกอริทึม Stemming ผลที่ได้จะเป็นพาธที่สอดคล้องกัน และขั้นตอนสุดท้ายจะเป็นการประเมินผลการกระจาย (จำนวนความถี่ที่พบในเอกสาร) ของสิ่งที่ถูกพิจารณา เช่น literal term หรือแท็ก



ภาพที่ 2.9 (a) ตัวอย่างเอกสารเอกซ์เอ็มแอล (b) สรุปผลลัพธ์เป็นโครงสร้างต้นไม้ [2]

จากภาพที่ 2.9 (b) แสดงผลลัพธ์พาธเป็นโครงสร้างต้นไม้ที่ถูกวิเคราะห์จากตัวอย่างเอกสารเอกซ์เอ็มแอล ดังแสดงในภาพที่ 2.9 (a) จะเห็นว่า คำหรือแท็กที่ไม่ถูกทำดัชนี เช่น แท็ก <page> จะไม่ถูกนำมาไว้ในผลสรุปของต้นไม้

ประโยชน์ที่ได้จากดัชนีสามารถสรุปได้ ดังนี้

- เป็นการรวมกันอย่างดีของสองดัชนี คือ ไฟล์ย้อนกลับและพาธดัชนี
- พาธดัชนีจะอยู่ในแท็กที่ถูกทำนอร์มอไลซ์ เพื่อใช้ในการค้นหาความคล้ายกันของเนื้อหาและโครงสร้าง
- ทั้งพาธดัชนีและไฟล์ย้อนกลับจะถูกขยายเพิ่มหรือลดขนาด จากการเพิ่มขึ้นหรือหายไปของเอกสารเอกซ์เอ็มแอล

2.2.3 Comparative Study of Clustering Techniques for Short Text Documents [4]

งานวิจัยที่นำเทคนิค K-means, SVD-based และ Graph-based มาเปรียบเพื่อวัดประสิทธิภาพการจัดกลุ่มเอกสาร ด้วยการรวบรวมข้อมูลที่มีลักษณะข้อความสั้นๆของทวิตเตอร์ ในการประเมินผลของงานวิจัยนี้จะใช้การวัดค่าความผิดพลาดของแต่ละเทคนิคดังกล่าว เพื่อหาประเภทของเทคนิคที่มีค่าความผิดพลาดต่ำที่สุด ในการจัดการเอกสารที่มีลักษณะเป็นข้อความสั้นๆ จะพิจารณาจากลักษณะความหนาแน่นของกลุ่มคำ โดยนำเทคนิคที่เรียกว่า TF-IDF : Term Frequency–Inverse Document Frequency แทนพิกัดตำแหน่งเวกเตอร์ของข้อมูลในเอกสารด้วยปริภูมิ d มิติ (d-Dimensional Space) เมื่อ d คือ ขนาดของคำศัพท์ เอกสาร doc_i แทนด้วย $(v_1, v_2, \dots v_d)$ เมื่อ v_j เป็น TF-IDF พิกัดตำแหน่งของคำที่ j^{th} ในเอกสารนั้นๆ งานวิจัยนี้จำกัดจำนวนของเอกสารที่จะใช้ในการพิจารณาไม่เกิน 3 เอกสารโดยใช้อัลกอริทึม Graph-based มาใช้ในการวัดค่าระยะความคล้ายกันของเอกสาร โดยพิจารณาจากการเชื่อมต่อกันในแต่ละโหนดของกราฟ

งานวิจัยนี้ทำการทดลองกลุ่มข้อมูลทวิตเตอร์ โดยกำหนดให้ข้อความที่ถูกทวิตในแต่ละครั้งคือหนึ่งเอกสาร และมีการวิเคราะห์ความแตกต่างกันของหัวข้อ เช่น Programmer Language, Computer Network, Cricket เป็นต้น สามารถแยกข้อมูลหัวข้อได้ประมาณ 1,678 คำหลังจากที่ได้ทำการแยกกลุ่มคำที่ไม่สำคัญออก (Stop Word) ผลลัพธ์ที่ได้จากงานวิจัยนี้พบว่าในแต่ละเทคนิคของการจัดกลุ่มเอกสารนั้น เทคนิค Graph-based ให้ค่าความผิดพลาดต่ำสุดที่ 2.95% ถัดมาคือ K-means และ SVD-based ตามลำดับ เมื่อพิจารณาเอกสารที่มีหัวข้อเด่นคาบเกี่ยวกันมากกว่าหนึ่งหัวข้อจะให้ค่าความผิดพลาดสูงมากกว่าเอกสารที่มีหัวข้อเด่นแบบเดียว

2.2.4 A Vector Space Model for Automatic Indexing [7]

งานวิจัยที่นำเสนอการค้นคืนเอกสารด้วยเทคนิคปริภูมิเวกเตอร์โมเดล โดยใช้การวิเคราะห์ข้อมูลดัชนีคำศัพท์ (Indexing Vocabulary) เพื่อสร้างต้นแบบในการหาความสัมพันธ์ของเอกสารที่เกี่ยวข้อง ใช้เทคนิคในการประมวลผลแบบไม่ต้องรู้ข้อมูลล่วงหน้าใช้การดำเนินการไปข้างหน้า (Straightforward) ด้วยการกำจัดคำที่ไม่เกี่ยวข้องออก ทำการเก็บข้อมูลของคำที่เกี่ยวข้องและพิจารณาให้ค่าน้ำหนักจากจำนวนความถี่ของคำที่พบในเอกสาร ผลลัพธ์ที่ได้ของงานวิจัยนี้เป็นการวิเคราะห์หาความสัมพันธ์ของเอกสารโดยใช้ดัชนีคำแบบอัตโนมัติ ซึ่งมีการกำหนดโครงแบบของการวิเคราะห์เป็นต้นแบบจากข้อมูลคำดัชนีต่างๆในเอกสารที่เกี่ยวข้อง และเมื่อกำหนดการพิจารณาให้ค่าน้ำหนักในแต่ละคำเพื่อนำมาวิเคราะห์เพิ่มเติมพบว่าจะให้ค่าความคล้ายกันของเอกสารที่เกี่ยวข้องเพิ่มมากขึ้น

บทที่ 3
วิธีดำเนินการวิจัย

3.1 แนวคิดวิธีการดำเนินการวิจัย

งานวิจัยนี้เกิดจากแนวคิดที่ต้องการรวบรวมข้อมูลสารสนเทศที่เกี่ยวข้องสัมพันธ์กับความต้องการของผู้ใช้งานจากแหล่งข้อมูลต่างๆ และถูกนำมาแสดงผลให้กับผู้ใช้ตามที่ผู้ใช้ได้ออกแบบไว้ ระบบที่อยู่ภายใต้แนวคิดนี้จะต้องสามารถรวบรวมและจัดเก็บข้อมูลต่างๆไว้ แหล่งข้อมูลจะมีรูปแบบและโครงสร้างที่หลากหลาย ซึ่งจะถูกนำมาวิเคราะห์เนื้อหาของข้อมูลเพื่อจัดลำดับความคล้ายและทำการสกัดบริบทที่ตรงกับความต้องการผู้ใช้ โดยแบ่งแนวคิดออกเป็น 2 กลไกหลักดังนี้

<u>กลไกการค้นคืนเอกสาร</u> (Document Retrieval Mechanism)

เริ่มต้นจากการเตรียมข้อมูล โดยการเตรียมเอกสารที่จะนำมาใช้ในการประมวลผลเพื่อตรวจสอบ เปลี่ยนแปลงและกำหนดให้โครงสร้างเอกสารอยู่ในรูปแบบเดียวกันคือเอกซ์เอ็มแอล เนื่องจากรูปแบบเอกสารในปัจจุบันมีหลากหลายประเภท เช่น พรีพรอเซสเอกซ์เอ็มแอล (ppXML) เอกสารไมโครซอฟต์เวิร์ด และเอกสารโอเพนออฟฟิส เป็นต้น ดังนั้นงานวิจัยนี้จึงกำหนดให้ใช้รูปแบบมาตรฐานเอกสารเอกซ์เอ็มแอลเท่านั้น เนื่องจากในขั้นตอนถัดไปของงานวิจัยนี้จะใช้ภาษาเอกซ์คิวรีเพื่อสืบค้นข้อมูลเอกซ์เอ็มแอล ซึ่งในการดำเนินการของขั้นตอนการรวบรวมข้อมูลใช้ตัวอย่างโครงสร้างข้อมูลเอกสาร .docx และจะถูกเปลี่ยนแปลงเป็นรูปแบบเอกซ์เอ็มแอล ในการวิเคราะห์รวบรวมคำสำคัญและคำที่เกี่ยวข้อง เพื่อใช้ในการค้นหา ในการเตรียมข้อมูลคำสำคัญจะถูกนำมาใช้วิเคราะห์หาคำศัพท์ที่มีความหมายใกล้เคียงหรือเหมือนกัน (Synonym หรือ Thesaurus) จากฐานข้อมูลคำศัพท์ออนไลน์ (Dictionary Online) [16] เพื่อนำไปใช้ค้นหาข้อมูลคำศัพท์ที่มีความหมายใกล้เคียงและผลลัพธ์ชุดข้อมูลดังกล่าวซึ่งเรียกว่า คำศัพท์ควบคุม (Controlled Vocabulary) จะถูกนำไปประมวลผลความคล้ายด้วย สาเหตุที่ต้องใช้คำที่เกี่ยวข้องมาประมวลผลร่วมกับคำสำคัญเพื่อจะได้ผลลัพธ์ของบริบทที่มีความใกล้เคียงกับคำที่ค้นหามากยิ่งขึ้น เนื่องจากในแต่ละเอกสารอาจจะมีการกล่าวถึงเนื้อหาเดียวกันแต่ใช้คำศัพท์ที่ต่างกัน

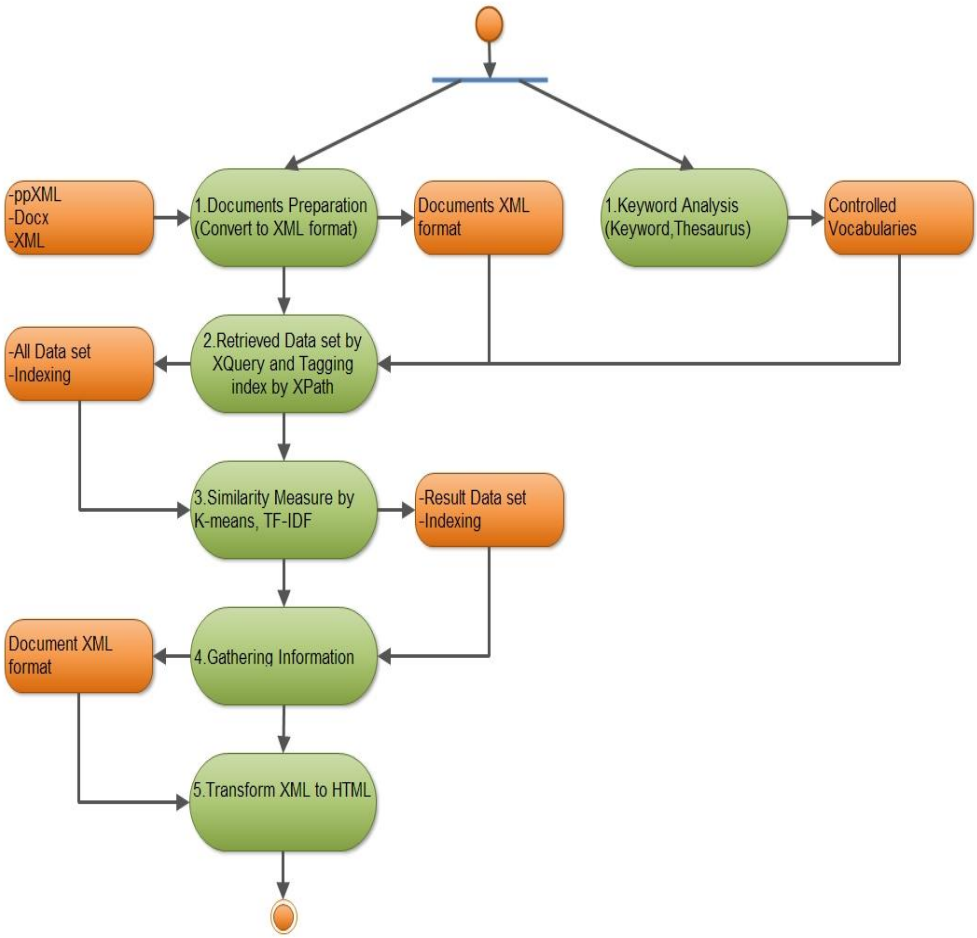
หลังจากนั้นจะทำการรวบรวมข้อมูลและแท็กดัชนี (Collected Data set by XQuery and Tagging index by XPath) โดยเริ่มจากการเก็บชุดข้อมูลของคำสำคัญและคำที่เกี่ยวข้องจากเอกสารที่ถูกนำมาประมวลผลด้วยวิธีการสืบค้นข้อมูลเอกซ์เอ็มแอลด้วยภาษาที่เรียกว่าเอกซ์คิวรี ผลลัพธ์ที่ได้จากการสืบค้นนี้จะเป็นชุดข้อมูลเพื่อนำไปใช้ในการวิเคราะห์และวัดค่าความคล้ายของเอกสารต่อไป ซึ่งในการสืบค้นแบบเอกซ์คิวรี จะใช้วิธีการค้นหาไปยังโหนดต่างๆ ที่เป็นส่วนประกอบของเอกสารโครงสร้างเอกซ์เอ็มแอล ดังนั้นเพื่อให้การค้นหาข้อมูลสามารถแสดงผลรวดเร็วขึ้นจะมีขั้นตอนการกำหนดแท็กดัชนีไว้ขณะที่สืบค้นข้อมูลในแต่ละโหนด กระบวนการนี้นำวิธีการดำเนินการด้วยภาษาเอกซ์พาธมาใช้เพื่อระบุตำแหน่งคำที่ทำการสืบค้น เนื่องจากภาษาเอกซ์พาธเป็นสับเซตของภาษาเอกซ์คิวรีในการกำหนดชื่อแท็กจะเพิ่มการระบุองค์ประกอบที่แยกเป็นส่วนๆ เช่น หัวข้อ (Title) หัวเรื่อง (Subject) บทคัดย่อ (Abstract) หรือรายละเอียด (Description) จากเอกสารโครงสร้างเอกซ์เอ็มแอล

<u>กลไกการนำเสนอสารสนเทศ</u> (Presentation Mechanism)

การนำเสนอ (Presentation) เป็นกระบวนการที่นำผลลัพธ์จากการประมวลผลการค้นคืนเอกสารที่เป็นข้อมูลเอกซ์เอ็มแอลและทำการแปลงข้อมูลที่ได้จากเอกสารเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอล

การติดต่อผู้ใช้ (User Interface) ทำหน้าที่ในการรับ-ส่งข้อมูลจากผู้ใช้เพื่อระบุคำสืบค้นและสร้างรูปแบบการนำเสนอ ซึ่งสามารถระบุพิกัดตำแหน่งการแสดงผลไว้โดยแยกการแสดงผลเป็นส่วนๆ เช่น หัวข้อ หัวเรื่อง บทคัดย่อ หรือรายละเอียดของบริบทที่เป็นผลลัพธ์ ในการกำหนดรูปแบบการแสดงผลจะใช้เป็นรูปแบบมาตรฐานเอกซ์เอสแอลที เพื่อนำไปใช้ในการแปลงข้อมูลเอกสารเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอล

ดังนั้นงานวิจัยนี้ต้องการพัฒนากรอบงานสารสนเทศควบรวมจากเอกสารที่มีโครงสร้างและแสดงผลให้กับผู้ใช้ได้อย่างเหมาะสม โดยมีขั้นตอนที่ดำเนินการต่อจากเสิร์ชเอนจินในระบบจัดการเอกสารดังในภาพที่ 3.1

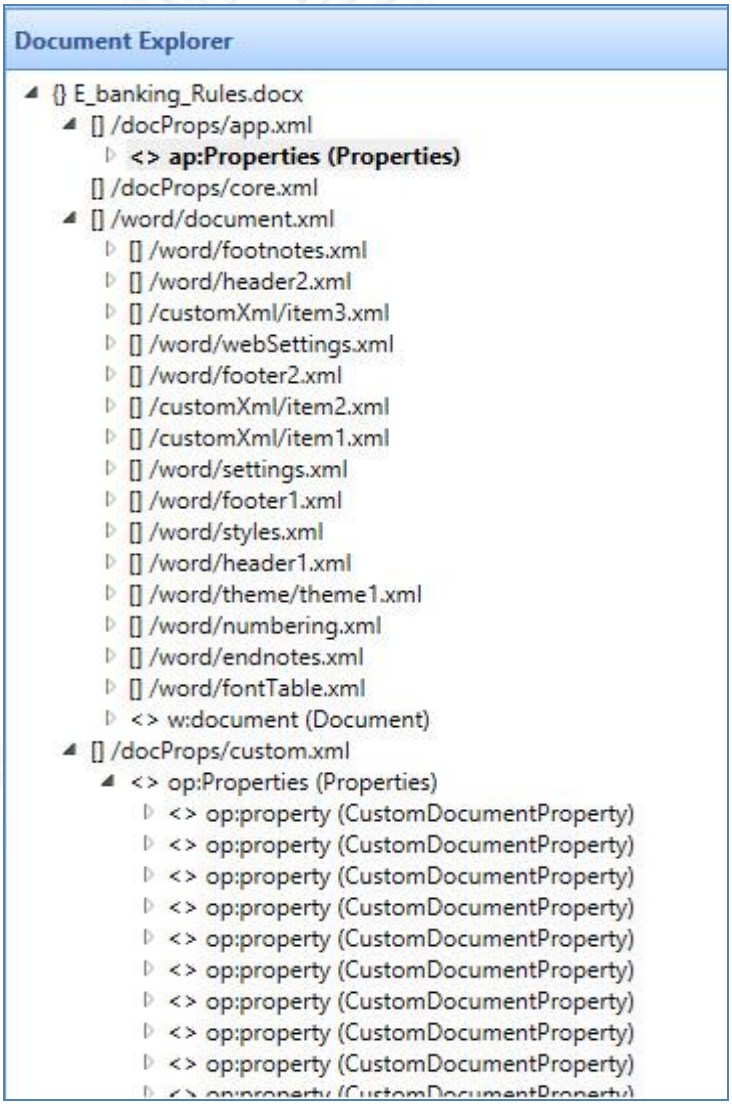


ภาพที่ 3.1 ขั้นตอนการทำงานของกรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้าง

ขั้นตอนที่ 1. ประกอบด้วย 2 กระบวนการย่อยดังนี้

1.การเตรียมเอกสาร (Documents Preparation)

เป็นขั้นตอนการตรวจสอบประเภทโครงสร้างของเอกสาร เมื่อพบว่าเอกสารไม่ได้อยู่ในรูปแบบโครงสร้างเอกซ์เอ็มแอลจะทำการแปลงไฟล์เอกสารจากนามสกุล .docx เป็นเอกสารโครงสร้างข้อมูลเอกซ์เอ็มแอลไฟล์นามสกุล .xml ในภาพที่ 3.2 เป็นตัวอย่างการแจงองค์ประกอบโครงสร้างเอกสารไมโครซอฟต์เวิร์ด(.docx) เวอร์ชัน 2007 ด้วยเครื่องมือไลบรารีทำถูกนำมาพัฒนาระบบในงานวิจัยนี้คือ OpenXML SDK 2.5 Productivities for Microsoft Office และในภาพที่ 3.3 แสดงข้อมูลรายละเอียดของคุณสมบัติเอกสาร .docx ในรูปแบบเอกซ์เอ็มแอล



ภาพที่ 3.2 ตัวอย่างการตรวจสอบองค์ประกอบโครงสร้างเอกสาร .docx ด้วยเครื่องมือ OpenXML 2.5 SDK Productivities for Microsoft Office

```
                        Analyzing A Banks Financial Statements
<ap:Properties
xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes"
xmlns:ap="http://schemas.openxmlformats.org/officeDocument/2006/extended-properties
">
  <ap:Template>Normal</ap:Template>
  <ap:TotalTime>1</ap:TotalTime>
  <ap:Pages>6</ap:Pages>
  <ap:Words>1849</ap:Words>
  <ap:Characters>10542</ap:Characters>
  <ap:Application>Microsoft Office Word</ap:Application>
  <ap:DocSecurity>4</ap:DocSecurity>
  <ap:Lines>87</ap:Lines>
  <ap:Paragraphs>24</ap:Paragraphs>
  <ap:ScaleCrop>false</ap:ScaleCrop>
  <ap:HeadingPairs>
    <vt:vector baseType="variant" size="2">
      <vt:variant>
        <vt:lpstr>Title</vt:lpstr>
      </vt:variant>
      <vt:variant>
        <vt:i4>1</vt:i4>
      </vt:variant>
    </vt:vector>
  </ap:HeadingPairs>
  <ap:TitlesOfParts>
    <vt:vector baseType="lpstr" size="1">
      <vt:lpstr />
    </vt:vector>
  </ap:TitlesOfParts>
  <ap:Company>Grizli777</ap:Company>
  <ap:LinksUpToDate>false</ap:LinksUpToDate>
  <ap:CharactersWithSpaces>12367</ap:CharactersWithSpaces>
  <ap:SharedDoc>false</ap:SharedDoc>
  <ap:HyperlinksChanged>false</ap:HyperlinksChanged>
  <ap:AppVersion>12.0000</ap:AppVersion>
</ap:Properties>
```

ภาพที่ 3.3 ตัวอย่างคุณสมบัติโครงสร้างเอกสาร .docx ที่อยู่ในรูปแบบเอกซ์เอ็มแอล

2.การวิเคราะห์คำสำคัญ (Keyword Analytics)

การวิเคราะห์ศัพท์ควบคุมเป็นปัจจัยสำคัญในการค้นคืนเอกสารและบริบทที่เกี่ยวข้องกับคำสืบค้น ผลลัพธ์ที่ได้จะถูกนำไปประมวลผลความคล้ายกันของเอกสารและบริบท ทฤษฎีในการจัดการเรื่องคำศัพท์และดัชนีจึงมีความสำคัญและต้องใช้เวลาและปริมาณอย่างมากในการรวบรวมข้อมูลเพื่อให้ได้ประสิทธิภาพ งานวิจัยนี้จึงพิจารณาใช้บริการแหล่งข้อมูลออนไลน์ที่ทำการรวบรวมประมวลผลศัพท์ควบคุมและคำเสมือนหรือคำใกล้เคียง ตัวอย่างโค้ดโปรแกรมพีเอชพีที่เรียกใช้แหล่งข้อมูลออนไลน์ **url : “**https://nkp.iaea.org/INISMLThesaurus/en/ind.html**”** ที่ให้ผลลัพธ์เป็นแบบเอชทีเอ็มแอล หรือ .pdf ในตัวอย่างโค้ดเรียกว่า “Screen Scraper” [17] เป็นการนำเสนอแนวคิดของเว็บเซอร์วิสอีกรูปแบบหนึ่ง ผลลัพธ์ที่ได้จะเป็นข้อมูลรูปแบบ JSONP: JavaScript Notation with Padding

```
<?php
require 'scraperwiki/simple_html_dom.php';
$html = new simple_html_dom();
$html =
file_get_html("https://nkp.iaea.org/INISMLThesaurus/en/ind.html");

foreach($html->find('body h2') as $data1){
  $targetUrl = $data1->find('a',0)->href;
  $targetUrl ="https://nkp.iaea.org/INISMLThesaurus/en/". $targetUrl;
  $htmlTree = new simple_html_dom();
  $htmlTree = file_get_html($targetUrl);
  foreach($htmlTree->find('body a') as $data2){
    $keyword = $data2->find('p',0)->plaintext;
    $record = array('keyword' => $keyword);
    scraperwiki::save(array('keyword'), $record);
  }
}
?>
```

ภาพที่ 3.4 ตัวอย่างโค้ดโปรแกรมพีเอชพีที่ร้องขอคำศัพท์ควบคุมด้วยรูปแบบเอชทีเอ็มแอล

ขั้นตอนที่ 2. เป็นขั้นตอนการเก็บรวบรวมข้อมูลจากเอกสารเอกซ์เอ็มแอลโดยใช้การสืบค้นจากคำสำคัญด้วยภาษาเอกซ์คิวรี เซตข้อมูลผลลัพธ์จะถูกรวบรวมเพื่อนำไปประมวลผลความคล้ายกันของบริบทในขั้นตอนถัดไป ตัวอย่างภาษาเอกซ์คิวรี ในภาพที่ 3.5 เป็นการสืบค้นข้อมูลจากเอกสารชื่อ InformationRetrieval.xml กำหนดให้ค้นหาคำสำคัญคือ “Clustering” และค้นหาคำที่มีความหมายใกล้เคียงกันคือ “Grouping” และ “Gathering”

```
for $b in doc("InformationRetrieval.xml") //Document
where some $p in $b //Paragraph
satisfies(contains($p,"Clustering")
        and contains($p,"Grouping")
        and contains($p,"Gathering"))
return $b/title
```

ภาพที่ 3.5 ตัวอย่างภาษาเอกซ์คิวรีเพื่อสืบค้นข้อมูลจากเอกสารโครงสร้างเอกซ์เอ็มแอล

ขั้นตอนที่ 3. การวิเคราะห์ความคล้ายของบริบทจะใช้หลักทฤษฎีของการจัดกลุ่มข้อความ (Text Clustering) ที่มีพื้นฐานโมเดลปริภูมิเวกเตอร์ ผลลัพธ์ที่ได้จะเป็นลำดับของเอกสารและบริบทที่มีเนื้อหาสาระสัมพันธ์และเกี่ยวข้องกับคำสำคัญที่ใช้ค้นหา วัดค่าความคล้ายกัน (Similarity Measure) จากทฤษฎีการจัดกลุ่มข้อความที่ใช้การเรียนรู้ที่ไม่มีการสอนล่วงหน้า (Unsupervise Learning) งานวิจัยนี้จะใช้อัลกอลิทึม K-means มาใช้ในการประมวลผลเพื่อหาจุดศูนย์กลางของกลุ่ม (Cluster Centroid) ของคำในแต่ละเอกสารดังสมการ (3.1) [7]

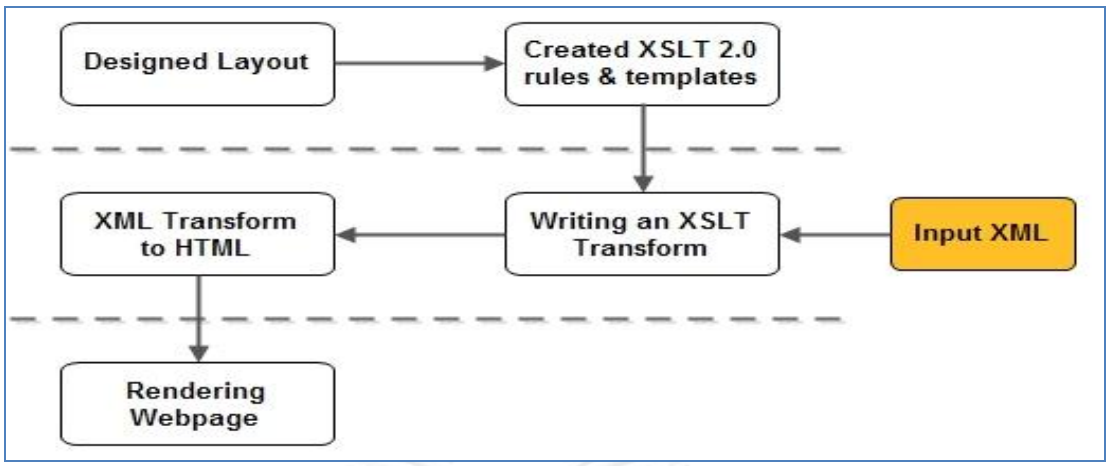
$$dist_{cos}\left(doc_i, doc_j\right) = 1 - \frac{\sum_{k=1}^{d} doc_i^{(k)} \times doc_j^{(k)}}{\|doc_i\| \times \|doc_j\|} \qquad (3.1)$$

และเทคนิคปริภูมิเวกเตอร์โมเดล [4] โดยใช้เทคนิควิธีวัด TF-IDF โดยการกำหนดค่าน้ำหนักคำสืบค้น เพื่อเพิ่มประสิทธิภาพในการวัดผลความคล้ายในระดับของมิติในเอกสารดังสมการ (3.2) [6]

$$Sim(\vec{D}, \vec{Q}) = \sum_{t_i \in Q,D} w_{t_i Q} * w_{t_i D} \qquad (3.2)$$

ขั้นตอนที่ 4. จากผลลัพธ์ลำดับของเอกสารและบริบทจะถูกนำมารวบรวมเพื่อแสดงผล ซึ่งใช้รูปแบบโครงสร้างข้อมูลเอกซ์เอ็มแอล

ขั้นตอนที่ 5. ในการแสดงผลลัพธ์ให้กับผู้ใช้งานที่เป็นเว็บเพจ โดยมีการออกแบบรูปแบบการแสดงผล เพื่อระบุตำแหน่งของข้อมูล ดังนั้นขั้นตอนนี้จะทำการแปลงข้อมูลเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอลและเอกซ์เอชทีเอ็มแอล โดยใช้ภาษาเอกซ์เอสเอลทีมาช่วยทำให้ข้อมูลแสดงผลตามรูปแบบที่กำหนดไว้ ในภาพที่ 3.6 เป็นการแสดงขั้นตอนการแปลงข้อมูลเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอลเพื่อแสดงผลข้อมูลเป็นเว็บเพจ ซึ่งได้มีการกำหนดรูปแบบการแสดงผลไว้ก่อนหน้าเป็นแผ่นแบบ (Template) โดยผู้ใช้งาน



ภาพที่ 3.6 ขั้นตอนการเปลี่ยนข้อมูลเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอลด้วยเอกซ์เอสแอลที

3.2 ข้อมูลระบบ

3.2.1 การเตรียมข้อมูลเอกสารและการนำเข้าแฟ้มเอกสาร

เอกสารที่ใช้ในงานวิจัยนี้จะเป็นเอกสารมีโครงสร้างเอกซ์เอ็มแอล ซึ่งถูกเปลี่ยนแปลงรูปแบบโครงสร้างจากเอกสารไมโครซอฟต์เวิร์ด เวอร์ชัน 2007 หรือล่าสุดเท่านั้นและเอกสารที่ใช้ทดสอบเป็นเอกสารที่เกี่ยวข้องภายในองค์กรเอกชนแห่งหนึ่ง ภายใต้ธุรกิจทางการเงินการธนาคาร การนำเข้าระบบจัดการเอกสารไมโครซอฟต์แชร์พอยต์ เซิร์ฟเวอร์ 2013 ด้วยเว็บเพจที่ถูกพัฒนาจากไซต์ต้นแบบ รายละเอียดไซต์ต้นแบบในเอกสาร ภาคผนวก ข.

3.2.2 ข้อมูลคำศัพท์ควบคุม

ศัพท์ควบคุมเป็นผลลัพธ์ที่ได้จากแหล่งข้อมูลออนไลน์ การสร้างส่วนเชื่อมต่อกับแหล่งข้อมูลออนไลน์ด้วยไลบรารีของกลุ่มผู้พัฒนา OCLC ในการทดลองงานวิจัยของศัพท์ควบคุมที่เรียกว่า Terminology Service [19] ที่ให้บริการในรูปแบบที่แตกต่างกันของแต่ละประเภทของแหล่งข้อมูลออนไลน์ การทำงานของบริการ TS มีคุณลักษณะดังนี้

1. สามารถค้นหารายละเอียดของศัพท์ควบคุมได้
2. สามารถค้นหาแนวคิดหรือหัวข้อ (Concept/Heading) ของศัพท์ควบคุมได้
3. สามารถค้นคืนแนวคิดหรือหัวข้อเดียวที่ถูกบันทึกข้อมูลไว้
4. สามารถค้นคืนแนวคิดหรือหัวข้อที่ได้ผลลัพธ์เป็นแบบหลากหลายความหมาย โดยส่งผลลัพธ์ในรูปแบบ HTML, MARC XML, ZThes และ SKOS
5. สามารถค้นหาด้วยไวยากรณ์ SRU CQL

ตัวอย่างการใช้ไวยากรณ์ SRU ในการค้นหาและค้นคืน

ตัวอย่างที่ 1. การค้นหาเทอมดัชนีที่อยู่ในส่วนของนวนิยายวิทยาศาสตร์

http://tspilot.oclc.org/gsafd/?query=oclcts.preferredTerm+%3D+%22science+fiction%22&version=1.1&operation=searchRetrieve

gsafd คือ รหัสของหัวเรื่อง (subject code) ที่มีหัวข้ออยู่ในหมวดนวนิยาย

ตัวอย่างที่ 2. การค้นหาเทอมอื่นๆที่อยู่ในหมวดนิยายที่เกี่ยวกับฆาตกรรมลึกลับ

http://tspilot.oclc.org/gsafd/?query=oclcts.alternativeTerms+%3D+%22whodunits%22+or+oclcts.alternativeTerms+%3D+%22thrillers%22&version=1.1&operation=searchRetrieve

ผลลัพธ์ที่ได้จากการค้นหาและค้นคืนสามารถระบุรูปแบบ HTML, MARC XML, ZThes และ SKOS ในงานวิจัยนี้เลือกรูปแบบ MARC XML เพราะสามารถใช้ภาษาเอกซ์พาธมาช่วยในการดึงข้อมูลผลลัพธ์ รายละเอียดการทำงานของขั้นตอนนี้จะอยู่ในบทที่ 4 ซึ่งเป็นกระบวนการพัฒนาระบบในส่วนของคำศัพท์ควบคุม

3.3 เครื่องมือที่เลือกใช้ในการพัฒนาระบบงานวิจัย

3.3.1 เครื่องมือในการจัดเก็บเอกสารและการสืบค้น

งานวิจัยนี้นำระบบไมโครซอฟต์แชร์พอยต์ เวอร์ชัน 2013 ซึ่งเป็นระบบการจัดการเอกสารที่ใช้ติดตั้งระบบปฏิบัติการและฐานข้อมูลเพื่อที่จะนำมาช่วยทำเป็นเว็บเพจ เป็นศูนย์กลางสำหรับการเชื่อมโยงไปสู่ผู้ใช้ ข่าวสารและองค์กร โดยช่วยเพิ่มความสามารถด้วยเครื่องมือในการจัดการเพจไซต์และทำให้ผู้ใช้นำข้อมูลไปแสดงในเพจไซต์เพื่อเผยแพร่ให้กับบุคคลากรอื่นๆในองค์กรได้ เนื่องจากระบบมีการนำไปใช้งานที่ยืดหยุ่นทำให้ผู้ใช้สามารถค้นหาข้อมูลที่มีความสัมพันธ์กันได้อย่างรวดเร็วผ่านฟังก์ชันการค้นหาที่มีประสิทธิภาพ ด้วยรูปแบบลักษณะที่เป็นเว็บทำให้เกิดการใช้งานที่คุ้นเคย ขณะเดียวกันองค์กรก็สามารถเจาะจงข่าวสาร โปรแกรม และการอัพเดทต่างๆ ไปยังผู้ใช้ที่ต้องการได้ตามหน้าที่รับผิดชอบการทำงาน การเป็นสมาชิกในทีมงาน ความสนใจ หรือตามเงื่อนไขอื่นที่กำหนด

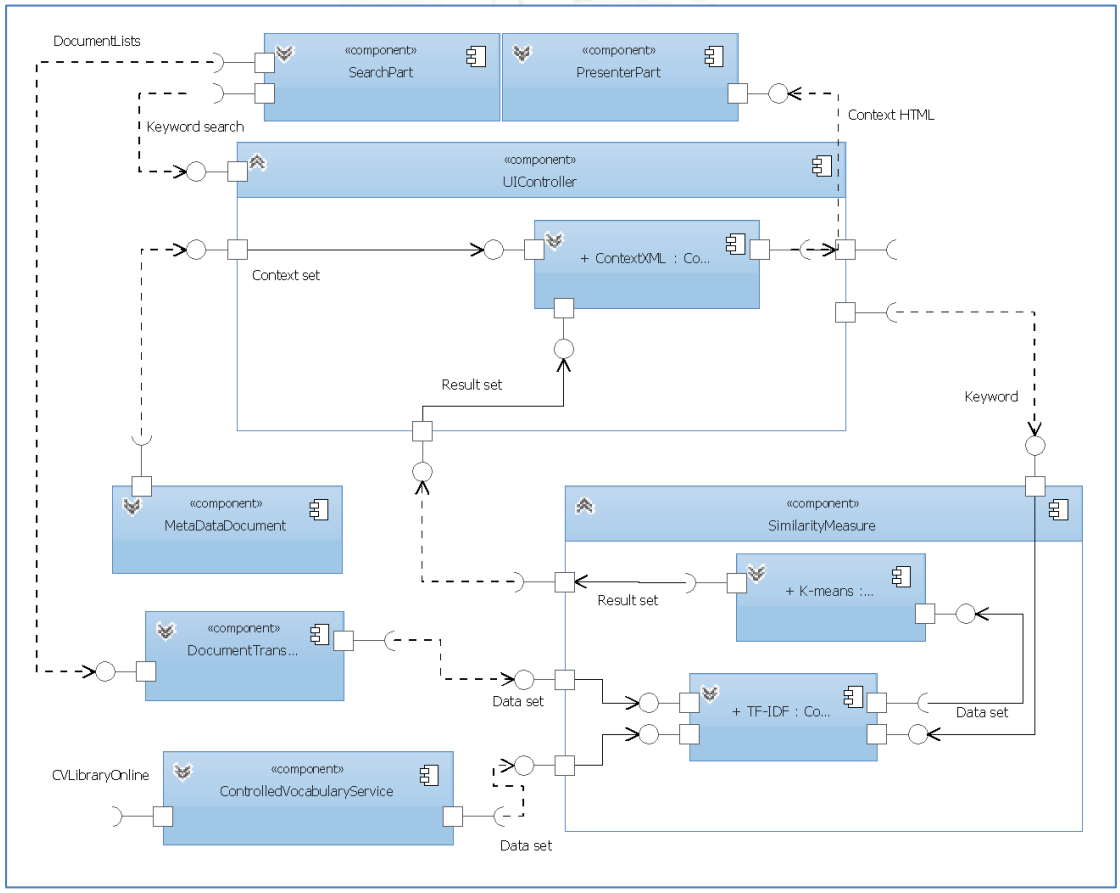
3.3.2 เครื่องมือที่ใช้ในการออกแบบและพัฒนาระบบ

ระบบไมโครซอฟต์แชร์พอยต์ที่ถูกนำมาใช้ในการดำเนินงานของงานวิจัยนี้สามารถเชื่อมต่อกับเครื่องมือที่ใช้ในการออกแบบและพัฒนาระบบของไมโครซอฟต์ดอตเน็ท (Microsoft .Net) ซึ่งภาษาที่ใช้ในการพัฒนาคือ C# ร่วมกับ Asp.Net ใช้พัฒนาเว็บเพจร่วมกับเพจไซต์ที่ใช้เป็นส่วนติดต่อประสานและแสดงผลให้กับผู้ใช้งานระบบ

บทที่ 4
การออกแบบและพัฒนาระบบ

4.1 สถาปัตยกรรมระบบ

วิจัยนี้ได้นำเสนอแนวทางการพัฒนากรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้าง จากบทที่ 3 แนวคิดในการดำเนินงานวิจัยที่ประกอบด้วย 5 กระบวนการทำงาน สามารถออกแบบสถาปัตยกรรมแอปพลิเคชันระบบโดยกำหนดเป็นแบบระดับชั้นของกระบวนการ เพื่อสนับสนุนให้มีการบริหารจัดการโครงสร้างของแอปพลิเคชันที่ชัดเจน สะดวก และจัดการง่ายในการพัฒนาระบบ



ภาพที่ 4.1 สถาปัตยกรรมแอปพลิเคชันระบบของกรอบงานสารสนเทศควบรวมสำหรับการค้นคืนเอกสารมีโครงสร้างในองค์กร

ดังนั้นการพัฒนาระบบจึงถูกแยกออกเป็นแบบ 2 ระดับชั้น (2 Tiers) ประกอบด้วย ระดับชั้นแรก คือ ชั้นของการนำเสนอสารสนเทศ (Presentation Tier or User Interface Layer) เรียกว่า กลไกการนำเสนอซึ่งมาจากแนวคิดของงานวิจัยนี้ ระดับชั้นที่สอง คือ ชั้นของแอปพลิเคชัน (Application Tier or Logic Layer) ที่เรียกว่า กลไกการค้นคืนเอกสาร ซึ่งในแต่ละระดับชั้นจะประกอบด้วย คอมโพเนนต์ย่อยต่างๆ ดังภาพที่ 4.1

4.1.1 ระดับชั้นการนำเสนอ (Presentation Tier)

ระดับชั้นการนำเสนอสารสนเทศเป็นชั้นที่ทำงานบนแพลตฟอร์มของไมโครซอฟต์แชร์พอยต์ สามารถแยกองค์ประกอบของการพัฒนาแอปพลิเคชันออกเป็น 3 คอมโพเนนต์ คือ 1. เสิร์ชพาร์ท (ส่วนที่ใช้สืบค้น) จะทำหน้าที่รับค่าคำสำคัญที่จะใช้ในการสืบค้นเอกสารในระบบไมโครซอฟต์แชร์พอยต์ ในการพัฒนาส่วนนี้จะเรียกใช้ไลบรารี Microsoft.SharePoint โดยผ่านทางแอปพลิเคชันโปรแกรมอินเตอร์เฟส (API) ร่วมกับไลบรารี Microsoft.Office.Server.Search.Query ซึ่งผลลัพธ์ที่ได้จะเป็นรายการเอกสารเพื่อส่งให้กับอินเตอร์เฟสของคอมโพเนนต์ DocumentTranform 2. พรีเซนเตอร์พาร์ท (ส่วนของการนำเสนอ) เป็นคอมโพเนนต์ส่วนที่ถูกพัฒนาเพื่อรับข้อมูลเอชทีเอ็มแอลและนำเสนอสารสนเทศตามรูปแบบที่กำหนดตำแหน่งไว้และ 3. ยูไอคอลโทลเลอร์ (ส่วนควบคุมส่วนติดต่อประสานผู้ใช้งาน) เป็นส่วนที่ทำหน้าที่เป็นส่วนรับส่งข้อมูลคำสำคัญไปยังระดับชั้นแอปพลิเคชันรับ-ส่งข้อมูลที่เป็นผลลัพธ์จากคอมโพเนต์การวัดค่าความคล้าย และทำหน้าที่แปลงข้อมูลเอกซ์เอ็มแอลเป็นเอชทีเอ็มแอลโดยคอมโพเนต์ย่อยชื่อ ContextXML

4.1.2 ระดับชั้นแอปพลิเคชัน (Application Tier)

ระดับชั้นแอปพลิเคชันเป็นการทำงานในส่วนด้านหลัง (Back-end) ของระบบซึ่งประกอบด้วยคอมโพเนนต์ 4 ส่วนหลักดังนี้ 1. การสร้างเอกสารเมทาดาตา (MetadataDocument) เป็นคอมโพเนนต์ที่ทำหน้าที่กำหนดแท็กในการทำดัชนีในเอกสารเพื่อใช้ในการรวบรวมบริบทตามที่ต้องการหลังจากการประมวลผลความคล้าย 2. การแปลงข้อมูลเอกสาร (DocumentTransform) เป็นคอมโพเนนต์ที่ใช้แปลงเอกสารไมโครซอฟต์เวิร์ดเป็นเอกสารเอกซ์เอ็มแอล เพื่อนำไปใช้ในการประมวลผลความคล้าย 3. บริการคำศัพท์ควบคุม (ControlledVocabularyService) เป็นส่วนทำหน้าที่เชื่อมต่อกับหน่วยบริการคำศัพท์ควบคุมออนไลน์ซึ่งเป็นส่วนภายนอกระบบและ 4. SimilarityMeasure ประกอบด้วย 2 คอมโพเนนต์ย่อยคือ K-means และ TFIDF ทำหน้าที่ในการประมวลผลความคล้ายกันของบริบทที่ได้จากเอกสาร

4.2 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบประกอบด้วยรายการฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

4.2.1 สภาพแวดล้อม

1. คอมพิวเตอร์โซนี ไวโอ รุ่น VPCEB15FH ปี 2010
2. หน่วยประมวลผลอินเทล คอร์ไอทรี 2.13 กิกะเฮิร์ต (i3-330M 2.13 GHz)
3. หน่วยความจำขนาด 8 กิกะไบต์
4. ฮาร์ดดิสก์ไดรฟ์ขนาดความจุ 320 กิกะไบต์
5. ระบบปฏิบัติการไมโครซอฟต์วินโดว์ 8 แบบ 64 บิท
6. โปรแกรมออราเคิล วีเอ็ม เวอร์ชวลบอกซ์ เวอร์ชัน 4.3.8
   6.1 เวอร์ชวลหน่วยความจำขนาด 4 กิกะไบต์
   6.2 เวอร์ชวลฮาร์ดดิสก์ไดรฟ์ขนาดความจุ 100 กิกะไบต์
   6.3 ระบบปฏิบัติการไมโครซอฟต์วินโดว์เซิร์ฟเวอร์ 2012 แบบ 64 บิท

4.2.2 เครื่องมือที่ใช้ในการพัฒนาระบบ

1. โปรแกรมไมโครซอฟต์แชร์พอยต์เซิร์ฟเวอร์ 2013 (Microsoft SharePoint Server 2013)
2. โปรแกรมไมโครซอฟต์วิชวลสตูดิโอ ดอทเน็ท อัลติเมต 2012 (Microsoft Visual Studio .Net Ultimate 2012)
3. โปรแกรมไมโครซอฟต์ออฟฟิศ 2010 (Microsoft Office 2010)
4. โปรแกรมไลบรารีโอเพนเอกซ์เอ็มแอล เอสดีเค เวอร์ชัน 2.5 (OpenXML SDK 2.5)
5. โปรแกรมไมโครซอฟต์แชร์พอยต์ ดีไซเนอร์ 2013

4.3 การพัฒนาระบบ

4.3.1 การติดตั้งซอฟต์แวร์ในการพัฒนาระบบ (Development Software Installation)

หลังจากเตรียมสภาพแวดล้อมและเครื่องมือสำหรับการพัฒนาระบบเรียบร้อยแล้ว จะทำการติดตั้งเครื่องมือทั้งหมดในเครื่องคอมพิวเตอร์ที่ใช้พัฒนาระบบ โดยมีลำดับการติดตั้งเครื่องมือเป็นไปตามขั้นตอนต่อไปนี้

1. ติดตั้งโปรแกรมออราเคิล วีเอ็ม เวอร์ชวลบอกซ์ เวอร์ชัน 4.3.8
2. ติดตั้งระบบปฏิบัติการไมโครซอฟต์วินโดว์เซิร์ฟเวอร์ 2012 แบบ 64 บิท (เวอร์ชวลโอเอส)
3. ติดตั้งชุดระบบจัดการเอกสารและเว็บพอร์ทอลไมโครซอฟต์แชร์พอยต์ เซิร์ฟเวอร์ 2013
4. ติดชุดโปรแกรมไมโครซอฟต์ออฟฟิศ 2010

5. ติดตั้งชุดเครื่องมือพัฒนาโปรแกรมไมโครซอฟต์วิชวลสตูดิโอ ดอทเน็ท อัลติเมท 2012
6. ติดตั้งเครื่องมือพัฒนาโปรแกรมโครซอฟต์แชร์พอยต์ ดีไซเนอร์ 2013
7. ติดตั้งไลบรารีโปรแกรมโอเพนเอกซ์เอ็มแอล เอสดีเค เวอร์ชัน 2.5

4.3.2 การพัฒนากระบวนการด้านหลัง (Back-end Processes Development)

4.3.2.1 การสร้างกระบวนการแปลงโครงสร้างเอกสารเอกซ์เอ็มแอล

ในการสร้างเอกสารโครงสร้างเอกซ์เอ็มแอลจากเอกสารต้นฉบับไมโครซอฟต์เวิร์ด .docx โดยการสร้างคลาสชื่อ XMLConvertor ประกอบด้วย 2 เมทอด คือ 1. TranformToXML() และ 2. PreparedToXML() ดังแสดงในภาพที่ 4.2-4.4 การทำงานของเมทอด TranformToXML จะโหลดข้อมูลในไฟล์ต้นฉบับด้วยคลาส WordprocessingDocument ในไลบรารีโอเพนเอกซ์เอ็มแอลที่มีการประมวลผลข้อมูลโครงสร้างเอกซ์เอ็มแอล และทำการสืบค้นโหนดข้อมูลประเภทที่มีแอทริบิวต์แบบ Paragraph และ Default เพื่อทำการส่งเป็นพารามิเตอร์ของเมทอด PreparedToXML การทำงานของเมทอดนี้จะตรวจสอบโหนดอิลิเมนท์ 3 ประเภทคือ **W.document**, **W.p** และ **W.sdt** และทำการดึงข้อมูลข้อความในแต่ละโหนดเพื่อแปลงเป็นโหนดแท็ก 2 ประเภท คือ **<e:document>** ซึ่งเป็นโหนดราก และ **<e:p>** เป็นโหนดลูกที่เก็บข้อความในแต่ละย่อหน้า

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Xml.Linq;
using DocumentFormat.OpenXml.Packaging;
using OpenXmlPowerTools;

namespace CIF_Lib.XMLTransform
{
    class XMLConvertor
    {
        private XNamespace E_NameSpace = "http://www.cp.eng.chuala.ac.th/Appl/Coder";
        public string SourFilePath { get; set; }
        public string DestFilePath { get; set; }
        public XMLConvertor()
        {
        }

        public bool TransformToXml(string[] keyword)

        private bool isSeachInFile(WordprocessingDocument wordDoc, string[] strFinding)

        private object PreparedToXML(XNode node, string defaultParagraphStyleId)

    }
}
```

ภาพที่ 4.2 ตัวอย่างคลาส XMLConvertor เพื่อทำการแปลงโครงเอกสารไมโครซอฟต์เวิร์ดเป็นเอกซ์เอ็มแอล

```
    public bool TransformToXml(string[] keyword)
    {
        bool bRet = false;
        try
        {
            byte[] byteArray = File.ReadAllBytes(SourFilePath);
            using (MemoryStream memoryStream = new MemoryStream())
            {
                memoryStream.Write(byteArray, 0, byteArray.Length);
                using (WordprocessingDocument wordDoc =
                    WordprocessingDocument.Open(memoryStream, true))
                {
                    RevisionAccepter.AcceptRevisions(wordDoc);
                    SimplifyMarkupSettings settings = new SimplifyMarkupSettings
                    {
                        RemoveComments = true,
                        RemoveContentControls = false,
                        RemoveEndAndFootNotes = true,
                        RemoveFieldCodes = true,
                        RemoveLastRenderedPageBreak = true,
                        RemovePermissions = true,
                        RemoveProof = true,
                        RemoveRsidInfo = true,
                        RemoveSmartTags = true,
                        RemoveSoftHyphens = true,
                        ReplaceTabsWithSpaces = true,
                    };
                    MarkupSimplifier.SimplifyMarkup(wordDoc, settings);
                    string defaultParagraphStyleId = wordDoc.MainDocumentPart
                        .StyleDefinitionsPart.GetXDocument().Root.Elements(W.style)
                        .Where(elm => (string)elm.Attribute(W.type) == "paragraph" &&
                            (string)elm.Attribute(W._default) == "1")
                        .Select(stl => (string)stl.Attribute(W.styleId))
                        .FirstOrDefault();
                    XElement simplerXml = (XElement)PreparedToXML(
                        wordDoc.MainDocumentPart.GetXDocument().Root,
                        defaultParagraphStyleId);
                    //Console.WriteLine(simplerXml);

                    if (isSeachInFile(wordDoc, keyword))
                    {
                        simplerXml.Save(DestFilePath);
                        bRet = true;
                    }
                }
            }

        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
```

ภาพที่ 4.3 ตัวอย่างโค้ดโปรแกรมของเมทอด TransformToXML()

```
private object PreparedToXML(XNode node, string defaultParagraphStyleId)
{
    XElement element = node as XElement;
    if (element != null)
    {
        if (element.Name == W.document)
            return new XElement(E_NameSpace + "document",
                new XAttribute(XNamespace.Xmlns + "e", E_NameSpace),
                element.Element(W.body).Elements()
                    .Select(e => PreparedToXML(e, defaultParagraphStyleId)));
        if (element.Name == W.p)
        {
            string styleId = (string)element.Elements(W.pPr)
                .Elements(W.pStyle).Attributes(W.val).FirstOrDefault();
            if (styleId == null)
                styleId = defaultParagraphStyleId;
            return new XElement(E_NameSpace + "p",
                new XAttribute("style", styleId),
                element.LogicalChildrenContent(W.r).Elements(W.t).Select(t => (string)t)
                    .StringConcatenate());
        }
        if (element.Name == W.sdt)
            return new XElement(E_NameSpace + "contentControl",
                new XAttribute("tag", (string)element.Elements(W.sdtPr)
                    .Elements(W.tag).Attributes(W.val).FirstOrDefault()),
                element.Elements(W.sdtContent).Elements()
                    .Select(e => PreparedToXML(e, defaultParagraphStyleId)));
        return null;
    }
    return node;
}
```

ภาพที่ 4.4 ตัวอย่างโค้ดโปรแกรมของเมทอด PreparedToXML()

จากภาพที่ 4.5 เป็นตัวอย่างข้อมูลเอกสารไมโครซอฟต์เวิร์ด .docx ก่อนจะถูกแปลงเป็นเอกสารโครงสร้างเอกซ์เอ็มแอลโดยใช้คลาส XMLConvertor และในภาพที่ 4.6 เป็นผลลัพธ์ข้อมูลเอกสารเอกซ์เอ็มแอลหลังจากถูกแปลงข้อมูล**สำเร็จ**

Analyzing a Bank's Financial Statements

Financial statements for banks present a different analytical problem than statements for manufacturing and service companies.

As a result, analysis of a bank's financial statements requires a distinct approach that recognizes a bank's unique risks.

Banks take deposits from savers and pay interest on some of these accounts.

They pass these funds on to borrowers and receive interest on the loans.

Their profits are derived from the spread between the rate they pay for funds and the rate they receive from borrowers.

This ability to pool deposits from many sources that can be lent to many different borrowers creates the flow of funds inherent in the banking system.

By managing this flow of funds, banks generate profits, acting as the intermediary of interest paid and interest received, and taking on the risks of offering credit.

Leverage and Risk
Banking is a highly leveraged business requiring regulators to dictate minimal capital levels to help ensure the solvency of each bank and the banking system.

As one of the most highly regulated banking industries in the world, investors have some level of assurance in the soundness of the banking system.

As a result, investors can focus most of their efforts on how a bank will perform in different economic environments.

Review the sample income statement and balance sheet for a large bank.

The first thing to notice is that the line items in the statements are not the same as your typical manufacturing or service firm.

Instead, there are items that represent interest earned or expensed on the (Income Statement, as well as deposits and loans (Balance sheet).

ภาพที่ 4.5 ตัวอย่างเอกสารต้นฉบับไมโครซอฟต์เวิร์ด .docx

```
                    Analyzing A Banks Financial Statements
<?xml version="1.0" encoding="utf-8"?>
<e:document xmlns:e="http://www.cp.eng.chuala.ac.th/Appl/Coder">
  <e:p style="Normal">Analyzing a Bank's Financial Statements</e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">Financial statements for banks present a different analytical
problem than statements for manufacturing and service companies. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">As a result, analysis of a bank's financial statements
requires a distinct approach that recognizes a bank's unique risks. Banks take
deposits from savers and pay interest on some of these accounts. They pass these
funds on to borrowers and receive interest on the loans. Their profits are derived
from the spread between the rate they pay for funds and the rate they receive from
borrowers. This ability to pool deposits from many sources that can be lent to many
different borrowers creates the flow of funds inherent in the banking system. By
managing this flow of funds, banks generate profits, acting as the intermediary of
interest paid and interest received, and taking on the risks of offering
credit.Leverage and RiskBanking is a highly leveraged business requiring regulators
to dictate minimal capital levels to help ensure the solvency of each bank and the
banking system. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">As one of the most highly regulated banking industries in the
world, investors have some level of assurance in the soundness of the banking
system. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">As a result, investors can focus most of their efforts on how
a bank will perform in different economic environments.Review the sample income
statement and balance sheet for a large bank. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">The first thing to notice is that the line items in the
statements are not the same as your typical manufacturing or service firm. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">Instead, there are items that represent interest earned or
expensed on the (Income Statement, as well as deposits and loans (Balance sheet).
As financial intermediaries, banks assume two primary types of risk as they manage
the flow of money through their business. Interest rate risk is the management of
the spread between interest paid on deposits and received on loans over time.
Credit risk is the likelihood that a borrower will default on a loan or lease,
causing the bank to lose any potential interest earned as well as the principal
that was loaned to the borrower. As investors, these are the primary elements that
need to be understood when analyzing a bank's financial statement.Interest Rate
RiskThe primary business of a bank is managing the spread between deposits
(liabilities, loans and assets). Basically, when the interest that a bank earns
from loans is greater than the interest it must pay on deposits, it generates a
positive interest spread, yield or net interest income. The size of this spread is
a major determinant of the profit generated by a bank. This interest rate risk is
primarily determined by the shape of the yield curve.</e:p>
  <e:p style="Normal">An upward sloping yield curve is favorable to a bank as the
bulk of its deposits are short term and their loans are longer term. This mismatch
of maturities generates the net interest revenue banks enjoy. </e:p>
  <e:p style="Normal"></e:p>
  <e:p style="Normal">When the yield curve flattens, this mismatch causes net
interest revenue to diminish.As a result, SPREAD (net interest income) will vary,
                                    Page 1
```


ภาพที่ 4.6 ตัวอย่างเอาต์พุตเอกสารเอกซ์เอ็มแอลที่ถูกแปลงข้อมูล

4.3.2.2 การสร้างการเชื่อมต่อและรวบรวมศัพท์ควบคุมออนไลน์

การเรียกใช้บริการศัพท์ควบคุมออนไลน์ของแหล่งข้อมูล TS ที่พัฒนาโดยกลุ่ม OCLC การทำงานของระบบจะทำการเชื่อมต่อด้วยโปรโตคอล เอชทีทีพี (http) ซึ่งทาง TS กำหนดให้ใช้เป็นเว็บแอปพลิชันโปรแกรมอินเตอร์เฟส (Web API) ในการพัฒนาระบบของงานวิจัยนี้ สร้างคลาส CVManager ที่ประกอบด้วย 2 เมทอด คือ เมทอด requestToCVLib() เพื่อเชื่อมต่อและส่งคำร้องขอค้นหาคำศัพท์ควบคุมจากแหล่งข้อมูล TS และเมทอด getControlVocabularies() รายละเอียดคลาส CVManager ดังแสดงในภาพที่ 4.7 กำหนดให้ใช้ไวยากรณ์ SRU ดังนี้

"http://tspilot.oclc.org/{0}/?query=oclcts.terms+any+%22{1}%22&version=1.1&operation=searchRetrieve&recordSchema=info%3Asrw%2Fschema%2F1%2Fmarcxml-v1.1&maximumRecords=10&startRecord=1&recordPacking=xml&sortKeys="

โดยกำหนดพารามิเตอร์ ดังนี้

{0} คือ รหัสหมวดหัวเรื่อง (Subject code) ที่จะใช้ค้นหา ในงานวิจัยนี้จะใช้รหัส "fast" เป็นหัวเรื่องของศัพท์ควบคุมที่ดำเนินจัดการโดย LCSH: Library of Congress Subject Heading [19]

{1} คือ คำที่ใช้สืบค้น เมื่อกำหนดพารามิเตอร์ครบทั้งหมดจะทำการร้องขอไปยังแหล่งข้อมูล TS ด้วยเว็บเอพีไอ รายละเอียดการทำงานของโปรแกรมในเมทอด requestToCVLib()

จากภาพที่ 4.7 โค้ดโปรแกรมของเมทอด getControlledVocabularies() เป็นการดึงข้อมูลที่เป็นผลลัพธ์การค้นหาจาก TS ในรูปแบบเอกซ์เอ็มแอลด้วยภาษาเอกซ์พาธ โดยใช้คลาส xPathUtility จะได้เป็นรายการของศัพท์ควบคุมเพื่อนำไปใช้ในการประมวลผลความคล้ายกันของเอกสารและบริบท ไวยากรณ์เอกซ์พาธของรายการข้อมูลที่เป็นผลลัพธ์ของ TS แสดงไว้ดังนี้

"//ns2:datafield[@tag='150']/ns2:subfield[@code='a']"

จากตารางที่ 2.1 สามารถอธิบายได้ดังนี้ ให้เลือกโหนดบริบทที่เป็นโหนดหลานหรือโหนดลูกของโหนด datafield โดยมี เอซิส ชื่อ ns2 ที่กำหนดให้มีเงื่อนไข @tag='150' หรือโหนด subfield ที่มีเงื่อไข @code='a'

```
public class CVManager
{    //string strUri = "http://tspilot.oclc.org/" + subject + "/?query=oclcts.terms+any+%22". $query
    //. '%22&version=1.1&operation=searchRetrieve&recordSchema=info%3Asrw%2Fschema%2F1%2Fmarcxml-v1.1
    //&maximumRecords=10&startRecord=1&recordPacking=xml&sortKeys=";
    CVServicePoint m_cvProxy = null;
    string m_strMsgForamt = "http://dev.sigwp.org/WikipediaThesaurusV3/Search.aspx?k={1}&t=0&l={0}";
    public CVManager(string strSubject, string strQuery)
    {
        m_strMsgForamt = String.Format(m_strMsgForamt, strSubject, strQuery);
        m_cvProxy = new CVServicePoint();
    }
    public string requestToCVLib()
    {
        return m_cvProxy.makeWebRequest(m_strMsgForamt);
    }
    public List<String> getControlledVocabularies()
    {
        string strHtml = m_cvProxy.makeWebRequest(m_strMsgForamt);


        List<String> lsRet = new List<String>();
        if (strHtml.Length > 0)
        {
            xPathUtility xPth = new xPathUtility();
            string strExpression = @"//ns2:datafield[@tag='150']/ns2:subfield[@code='a']";
            lsRet = xPth.GetElements(strHtml, strExpression);

        }
        return lsRet;

    }
}
```

ภาพที่ 4.7 ตัวอย่างคลาส CVManager

4.3.2.3 การสร้างเอกสารเมทาดาตา

การสร้างเมทาเดตาเอกสารในงานวิจัยนี้มีแนวคิดในการที่จะควบคุมบริบทของเอกสารด้วยการกำหนดดัชนีบริบทที่เป็นองค์ประกอบในเอกสารไมโครซอฟต์เวิร์ด ให้อยู่ในรูปแบบที่เป็นเอกซ์เอ็มแอลที่เรียกว่า **เมทาดาตา** เพื่อเก็บข้อมูลรายละเอียดประเภทองค์ประกอบของเอกสารนั้น หลักการของเมทาดาตาเกิดจากการกำหนดดัชนีและข้อมูลที่เกี่ยวกับองค์ประกอบเอกสาร 6 ประเภทดังนี้

1. ย่อหน้า (Paragraphs)
2. ตาราง (Tables)
3. ตารางแนวนอน (Rows)
4. เซลล์ของตาราง (Cells)
5. รายการหลัก (Lists)
6. รายการย่อย (List Items)

การทำงานของขั้นตอนนี้เริ่มจากคลาส IdentityManager รายละเอียดตัวอย่างคลาสในภาพที่ 4.8 เป็นไลบรารีของโอเพนเอกซ์เอ็มแอล เอสดีเค ทำการแก้ไขเอกสารด้วยการทำมาร์กอัปเพื่อเก็บข้อมูลในเอกสาร ซึ่งจะมีการกำหนดองค์ประกอบทั้ง 6 ประเภทที่จะถูกมาร์กอัปไว้ รายละเอียดการใช้งานคลาส IdentityManagerSettings ดังในภาพที่ 4.9

```
public class IdentityManager
{
    public static XElement ProcessIdentities(WordprocessingDocument wDoc, IdentityManagerSettings
settings)
    {
        ContentItemTypeMap = new Dictionary<string, ContentItemType>()
        {
            { settings.ContentItemTypeXmlValue.Table, ContentItemType.Table },
            { settings.ContentItemTypeXmlValue.Row, ContentItemType.Row },
            { settings.ContentItemTypeXmlValue.Cell, ContentItemType.Cell },
            { settings.ContentItemTypeXmlValue.Paragraph, ContentItemType.Paragraph },
            { settings.ContentItemTypeXmlValue.List, ContentItemType.List },
            { settings.ContentItemTypeXmlValue.ListItem, ContentItemType.ListItem },
        };

        ContentItemStatusMap = new Dictionary<string, ContentItemStatus>()
        {
            { settings.OperationTypeXmlValue.Existed, ContentItemStatus.Existed },
            { settings.OperationTypeXmlValue.Inserted, ContentItemStatus.Inserted },
        };

        /********************************* Checks before processing *********************************/
        var errors = CheckForDisallowedContent(wDoc);
        if (errors != null)
            return errors;

        /********************************* Clean up markup, accept certain revisions ******************
************/
        CleanupAndNormalizeMarkup(wDoc);

        /********************************* Do the identity processing proper ***************************
*****/
        ProcessIdentitiesInternal(wDoc, settings);

        /********************************* Finished *********************************/
        wDoc.MainDocumentPart.PutXDocument();
```

ภาพที่ 4.8 ตัวอย่างคลาส IdentityManager

```
IdentityManagerSettings settings = new IdentityManagerSettings
            {
                ContentItemTypeXmlValue = new ContentItemTypeXmlValue
                {
                    Table = "TABLE",
                    Row = "ROW",
                    Cell = "CELL",
                    Paragraph = "PARA",
                    List = "LIST",
                    ListItem = "ITEM",
                },
                CustomXmlXName = new CustomXmlXName
                {
                    IdentityManager = "identityManager",
                    FieldCode = "f",
                    Id = "id",
                    Item = "i",
                    ItemType = "t",
                    Operation = "o",
                    Unid = "u",
                },
                OperationTypeXmlValue = new OperationTypeXmlValue
                {
                    Existed = "EXISTS",
                    Inserted = "INSERTED",
                },
                RetrieveCustomXmlPart = wDoc => wDoc.MainDocumentPart.CustomXmlParts.First(),
                FieldCodeType = "COMMENTS"
            };
```

ภาพที่ 4.9 ตัวอย่างการนำไปใช้งานคลาส IdentityManagerSettings

ตัวอย่างเอกสารต้นฉบับไมโครซอฟต์เวิร์ดที่จะถูกสร้างเมทาเดตาในภาพที่ 4.10 ประกอบด้วย ย่อหน้าแรก, ตารางแรกที่ประกอบด้วย 5 แถวๆ 2 เซลล์, 3 ย่อหน้า, ตารางที่สองที่ประกอบด้วย 4 แถวๆ 2 เซลล์, 7 ย่อหน้า และสุดท้ายเป็นย่อหน้าที่ไม่มีข้อความ 2 ย่อหน้า และในภาพที่ 4.11 เป็นตัวอย่างเอกสารที่ถูกมาร์กอัปด้วยแท็กข้อความเอกซ์เอ็มแอล เริ่มจาก <f id="1"><i t="PARA" o="EXISTS" u="1" /></f> กำหนดเลขดัชนี เท่ากับ 1 เป็นประเภท (Type) ของย่อหน้า ประเภทโอเปอเรชันแบบ EXISTS นั่นคือเป็นข้อมูลเดิมที่ไม่ใช่ข้อมูลที่เพิ่มมาใหม่ และมีเลขดัชนียูนิต เท่ากับ 1 ดังนั้นสุดท้ายผลลัพธ์ที่ได้จากสร้างเมทาดาตาเพื่อใช้ในการเข้าถึงข้อมูลและการเลือกบริบทที่จะถูกรวบรวมในรูปแบบเอกซ์เอ็มแอลก่อนที่จะถูกแปลงเป็นเอชทีเอ็มแอลและแสดงข้อมูลกับผู้ใช้งาน

This is the first paragraph.

	222
For example, you can add a matching cover page, header, and sidebar.	333
Click Insert and then choose the elements you want from the different galleries. Video provides a powerful way to help you prove your point.	444 555
Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.	666

Themes and styles also help keep your document coordinated.

When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme.

When you apply styles, your headings change to match the new theme. Save time in Word with new buttons that show up where you need them.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	Maecenas porttitor congue massa.
Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.	Nunc viverra imperdiet enim.
Fusce est.	Vivamus a tellus.
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.	Proin pharetra nonummy pede.

Mauris et orci.

Aenean nec lorem.

In porttitor.

Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc.

Mauris eget neque at sem venenatis eleifend.

Ut nonummy.

ภาพที่ 4.10 ตัวอย่างเอกสารต้นฉบับไมโครซอฟต์เวิร์ดก่อนมาร์กอัป

<f id="1"><i t="PARA" o="EXISTS" u="1" /></f>This is the first paragraph.

<f id="39"><i t="ROW" o="EXISTS" u="16" /><i t="CELL" o="EXISTS" u="17" /><i t="TABLE" o="EXISTS" u="2" /><i t="PARA" o="INSERTED" /></f>	<f id="10"><i t="CELL" o="EXISTS" u="19" /><i t="PARA" o="EXISTS" u="20" /></f>222
<f id="11"><i t="ROW" o="EXISTS" u="21" /><i t="CELL" o="EXISTS" u="22" /><i t="PARA" o="EXISTS" u="23" /></f>For example, you can add a matching cover page, header, and sidebar.	<f id="12"><i t="CELL" o="EXISTS" u="24" /><i t="PARA" o="EXISTS" u="25" /></f>333
<f id="13"><i t="ROW" o="EXISTS" u="26" /><i t="CELL" o="EXISTS" u="27" /><i t="PARA" o="EXISTS" u="28" /></f>Click Insert and then choose the elements you want from the different galleries.	<f id="14"><i t="CELL" o="EXISTS" u="29" /><i t="PARA" o="EXISTS" u="30" /></f>444
<f id="15"><i t="ROW" o="EXISTS" u="31" /><i t="CELL" o="EXISTS" u="32" /><i t="PARA" o="EXISTS" u="33" /></f>Video provides a powerful way to help you prove your point.	<f id="16"><i t="CELL" o="EXISTS" u="34" /><i t="PARA" o="EXISTS" u="35" /></f>555
<f id="17"><i t="ROW" o="EXISTS" u="36" /><i t="CELL" o="EXISTS" u="37" /><i t="PARA" o="EXISTS" u="38" /></f>Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.	<f id="18"><i t="CELL" o="EXISTS" u="39" /><i t="PARA" o="EXISTS" u="40" /></f>666

<f id="19"><i t="PARA" o="EXISTS" u="41" /></f>Themes and styles also help keep your document coordinated.

<f id="20"><i t="PARA" o="EXISTS" u="42" /></f>When you click Design and choose a new Theme, the pictures, charts, and SmartArt graphics change to match your new theme.

<f id="21"><i t="PARA" o="EXISTS" u="43" /></f>When you apply styles, your headings change to match the new theme. Save time in Word with new buttons that show up where you need them.

<f id="22"><i t="TABLE" o="EXISTS" u="44" /><i t="ROW" o="EXISTS" u="45" /><i t="CELL" o="EXISTS" u="46" /><i t="PARA" o="EXISTS" u="47" /></f>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	<f id="23"><i t="CELL" o="EXISTS" u="48" /><i t="PARA" o="EXISTS" u="49" /></f>Maecenas porttitor congue massa.
<f id="24"><i t="ROW" o="EXISTS" u="50" /><i t="CELL" o="EXISTS" u="51" /><i t="PARA" o="EXISTS" u="52" /></f>Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.	<f id="25"><i t="CELL" o="EXISTS" u="53" /><i t="PARA" o="EXISTS" u="54" /></f>Nunc viverra imperdiet enim.
<f id="26"><i t="ROW" o="EXISTS" u="55" /><i t="CELL" o="EXISTS" u="56" /><i t="PARA" o="EXISTS" u="57" /></f>Fusce est.	<f id="27"><i t="CELL" o="EXISTS" u="58" /><i t="PARA" o="EXISTS" u="59" /></f>Vivamus a tellus.
<f id="28"><i t="ROW" o="EXISTS" u="60" /><i t="CELL" o="EXISTS" u="61" /><i t="PARA" o="EXISTS" u="62" /></f>Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.	<f id="29"><i t="CELL" o="EXISTS" u="63" /><i t="PARA" o="EXISTS" u="64" /></f>Proin pharetra nonummy pede.

<f id="30"><i t="PARA" o="EXISTS" u="65" /></f>Mauris et orci.

<f id="31"><i t="PARA" o="EXISTS" u="66" /></f>Aenean nec lorem.

ภาพที่ 4.11 ตัวอย่างเมทาดาตาเอกสารไมโครซอฟต์เวิร์ดหลังจากมาร์กอัป

4.3.2.4 การประมวลผลความคล้ายด้วยทฤษฎีโมเดลปริภูมิเวกเตอร์

จากทฤษฎีการวัดค่าความคล้ายกันที่ใช้โมเดลปริภูมิเวกเตอร์ในบทที่ 2 งานวิจัยนี้ได้พัฒนาโปรแกรมการประมวลผลความคล้ายด้วยอัลกอริทึม K-means และเทคนิควิธีวัด TF-IDF ซึ่งประกอบด้วยขั้นตอนการทำงานดังนี้

1. สร้างคลาส DataCollector เพื่อรวบรวมเซตข้อมูลคำที่จะใช้ประมวลจากเอกสารโครงสร้างเอกซ์เอ็มแอล (.xml) ตัวอย่างในภาพที่ 4.12 ที่ถูกแปลงจากเอกสารต้นฉบับไมโครซอฟต์เวิร์ด (.docx) เทคนิคที่นำมาใช้ในการดึงข้อมูลคือภาษาเอกซ์คิวรี ดังแสดงในภาพที่ 4.13 โดยใช้คำสืบค้น “Banking” ในเอกสาร iBanking.xml โดยกำหนดเนมสเปสของด็อกคูเมนท์เป็น

   “http://www.cp.eng.chula.ac.th/Appl/Coder”

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <e:document xmlns:e="http://www.cp.eng.chuala.ac.th/Appl/Coder">
3    <e:p style="Normal"></e:p>
4    <e:p style="Normal">Saudi Arabian Monetary Agency</e:p>
5    <e:p style="Normal"></e:p>
6    <e:p style="Normal">e-Banking Rules </e:p>
7    <e:p style="Normal"></e:p>
8    <e:p style="Normal">Banking Technology Department</e:p>
9    <e:p style="Normal">                              APRIL 2010</e:p>
10   <e:p style="Heading1">Introduction: </e:p>
11   <e:p style="Normal"></e:p>
12   <e:p style="Heading2">Electronic Banking Definition:</e:p>
13   <e:p style="Normal"></e:p>
```

ภาพที่ 4.12 ตัวอย่างเอกสาร iBanking.xml ที่ใช้ในการดึงข้อมูล “Banking”

```
NAMESPACE e = "http://www.cp.eng.chula.ac.th/Appl/Coder"
<Q8>
   {
       FOR $s IN document("iBanking.xml")//e:p
       WHERE some $p in $s //*:p in 'stringQuery'
           Satisfies(contains('Banking'))
       RETURN $s
   }
</Q8>
```

ภาพที่ 4.13 ตัวอย่างไวยากรณ์เอกซ์คิวรีที่ใช้ดึงข้อมูลจาก iBanking.xml ด้วยคำสืบค้น “Banking”

2. สร้างคลาส TFIDFManager เพื่อใช้ในการคำนวณจากสมการ (2.1) ด้วยเมทอด createVector() ประกอบด้วยพารามิเตอร์ DTVector ใช้เก็บรายการผลลัพธ์ของสมการ (2.1) , wordlist รายการเซตข้อมูลที่ถูกนำมาคำนวณ และ docs รายการของเอกสารที่นำมาคำนวณในสมการ รายละเอียดโค้ดโปแกรมแสดงในภาพที่ 4.14

```
private void createVector(Hashtable DTVector, List<String> wordlist, List<string> docs)
      {
          double[] queryvector;

          for (int j = 0; j < docs.Count; j++)
          {
              queryvector = new double[wordlist.Count];

              for (int i = 0; i < wordlist.Count; i++)
              {

                  double tfIDF = getTF(docs[j], wordlist[i]) * getIDF(wordlist[i], docs);
                  queryvector[i] = tfIDF;
              }

              if (j == 0) //is it a query?
              {
                  DTVector.Add("Query", queryvector);
              }
              else
              {
                  DTVector.Add(j.ToString(), queryvector);
              }
          }
      }
```

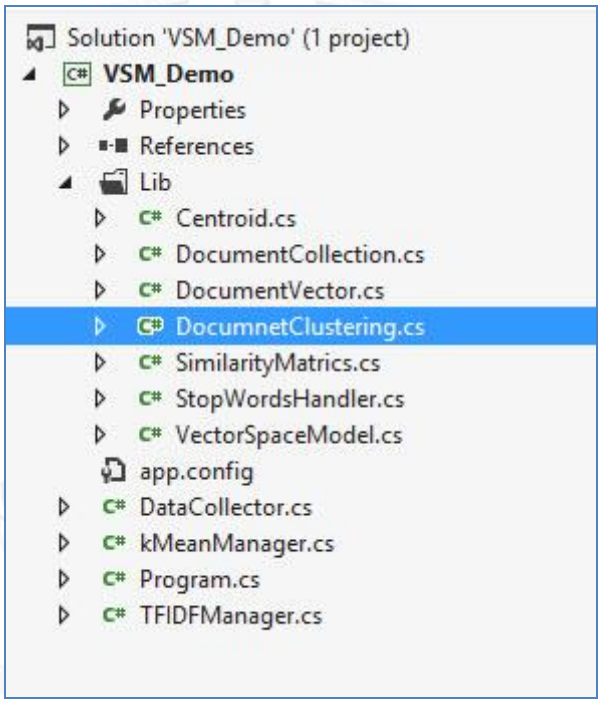
ภาพที่ 4.14 ตัวอย่างโค้ดโปรแกรมเมทอด createVector()

ในภาพที่ 4.15 เป็นตัวอย่างโค้ดโปรแกรมเมทอด classify() ที่ใช้ประมวลผลการจัดลำดับของเอกสาร เพื่อนำไปประมวลผลด้วยอัลกอริทึม K-means

```
private void classify(Hashtable DTVector, List<String> wordlist, Dictionary<int, Double> sortedList)
       {
           double temp = 0.0;

           IDictionaryEnumerator _enumerator = DTVector.GetEnumerator();

           double[] queryvector = new double[wordlist.Count];

           Array.Copy((double[])DTVector["Query"], queryvector, wordlist.Count);

           while (_enumerator.MoveNext())
           {
               if (_enumerator.Key.ToString() != "Query")
               {
                   temp = cosinetheta(queryvector, (double[])_enumerator.Value);

                   sortedList.Add(Convert.ToInt32(_enumerator.Key),temp);

               }
           }
       }
```

ภาพที่ 4.15 ตัวอย่างโค้ดโปรแกรมเมทอด classify()

3. การคำนวณ K-means เริ่มจากใช้คลาส Centroid เพื่อเก็บรายการข้อมูลก่อนทำกระบวนการจัดกลุ่ม และเตรียมข้อมูลในการประมวลผลด้วยคลาส DocumentsClustering รายละเอียดของคลาสทั้งที่ใช้ประมวลผลความคล้ายในภาพที่ 4.16 ซึ่งจะมีขั้นตอนการเริ่มต้นด้วยการจุดศูนย์กลางกลุ่มข้อมูล (Initializing Cluster Center) โดยเรียกการทำงานของเมทอด InitializeClusterCentroid() จากนั้นหากลุ่มจุดศูนย์กลางที่ใกล้ที่สุด (Finding Closet Cluster Center) เพื่อหาจุดศูนย์กลางใหม่ จากเมทอด FindClosestClusterCenter()



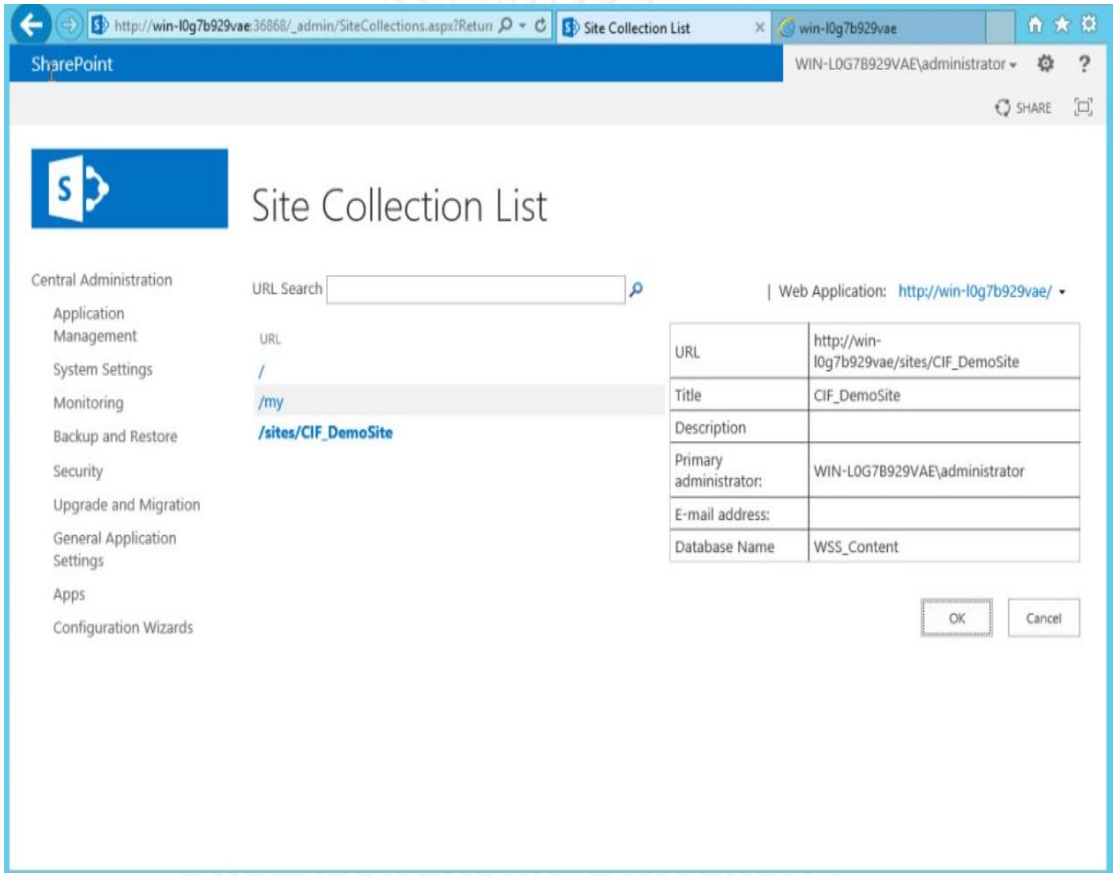
ภาพที่ 4.16 ตัวอย่างองค์ประกอบคลาสโปรแกรมที่ใช้ประมวลผลความคล้าย

4. เมื่อระบบทำการประมวลผลเสร็จ ผลลัพธ์ที่ได้จะเป็นรายการเอกสารที่ถูกเรียงลำดับความคล้ายกันของบริบทในเอกสาร ที่มีความเกี่ยวข้องและสัมพันธ์มากสุดกับคำสืบค้นและค่อยๆ ลดลำดับลงมา

4.3.3 การพัฒนาส่วนต่อประสานผู้ใช้ (User Interface Development)

4.3.3.1 การสร้างไซต์ส่วนตัวจากไซต์ต้นแบบ

ในภาคผนวก ข เป็นการสร้างไซต์ต้นแบบในแชร์พอยต์ให้ผู้ใช้สามารถสร้างไซต์ โดยออกแบบเองได้ ด้วยการกำหนดตำแหน่งการแสดงหน้าเพจแบบหลากหลาย สามารถกำหนดหน้าเพจที่จะแสดงให้เป็นเว็บเพจจากภายในหรือภายนอกระบบแชร์พอยต์ได้ ซึ่งในกระบวนการนี้จะเป็นขั้นตอนการออกแบบไซต์จากไซต์ต้นแบบ ตัวอย่างไซต์ต้นแบบงานวิจัยนี้เป็นทีมไซต์ ชื่อ **CIF_DemoSite** ดังแสดงในภาพที่ 4.17



ภาพที่ 4.17 ตัวอย่างหน้าจอเพจแสดงข้อมูลเพจไซต์ต้นแบบชื่อ CIF_DemoSite

เมื่อกำหนดเพจไซต์ที่ต้องการออกแบบจากไซต์ต้นแบบในส่วนของการพัฒนาระบบจะเป็นเว็บพาร์ท เพื่อให้เพจไซต์ที่ถูกออกแบบสามารถกำหนดตำแหน่งการแสดงผลได้ ซึ่งในงานวิจัยนี้จะพัฒนาเว็บพาร์ททั้งหมด 4 ส่วนคือ

1. เว็บพาร์ทสำหรับการค้นหา (Searching web part)
2. เว็บพาร์ทสำหรับส่วนที่เป็นหัวเรื่อง ชื่อเรื่องที่สำคัญ (Title/Subject web part)
3. เว็บพาร์ทสำหรับรายละเอียดบริบท (Contents web part)
4. เว็บพาร์ทสำหรับข้อมูลอื่นๆ ทั่วไป (Others detail web part)

การพัฒนาเว็บพาร์ทสำหรับการค้นหา (Searching web part)

ขั้นตอนในการพัฒนาเว็บพาร์ทค้นหาโดยใช้เครื่องมือไมโครซอฟต์วิชวลสตูดิโอ 2012 และไมโครซอฟต์ แชร์พอยต์ เซิร์ฟเวอร์ 2013 ที่เปิดให้สามารถพัฒนาระบบเชื่อมต่อกับระบบ โดยการเรียกใช้ API ที่ทางไมโครซอฟต์ได้มีการพัฒนาเพื่อสนับสนุนการพัฒนาระบบแชร์พอยต์ ดังนั้นในภาพที่ 4.18 งานวิจัยนี้จะเรียกใช้ไลบรารี Microsoft.SharePoint โดยการสร้างคลาส SearchManager ทำหน้าที่เรียกการค้นหาจากระบบการจัดการเอกสารแชร์พอยต์ ด้วยเมทอด Searching() โดยรับพารามิเตอร์ 2 ส่วน คือ คำสำคัญที่ใช้ค้นหา และไซต์พาธ แหล่งข้อมูลที่ต้องการให้ระบบทำการค้นหาเอกสาร ผลลัพธ์ที่ได้จากเมทอดนี้จะเป็นตารางรายการเอกสารที่ประกอบด้วยชื่อเจ้าของเอกสาร (Author) และขนาดของเอกสาร (Size)

```
class SearchManager
{
    public DataTable Searching(string strQuery, string strSiteDomain)
    {
        using (SPSite siteCollection = new SPSite(strSiteDomain))
        {
            KeywordQuery keywordQuery = new KeywordQuery(siteCollection);
            keywordQuery.QueryText = strQuery;
            keywordQuery.SortList.Add("Author", SortDirection.Ascending);
            keywordQuery.SortList.Add("Size", SortDirection.Descending);

            SearchExecutor searchExecutor = new SearchExecutor();
            ResultTableCollection resultTableCollection = searchExecutor.ExecuteQuery(keywordQuery);
            var resultTables = resultTableCollection.Filter("TableType", KnownTableTypes.RelevantResults);

            var resultTable = resultTables.FirstOrDefault();

            return resultTable.Table;
        }
    }
}
```

ภาพที่ 4.18 ตัวอย่างคลาส SearchManager ที่ใช้ไลบรารีของ Microsoft.SharePoint เพื่อค้นหาเอกสารในระบบแชร์พอยต์

4.3.3.2 การสร้างเว็บเซอร์วิส (CIF Webservice)

ในการเชื่อมต่อข้อมูลระหว่างระดับชั้นแอปพลิเคชันที่เป็นคอมโพเนนต์ UIController กับระดับชั้นนำเสนอสารสนเทศที่เป็นคอมโพเนนต์เสิร์ชเว็บพาร์ท (Seach web part) โดยใช้เว็บเซอร์วิสเนื่องจากมีการพัฒนาระบบที่สามารถกำหนดรูปแบบมาตรฐานในการรับ ส่งข้อมูลได้ง่าย สะดวกและในการนำไปใช้สามารถกำหนดให้การทำงานของระดับชั้นแอปพลิเคชันอยู่ที่ต่างระบบทางกายภาพหรือสภาวะแวดล้อม ที่สามารถเชื่อมต่อข้อมูลทางโปรโตคอลเอชทีทีพี (http Protocol) ในภาพที่ 4.19 ตัวอย่างโค้ดโปรแกรมคลาส CIFWebservice ที่แยกเมทอดการทำงานในแต่ละฟังก์ชันประกอบด้วย 1. เมทอด GetDataSimilarity 2. เมทอด GenDocumentsData และ 3. GetCVTheasuarus

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using CIF_Lib.CIF_UIController;

namespace CIFWebService
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.
    // NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc or Service1.svc.cs at the Solution Explorer and start debugging.
    public class CIFService : ICIFService
    {
        public List<String> GetDataSimilarity(string strKeyword, string[] astrTheasurus, string strSourceFolder)
        {
            UIControlManager uiCm = new UIControlManager();
            uiCm.DestFilePath = @"c:\CIF-Demo\destfile";
            uiCm.SourFilePath = @"C:\CIF-Demo\Docx";
            uiCm.Keyword = strKeyword;
            uiCm.Theasurus = astrTheasurus;
            uiCm.FolderName = strSourceFolder;

            return uiCm.TextClustering();

            //return string.Format("You entered: {0}", strKeyword);
        }

        public string GenDocumentsData(string strKeyword, string[] astrFolder, string[] astrTheasurus)
        {
            UIControlManager uiCm = new UIControlManager();
            uiCm.DestFilePath = @"c:\CIF-Demo\destfile";
            uiCm.SourFilePath = @"C:\CIF-Demo\Docx";
            uiCm.Theasurus = astrTheasurus;
            uiCm.Keyword = strKeyword;
            string strRet = "";
            if (uiCm.PrepareDataProcess(astrFolder))
                strRet = uiCm.FolderName;
            return strRet;

            //return string.Format("You entered: {0}", strKeyword);
        }

        public List<String> getCVTheasurus(string strKeyword)
        {
            UIControlManager uiCm = new UIControlManager();
            uiCm.Subject = "English";
            uiCm.Keyword = strKeyword;

            return uiCm.PrepareControlVocabulary();

            //return string.Format("You entered: {0}", strKeyword);
        }

        public CompositeType GetDataUsingDataContract(CompositeType composite)
        {
            if (composite == null)
            {
                throw new ArgumentNullException("composite");
            }
            if (composite.BoolValue)
            {
                composite.StringValue += "Suffix";
            }
            return composite;
        }
    }
}
```

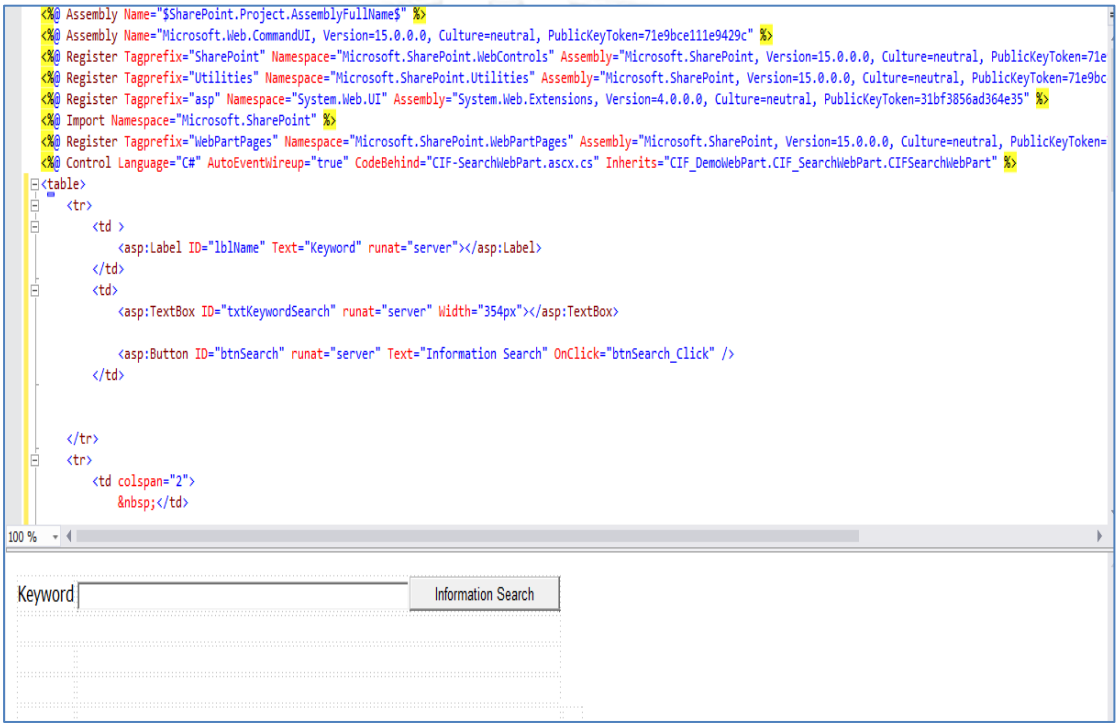
ภาพที่ 4.19 ตัวอย่างคลาส CIFWebservice ที่ใช้เชื่อมต่อข้อมูลระหว่างระดับชั้นแอปพลิเคชันกับระดับชั้นนำเสนอสารสนเทศ

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Linq;
using System.Text;
using System.Data;
using System.ServiceModel;
using System.IO;
//using Microsoft.SharePoint;
//using Microsoft.Office.Server.Search.Query;
//using CIF_Lib.CIF_UIController;
namespace CIF_DemoWebPart.CIF_SearchWebPart
{
    [ToolboxItemAttribute(false)]
    public partial class CIFSearchWebPart : WebPart
    {
        public string m_txtSearch;
        // Uncomment the following SecurityPermission attribute only when doing Performance Profiling using
        // the Instrumentation method, and then remove the SecurityPermission attribute when the code is
    ready
        // for production. Because the SecurityPermission attribute bypasses the security check for callers
    of
        // your constructor, it's not recommended for production purposes.
        // [System.Security.Permissions.SecurityPermission(System.Security.Permissions.SecurityAction.Assert,
     UnmanagedCode = true)]
        public CIFSearchWebPart()
        {
        }

        protected override void OnInit(EventArgs e)
        {
            base.OnInit(e);
            InitializeControl();
        }
        protected void btnSearch_Click(object sender, EventArgs e)
        {
            try
            {
                //Web service binding call CIFWebservice
                BasicHttpBinding basicHttpbinding = new BasicHttpBinding(BasicHttpSecurityMode.None);
                basicHttpbinding.Name = "BasicHttpBinding_ICIFService";
                basicHttpbinding.Security.Transport.ClientCredentialType = HttpClientCredentialType.None;
                basicHttpbinding.Security.Message.ClientCredentialType = BasicHttpMessageCredentialType.
    UserName;

                EndpointAddress endpointAddress = new EndpointAddress("http://win-l0g7b929vae:8088/CIFService
    .svc");



                CIFWebServiceReference.CIFServiceClient obj = new CIFWebServiceReference.CIFServiceClient
    (basicHttpbinding, endpointAddress);

                //Create Asynchinouse process
                AsyncCallback aCallTheasurus = new AsyncCallback(getTheasurusAsyncCallback);
                IAsyncResult arTheasurus = obj.BegingetCVTheasurus(txtKeywordSearch.Text, aCallTheasurus,
    obj);

                while (!arTheasurus.IsCompleted)
                { }

                obj.Close();

            }
            catch
            {

            }

        }
```

ภาพที่ 4.20 ตัวอย่างการพัฒนาส่วนต่อผสานผู้ใช้ที่เป็นเว็บพาร์ทคอมโพเนนต์

4.3.3.3 การสร้างเว็บพาร์ทคอมโพเนนต์ (Web part component)

เสิร์ซเว็บพาร์ทเป็นส่วนต่อผสานผู้ใช้ที่พัฒนาเป็นคอมโพเนนต์สามารถแทรกเพิ่มเข้าไปในเว็บเพจของผู้ใช้งานได้เมื่อผู้ใช้งานออกแบบหน้าเว็บเพจ โดยกำหนดตำแหน่งการแสดงผลหรือคุณลักษณะของเพจส่วนตัว คอมโพเนนต์เว็บพาร์ทเป็นส่วนหนึ่งในการพัฒนาของระบบไมโครซอฟต์แชร์พอยต์ ที่ถูกเพิ่มเข้าในเครื่องมือการพัฒนาระบบของไมโครซอฟต์วิชวลสตูดิโอ ในภาพที่ 4.20 ตัวอย่างการพัฒนาส่วนต่อผสานผู้ใช้เว็บพาร์ทที่ประกอบด้วยช่องพิมพ์ข้อความ (Textbox) สำหรับรับข้อมูลคำสำคัญที่จะใช้ในการค้น และปุ่ม Information Search ที่ใช้ในการเริ่มกระบวนการค้นหา ภาพที่ 4.21 ตัวอย่างโค้ดโปรแกรมของปุ่ม Information Search ที่เรียกใช้งานเว็บเซอร์วิส



ภาพที่ 4.21 ตัวอย่างโค้ดโปรแกรมของการค้นหาของเสิร์ซเว็บพาร์ทคอมโพเนนต์ที่เรียกใช้งานเว็บเซอร์วิสที่เชื่อมต่อเพื่อส่งข้อมูลกับระดับชั้นแอปพลิเคชัน

บทที่ 5
การประเมินและวัดผล

5.1 แนวทางการประเมินผลงานวิจัย

องค์ประกอบของแนวทางการประเมินผลการค้นคืนสารสนเทศ [13] ดังนี้

- **รูปแบบทางกายภาพของอินพุท** (Physical Input Form) คือ รูปแบบของเอกสารและความหมายของเอกสาร ซึ่งรูปแบบของเอกสารอาจจะอยู่ในรูปแบบต่อไปนี้คือ ชื่อ เรื่อง (Title) , บทย่อ (Abstract) , บทสรุป (Summary) , หรือข้อความเต็ม (Full Text) ซึ่งจะมีผลต่อการสร้างดัชนีและการค้นหาเอกสารและยังมีผลต่อค่าใช้จ่ายของระบบด้วย

- **โครงสร้างของแฟ้มข้อมูลที่ใช้ในการค้นหา** (Organization of Search Files) อาจจะมีผลต่อขั้นตอนของการค้นหาข้อมูล, เวลาตอบกลับ (Response Time), ความพยายามของผู้ควบคุมระบบและรวมถึงประสิทธิภาพการดึงข้อมูลของระบบ

- **ภาษาดัชนี** (Indexing Language) ภาษาดัชนีประกอบด้วยเซตของดัชนีและกฎต่างๆ ซึ่งถูกใช้เพื่อกำหนดดัชนีให้กับเอกสารและข้อคำถามในการค้นหาข้อมูลที่อยู่ระหว่างขั้นตอนของการสร้างดัชนี ค่าที่เหมาะสมกับการแสดงเนื้อหาของเอกสาร จะถูกเลือกจากภาษาดัชนีและกำหนดให้กับรายการเอกสารตามกฎที่กำหนดขึ้นมา พารามิเตอร์ที่สำคัญคือ Exhaustively และ Specificity ของภาษาดัชนี Exhaustive Indexing Language จะบรรจุดัชนีต่างๆ ที่ให้ความหมายครอบคลุมทุกๆ ขอบเขตของเรื่อง (Subject Areas) ที่มีอยู่ในกลุ่มของเอกสารทั้งหมด ซึ่ง Exhaustive Indexing Product จะหมายถึงดัชนีต่างๆ ที่กำหนดให้กับเอกสารเหล่านั้นจะสะท้อนถึงเนื้อหาของเรื่องราวของเอกสารได้อย่างถูกต้องแต่ Specific Indexing Language จะใช้ดัชนีที่มีความหมายเจาะจงมากขึ้น ขอบเขตแคบลงและถูกต้อง ดังนั้นประสิทธิภาพการค้นคืนจะถูกวัดโดยการใช้ค่า Recall และ Precision โดยค่า **Recall** จะวัดความสามารถของระบบในการที่จะดึงเอกสารที่เกี่ยวข้องออกมา ในขณะที่ **Precision** จะวัดความสามารถในการที่จะขจัดเอกสารที่ไม่เกี่ยวข้องออกไป Indexing Exhaustively ที่อยู่ในระดับสูงจะให้ค่า Recall ที่สูง ซึ่งทำให้สามารถที่จะดึงเอกสารที่เกี่ยวข้องได้มาก ในขณะเดียวกันเมื่อมีการใช้ดัชนีที่มีความเจาะจงสูง Precision ก็จะมีแนวโน้มสูงขึ้นด้วย เนื่องจากเอกสารที่ถูกดึงออกมาส่วนใหญ่จะเป็นเอกสารที่เกี่ยวข้อง สามารถกล่าวได้ว่าการที่จะให้ค่า Recall สูงนั้น ควรใช้ Exhaustive Indexing Language เพราะจะได้เอกสารต่างๆที่ครอบคลุมเนื้อหาของเรื่องราวที่กำหนดมา แต่ถ้าต้องการจะให้ Precision ที่สูงควรที่จะใช้ภาษาดัชนีที่มีความเจาะจงสูง และดัชนีควรที่จะมีตัวบ่งชี้เนื้อหา (Content Indicators) เพิ่มเติม เช่น น้ำหนักของคำหรือความสัมพันธ์ระหว่างคำกับดัชนีอื่นๆ

- **วิธีการสร้างดัชนี** (Indexing Operation) ถ้าการสร้างดัชนีถูกกระทำด้วยคนที่ได้รับการอบรมมาอย่างดี ทำให้ตัวแปรที่จะมีผลกระทบต่อวิธีการสร้างดัชนีจะมีความสัมพันธ์กัน นอกจาก

Exhaustively ของการทำดัชนีและดัชนีที่ถูกกำหนดมานั้น ยังมีความสัมพันธ์กับอิทธิพลของประสบการณ์ของผู้สร้างดัชนีต่อประสิทธิภาพและความถูกต้องของผู้กำหนดคำดัชนี

- **วิธีการค้นหา** (Search Operations) เป็นหัวข้อที่ยากในการวัดค่าเนื่องจากผู้ใช้ระบบไม่สามารถระบุความต้องการของ Recall หรือ Precision หรือมีการประเมินผลเกี่ยวกับผลลัพธ์ที่ได้จากระบบน้อยมาก แต่กลวิธีในการค้นหาข้อมูลยังมีการทบทวนเพื่อตอบสนองต่อความต้องการของ Recall หรือ Precision ของผู้ใช้ดังนั้นการวัดประสิทธิภาพการค้นหา ได้แก่ ชนิดของโครงสร้างแฟ้มข้อมูลที่ถูกนำมาใช้ การเปรียบเทียบข้อคำถามกับเอกสาร อิทธิพลของกลวิธีการค้นหาข้อมูลต่อการตอบกลับ และมาตรฐานความเกี่ยวข้องผู้ใช้ระบบ (หมายถึง มาตรฐานที่ผู้ใช้ระบบใช้ในการพิจารณาว่าผลลัพธ์ที่ได้จากการค้นหานั้นเกี่ยวข้องกับเรื่องที่ผู้ใช้ต้องการนั้นมากหรือน้อย)

- **รูปแบบของการแสดงผลลัพธ์** (Display Format) เป็นการแสดงรูปแบบทางกายภาพของเอกสารที่ถูกค้นพบโดยระบบ เพื่อที่จะสนองตอบต่อการกำหนดคำสืบค้นของผู้ใช้ รูปแบบผลลัพธ์ที่ปรากฏออกมาจะมีผลต่อปริมาณความพยายามของผู้ใช้ที่จะดูผลลัพธ์ที่ได้จากค้นหาข้อมูล และความพอใจครั้งสุดท้ายที่ได้รับจากการค้นหาข้อมูลในแต่ละครั้ง เมื่อรูปแบบของผลลัพธ์มีความสมบูรณ์มากแสดงว่างานการประเมินความเกี่ยวข้อง (Relevance Assessment Task) สำหรับผู้ใช้ก็จะยิ่งง่ายขึ้น แต่ในทางตรงข้ามขณะที่ผลลัพธ์ได้ถูกลำดับจากขนาดไฟล์เล็กๆ และค่อยๆเพิ่มขนาดไฟล์ซึ่งจะทำให้เวลาที่ต้องการในการที่จะตรวจสอบผลลัพธ์จากการค้นหาจะเพิ่มขึ้นด้วย

5.2 อภิปรายผลการวิจัย

งานวิจัยนี้นำเสนอกลไกของกระบวนการต่อจากการทำงานของเสิร์ชเอนจิน ที่จะได้เป็นผลลัพธ์รายการของแหล่งข้อมูลจากระบบการบริหารจัดการเอกสาร และพัฒนาเว็บแอปพลิเคชันเพื่อใช้ในการออกแบบและแสดงผลให้กับผู้ใช้งาน การประเมินประสิทธิภาพระบบจะใช้ทฤษฎีการค้นคืนสารสนเทศในด้านความถูกต้องแม่นยำของระบบ โดยการวัดค่า Recall และค่า Precision [8] และแยกพิจารณาพารามิเตอร์จากจำนวนคำศัพท์ที่เกี่ยวข้องเป็นกลุ่ม 0-2 คำ, 3-5 คำและ6-10 คำขึ้นไป พบว่าในการค้นคืนเอกสารที่ใช้ในการทดลองทั้งหมด 105 เอกสาร ผลการวัดค่า Recall เฉลี่ยเท่ากับ 84% และค่า Precision เฉลี่ยเท่ากับ 83% เพื่อนำไปใช้ในการประมวลผลค่าเอฟ [8] จะได้ค่าเฉลี่ย 83% ซึ่งอยู่ในระดับดีปานกลาง ซึ่งการให้ค่าน้ำหนักของการประเมินค่าเอฟของงานวิจัยนี้กำหนดค่า $\beta = 1$ จากสมการ (2.4) ในบทที่ 2 จากตารางที่ 5.1 ผลการทดสอบพบว่าจำนวนพารามิเตอร์ของคำศัพท์ที่เกี่ยวข้องมีผลต่อค่า Recall และค่า Precision ดังนั้นปัจจัยนี้จะมีผลกระทบต่อประสิทธิภาพในการประมวลผลความถูกต้องแม่นยำและมีผลต่อการนำผลลัพธ์ที่ได้มาแสดงให้กับผู้ใช้เพื่อพิจารณาตัดสินว่าตรงกับความต้องการหรือไม่ ในกระบวนการวิเคราะห์และประเมินคำศัพท์ดัชนี (Indexing Vocabulary) ที่ใช้ค้นหาเพื่อการวิเคราะห์สกัดบริบทที่เป็นสาระสำคัญโดยมีการคาดหวังว่าจะตรงกับความต้องการของผู้ใช้งาน เทคนิคที่ถูกนำมาใช้วิเคราะห์คำสำคัญจะใช้การค้นหาข้อมูลคำจากเอกสารเอกซ์เอ็มแอลและตรวจสอบคำศัพท์ควบคุมเพื่อวิเคราะห์หาคำที่มีความสัมพันธ์และพิจารณาจำนวนกลุ่มคำที่เกี่ยวข้องมาใช้ร่วมในการค้นหาเอกสารและบริบทในเอกสาร ซึ่งจำนวนคำที่

เหมาะสมที่อยู่ที่ 3-5 คำ เนื่องจากจำนวนกลุ่มคำที่เกี่ยวข้องเพิ่มขึ้นจะมีผลต่อค่า Recall สูงขึ้นทำให้ได้จำนวนเอกสารที่เกี่ยวข้องสัมพันธ์กันเพิ่มจำนวนมากขึ้นด้วยแต่ในขณะเดียวกันพบว่ามีค่าแปรผกผันกับ Precision ให้ค่าลดลง เนื่องจากเนื้อหาสาระของเอกสารจะเพิ่มความหลากหลายและไม่ตรงประเด็นมากขึ้นด้วย ในการประมวลผลชุดข้อมูลที่เป็นผลลัพธ์จากการดำเนินการของภาษาเอกซ์คิวรีที่ได้เป็นจำนวนความถี่ของไอเท็มข้อมูลที่เป็นเงื่อนไขในการค้นหาและถูกนำไปวิเคราะห์ความคล้ายทั้งแบบภายในเอกสารและระหว่างเอกสารทำให้เพิ่มมิติในการพิจารณาความสัมพันธ์มากขึ้น ผลที่ได้คือบริบทที่ตรงกับความต้องการของผู้ใช้งาน

ตารางที่ 5.1 ผลการทดสอบระบบการค้นคืนของค่า Recall, Precision และค่าเอฟ

คำสำคัญ (Keyword)	จำนวนคำศัพท์ควบคุม (Controlled Vocabulary)	ค่ารีคอล (% Recall)	ค่าพรีซิชัน (% Precision)	ค่าน้ำหนัก-เบต้า (β)	%ค่าเอฟ (F-measure)
Interest	0-2	74	76	1	75
	3-5	86	89	1	87
	6-10	92	70	1	80
Loan	0-2	75	81	1	78
	3-5	83	90	1	86
	6-10	85	72	1	78
Finance	0-2	78	89	1	83
	3-5	89	91	1	90
	6-10	85	80	1	82
Account	0-2	71	88	1	79
	3-5	94	95	1	94
	6-10	94	78	1	85
เฉลี่ย (Average) (%)		84	83	-	83

ตารางที่ 5.2 ตัวอย่างการแจกแจงคำศัพท์ควบคุมที่เป็นคำเหมือนใช้ในการทดสอบ

คำสำคัญ (Keyword)	รายการคำศัพท์ควบคุมที่เป็นคำเหมือน(Synonym or Thesaurus)		
	0-2 คำ	3-5 คำ	6-10 คำ
Interest	Debt	Activity Concern Debt	Concern Debt Importance Return Profits Moment Substance Relevance Stake
Loan	Credit Expense	Credit Expense Financing Investment Trust	Allowance Advance Accommodation Credit Floater Investment Mortgage Payment Trust

ตารางที่ 5.2 ตัวอย่างการแจกแจงคำศัพท์ควบคุมที่เป็นคำเหมือนใช้ในการทดสอบ (ต่อ)

คำสำคัญ (Keyword)	รายการคำศัพท์ควบคุมที่เป็นคำเหมือน(Synonym or Thesaurus)		
	0-2 คำ	3-5 คำ	6-10 คำ
Finance	Banking Business	Banking Business Commerce Money	Account Banking Commerce Investment Funds Saving Settle Subsidize
Account	Detail Explanation	Detail Explanation Narrative Report	Annal Bulletin Detail History Narrative Report Tale Version

ตารางที่ 5.2 เป็นรายการข้อมูลคำศัพท์ควบคุมที่มีความหมายใกล้เคียงกับคำสำคัญซึ่งจะถูกนำมาใช้ในการประมวลผลความคล้ายกันของบริบทในเอกสาร โดยพิจารณากลุ่มคำประกอบด้วย 1. กลุ่มคำศัพท์ระหว่าง 0-2 คำ 2. กลุ่มคำศัพท์ระหว่าง 3-5 คำ และกลุ่มคำศัพท์ระหว่าง 6-10 คำ โดยผลลัพธ์จากการคำนวณความคล้ายจะเป็นค่าผลรวมของกลุ่มคำดังกล่าวร่วมกับคำสำคัญ ผลลัพธ์รายการข้อมูลแสดงในตารางที่ 5.1 และในตารางที่ 5.3 จากประเด็นหัวข้อการประเมินผลระบบการค้นคืนสารสนเทศ [8] พบว่างานวิจัยนี้มีความสามารถด้านความถูกต้องแม่นยำโดยพิจารณาจากค่า Recall และค่า Precision ที่ได้ค่าเฉลี่ยอยู่ในระดับดีปานกลาง ด้านความครบถ้วนสมบูรณ์และความครอบคลุมพิจารณาจากค่า Recall ที่อยู่ในระดับปานกลางในการค้นหาบริบทที่เกี่ยวข้อง ด้านการนำเสนอมีการออกแบบระบบให้ผู้ใช้งานสามารถกำหนดรูปแบบการแสดงผลตามความต้องการ และในส่วนที่ระบบขาดความสมบูรณ์ในด้านความเร็วในการประมวลผลและการควบคุมค่าใช้จ่ายนั้นยังต้องมีการปรับปรุงประสิทธิภาพ

ตารางที่ 5.3 หัวข้อประเด็นในการประเมินประสิทธิภาพระบบการค้นคืน

หัวข้อการประเมินผล	มี	ไม่มี
ความเร็ว		X
ความถูกต้องแม่นยำ	✓	
ความครบถ้วนสมบูรณ์	✓	
ความสำคัญก่อนหลัง	✓	
ความครอบคลุม	✓	
การนำเสนอ	✓	
ค่าใช้จ่าย		X

บทที่ 6
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

กรอบงานสารสนเทศควบรวมของงานวิจัยนี้มีการออกแบบและพัฒนากลไกของการประมวลผลบริบทที่มีการรวบรวมและแสดงผลสารสนเทศที่เหมาะสมกับความต้องการของผู้ใช้งานมากที่สุด การนำเสนอเพื่อความสะดวกในการพัฒนาระบบและการใช้งานของผู้ใช้จะต้องใช้เครื่องมือในการพัฒนามาช่วยเสริมด้านการออกแบบและปรับแต่งตำแหน่งการแสดงผลตามความต้องการของผู้ใช้งาน เช่น ไมโครซอฟต์แชร์พอยต์ (Microsoft SharePoint) หรืออื่นๆที่เทียบเท่า

6.2 ข้อจำกัดของงานวิจัย

งานวิจัยนี้ยังมีข้อจำกัดในเรื่องภาษาของบริบทที่จะถูกนำไปใช้ในการประมวลผลจะต้องเป็นภาษาอังกฤษเท่านั้น บริบทที่ใช้จะอยู่ในรูปแบบเอกสารที่มีโครงสร้างเอกซ์เอ็มแอลและอยู่ในกรอบของหน่วยงานภายในองค์กรเท่านั้น การควบคุมคำศัพท์ที่นำมาใช้จะถูกเก็บเป็นดัชนีของข้อมูลทำให้ขาดน้ำหนักด้านความหลากหลายของคำที่เกี่ยวข้อง และการวัดค่าด้านประสิทธิภาพในการประมวลผลจะขึ้นอยู่กับจำนวนของเอกสารที่นำมาใช้ในการประมวลผล โดยได้ผลลัพธ์จากเสิร์ชเอนจิน

6.3 แนวทางการวิจัยต่อ

ในอนาคตสิ่งที่งานวิจัยนี้ต้องพัฒนาเพิ่มเติมในเรื่องภาษาของบริบทจะต้องสนับสนุนภาษาอื่นๆ เช่น ภาษาไทย ภาษาจีน ซึ่งต้องอาศัยทฤษฎีของภาษามาช่วยแก้ไข เพิ่มเติมในการรวบรวมองค์ประกอบอื่นๆ ที่เกี่ยวข้องในเอกสาร เช่น รูปภาพ ซึ่งจะใช้ทฤษฎีเอกซ์เอ็มไอ (XMI) มาช่วยสนับสนุน และเรื่องของประสิทธิภาพความรวดเร็วในการประมวลผลให้ดียิ่งขึ้นและสามารถปรับปรุงประสิทธิภาพการประมวลผล ในด้านความถูกต้องแม่นยำของผลลัพธ์ด้วยการอาศัยข้อมูลในการประมวลผลครั้งก่อนหน้า เพื่อให้ตรงกับความต้องการของผู้ใช้งานแบบอัตโนมัติ

รายการอ้างอิง

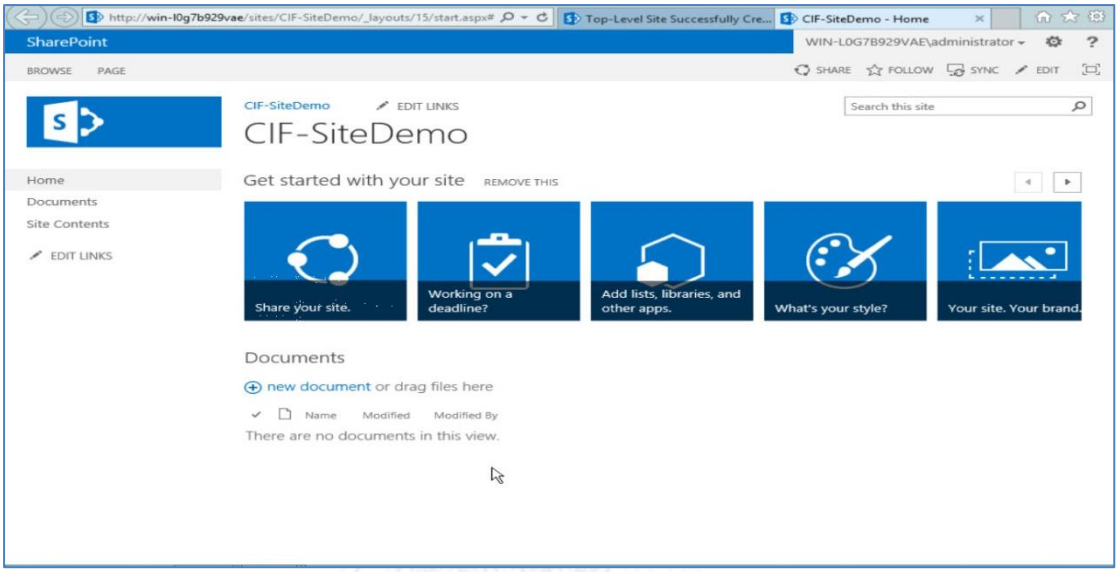
1. Liu, S., McMahon, C. A., Darlington, M. J., Culley, S. J., and Wild, P. J. *An automatic mark-up approach for structured document retrieval in engineering design.* The International Journal of Advanced Manufacturing Technology, 2008. **38**(3-4): p. 418-425.
2. Kotsakis, E., *Structured information retrieval in XML documents.* Proceedings of the 2002 ACM symposium on applied computing, 2002.
3. Fasheng, L., and Lu, X. *Survey on text clustering algorithm -Research present situation of text clustering algorithm*. in *Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on*. 2011.
4. Salton, G., Wong, A., and Yang, C. S. *A vector space model for automatic indexing.* Communications of the ACM, 1975. **18**(11): p. 613-620.
5. Wolfgang, M., Tobias, K., and Timo, M. *Vocabulary Patterns in Free-for-all Collaborative Indexing Systems*. in *Proceedings of the First International Workshop on Emergent Semantics and Ontology Evolution, ESOE 2007, co-located with ISWC 2007 + ASWC 2007 (Liming Chen, Philippe Cudré-Mauroux, Peter Haase, Andreas Hotho, Ernie Ong eds.)*. 2007. Busan, Korea: CEUR-WS.org.
6. Singhal, A. *Modern Information Retrieval: A Brief Overview.* IEEE Data Eng. Bull., 2001. **24**(4): p. 35-43.
7. Rangrej, A., Kulkarni, S. and Tendulkar, A. V. *Comparative study of clustering techniques for short text documents*, in *Proceedings of the 20th international conference companion on World wide web*. 2011, ACM: Hyderabad, India. p. 111-112.
8. ศุภชัย ตั้งวงศ์ศานต์. ระบบการจัดเก็บและการสืบค้นสารสนเทศด้วยคอมพิวเตอร์. พิมพ์ครั้งที่ 2, 2553, กรุงเทพมหานคร: โรงพิมพ์พิทักษ์การพิมพ์.
9. กานดา สายแก้ว. เอกซ์เอ็มแอล = *XML*. พิมพ์ครั้งที่ 1, 2553, ขอนแก่น: โรงพิมพ์มหาวิทยาลัยขอนแก่น.
10. (W3C), W.W.W.C. *XML Schema*. 2005 [cited 2014, Febuary 8]; Available from: http://www.3c.org/TR/xmlschema-0.
11. Erik, T.R. *Learning XML*. 2001, O'Reilly & Associates: United States of America, O'Reilly & Associates.
12. เอกพล จีรังสุวรรณ, สมนึก คีรีโต. การวิเคราะห์และเปรียบเทียบภาษาสืบค้นสำหรับ *XML* ระหว่าง *XQuery* กับ *XSLT.* NECTECT Technical Journal, 2544.

13. Ramkhamhaeng University, *E-Book Ramkhamhaeng University*. E-Book Ramkhamhaeng University [PDF] 2001 [cited 2014, February 8]; Available from: http://e-book.ram.edu/e-book/c/CT477/CT477-6.pdf.
14. Cambridge University Press, *Introduction to Information Retrieval*. Introduction to Information Retrieval 2008 [cited 2014, February 8]; Available from: http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html.
15. Microsoft Corporation. *OpenXMLDeveloper*. 2013 [cited 2014, February 8]; Available from: http://openxmldeveloper.org.
16. Dologite, D.G. *Developing knowledge-based system using VP-Expert*. 1993, USA: Semline, Inc.
17. Community, T.C.L. *Journal Code 4 Lib*. 2003 [cited 2014, March 23]; Available from: http://www.journal.code4lib.org.
18. Fung, K.Y. *XSLT : working with XML and HTML*. 2000, The United States of America: Addison-Wesley.
19. Project, O. *Terminology Services: Experimental for Controlled Vocabulary*. 2008 [cited 2014, March 23]; Available from: http://tspilot.oclc.org/resources/index.html.
20. Bray, T., Paoli, J. Sperberg-McQueen C. M., et. al. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 2008, November 26 [cited 2014, March 23]; Available from: http://www.w3.org/TR/REC-xml/.

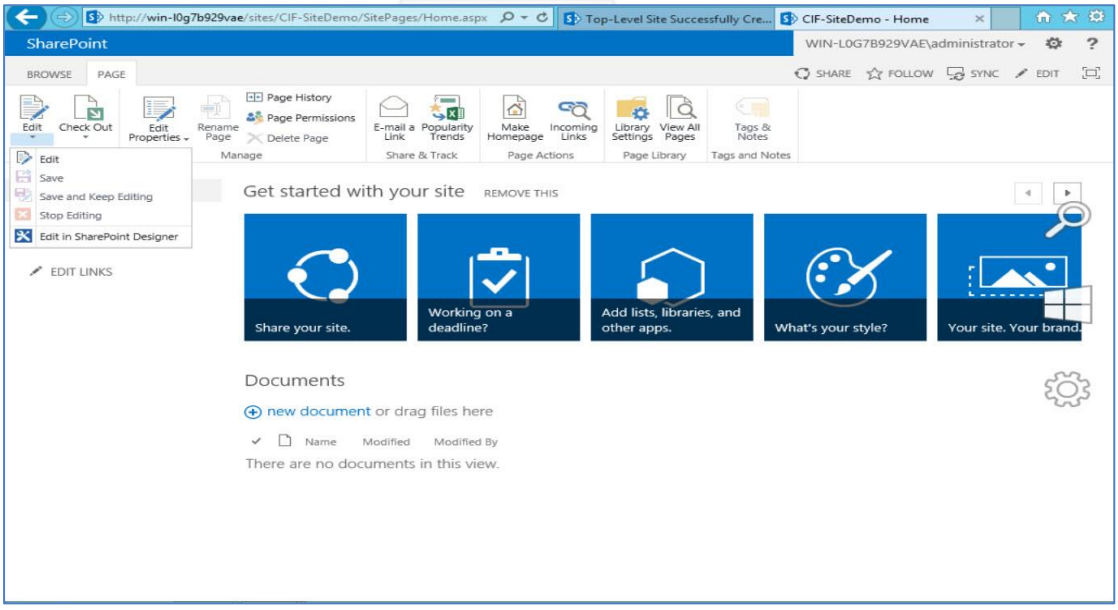
ภาคผนวก

ภาคผนวก ก.
การออกแบบไซต์ในแชร์พอยต์

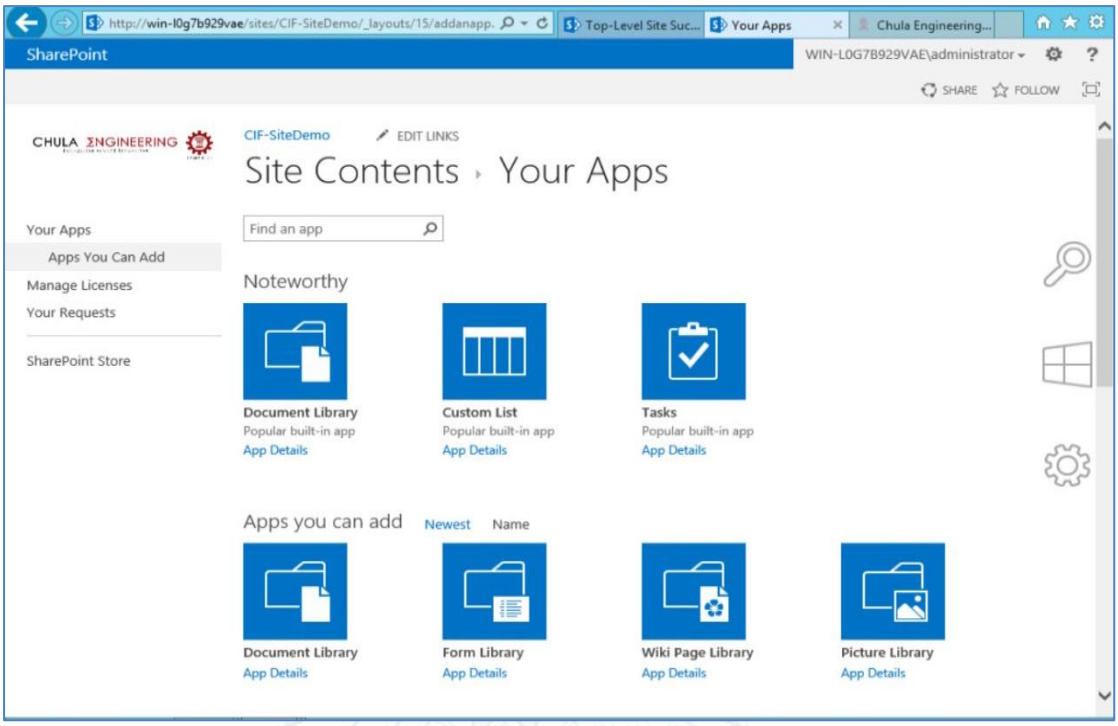
ก.1 การสร้างไซต์ CIF-SiteDemo



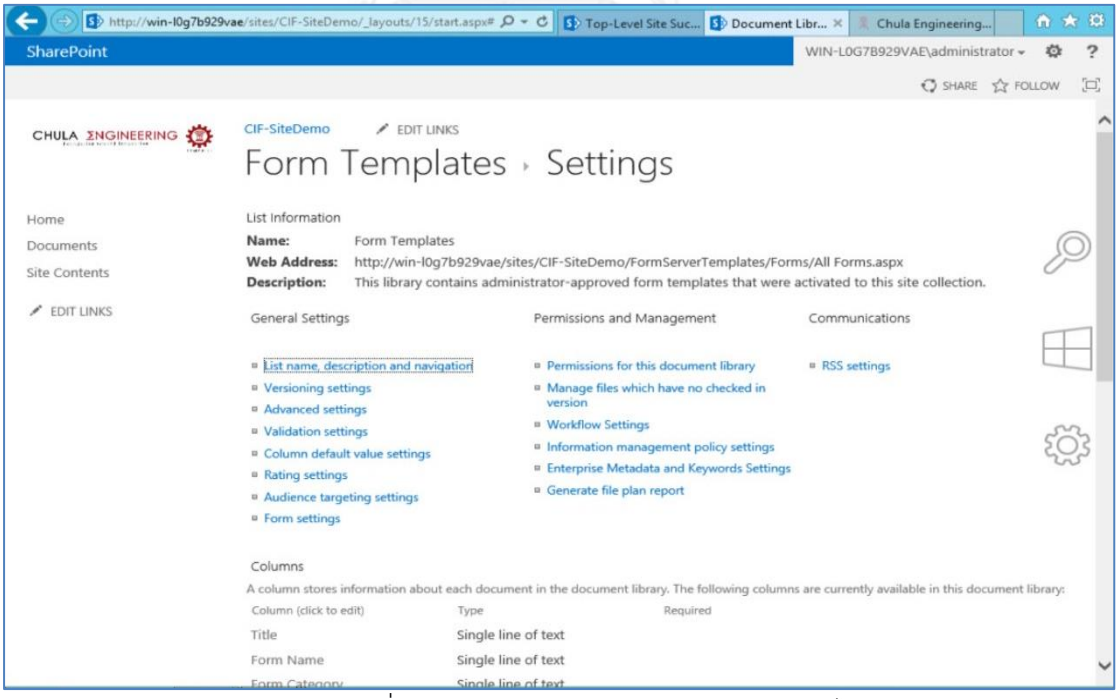
ภาพที่ ก-1 ตัวอย่างเพจไซต์ CIF-SiteDemo



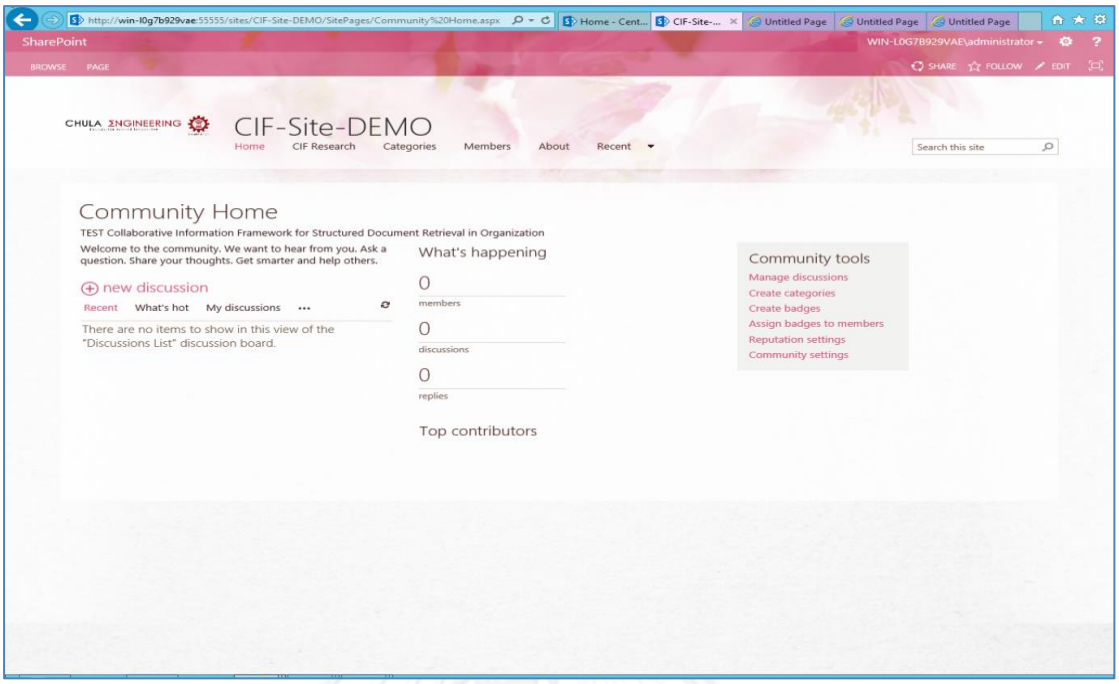
ภาพที่ ก-2 รายการเมนูที่สามารถแก้ไข เปลี่ยนแปลงรูปแบบไซต์



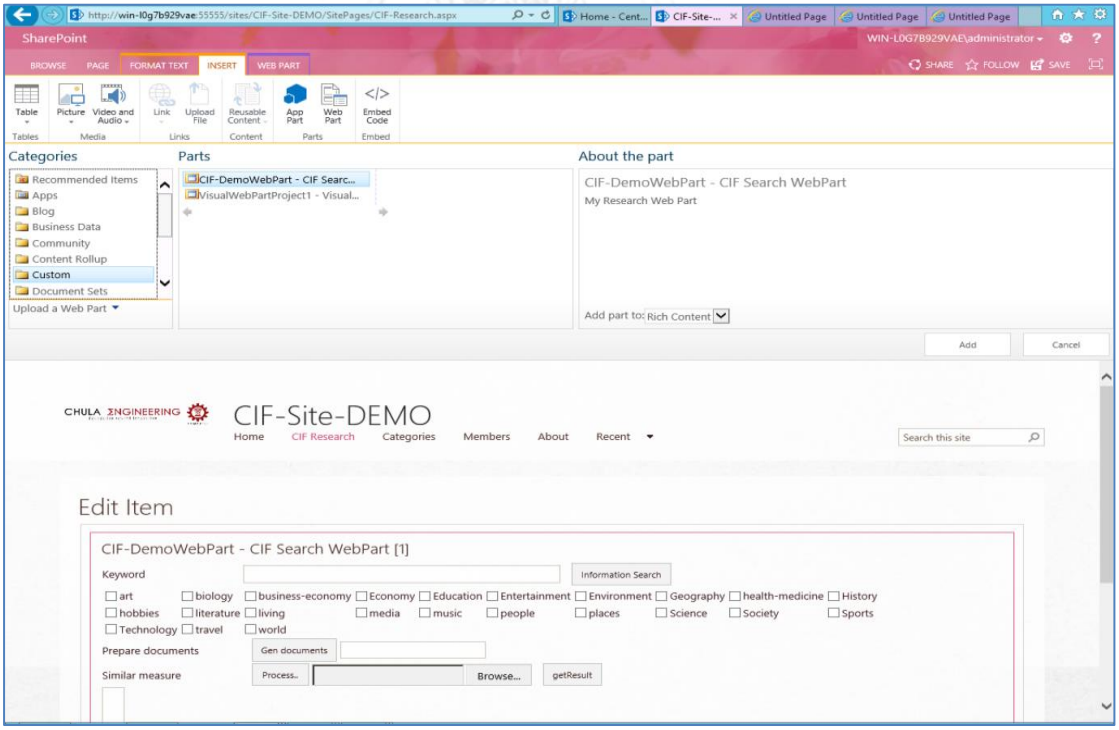
ภาพที่ ก-3 รายการแอปพลิเคชันภายในไซต์ที่จะถูกเลือกใช้งาน



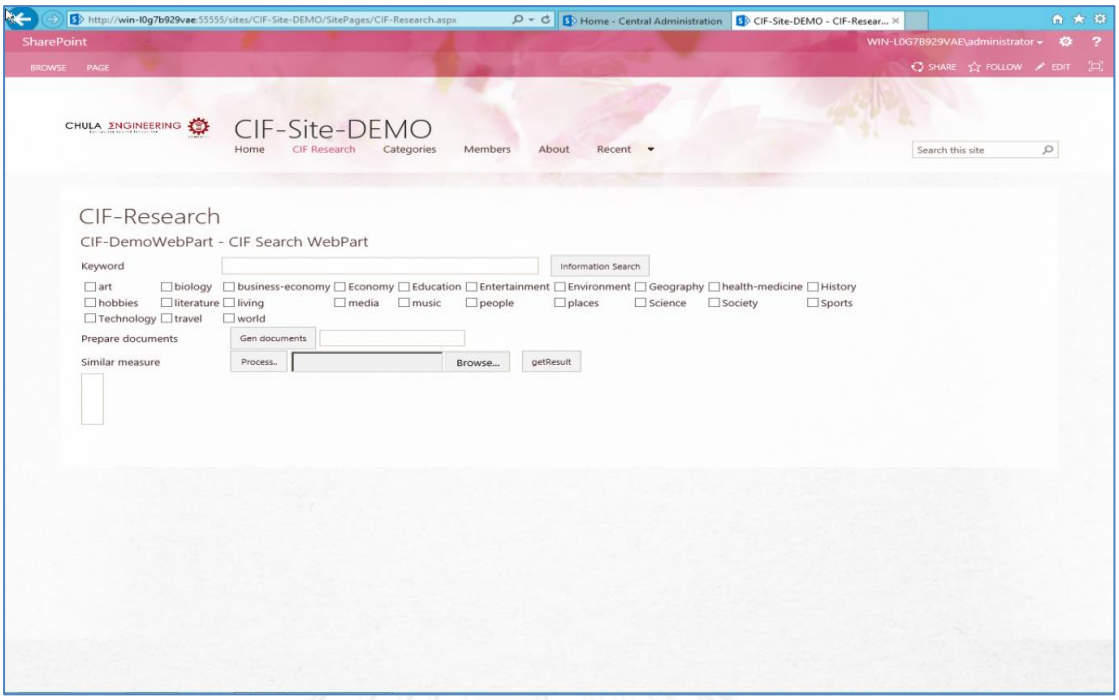
ภาพที่ ก-4 การกำหนดฟอร์มแผ่นแบบของเพจไซต์



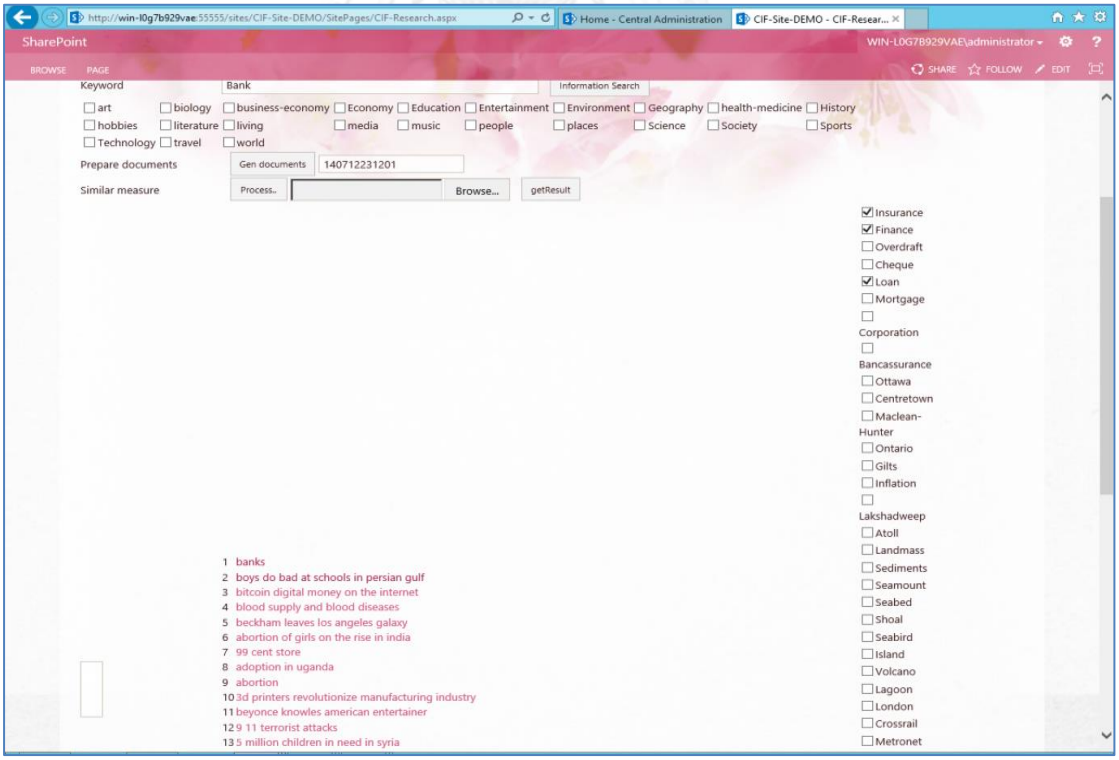
ภาพที่ ก-5 ตัวอย่างโฮมเพจของ CIF-Site-DEMO ที่สร้างจากแผ่นแบบของเพจไซต์ Community



ภาพที่ ก-6 ตัวอย่างการแทรกเว็บพาร์ท CIF-DemoWebPart ในเพจไซต์ CIF Research



ภาพที่ ก-7 ตัวอย่างเว็บเพจ CIF-Research ที่มีการออกแบบโดยแทรกเพิ่มเว็บพาร์ท



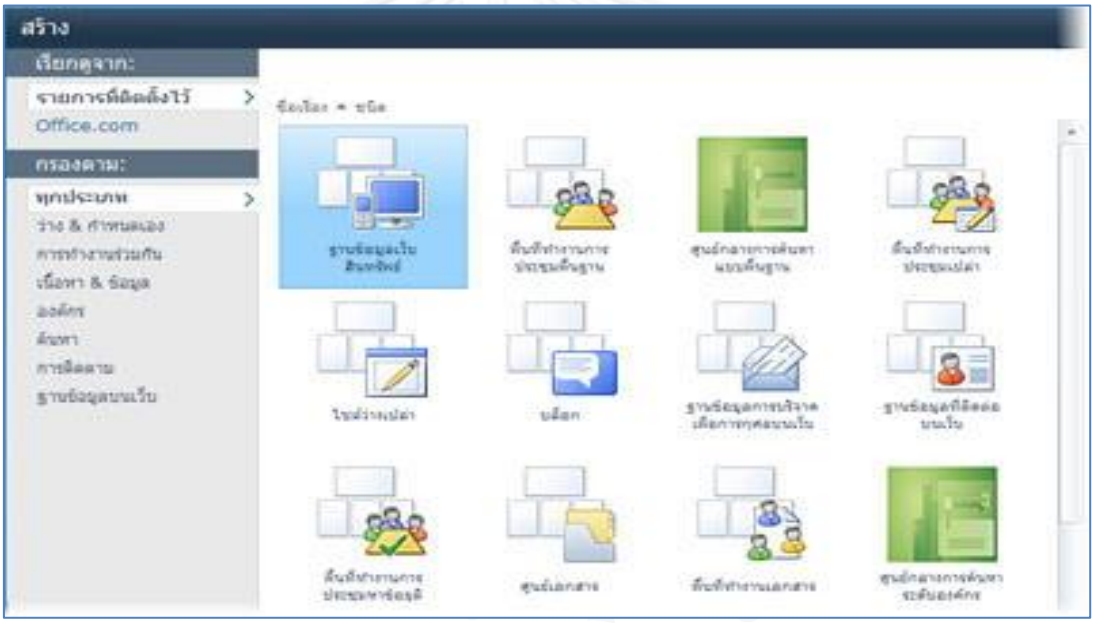
ภาพที่ ก-8 ตัวอย่างการทดสอบด้วยคำค้นหาและผลลัพธ์ที่ได้จากการประมวลผล

ภาคผนวก ข.
การสร้างและการเปิดไซต์ในแชร์พอยต์

ข.1 การบันทึกไซต์แชร์พอยต์เป็นต้นแบบ

- **ต้นแบบไซต์แชร์พอยต์**

ต้นแบบไซต์แชร์พอยต์ คือข้อกำหนดที่มีการสร้างไว้ล่วงหน้าโดยได้รับการออกแบบตามความต้องการของผู้ออกแบบและพัฒนา ซึ่งสามารถใช้ต้นแบบสร้างไซต์แชร์พอยต์ของผู้ออกแบบและพัฒนาเองในเบื้องต้น และกำหนดไซต์นั้นเองตามที่ความต้องการ ประเภทของต้นแบบไซต์เริ่มต้นอย่างเช่น ไซต์ทีม ไซต์บล็อก และไซต์งานกลุ่ม ดังแสดงในรูป ข-1



ภาพที่ ข-1 รายการต้นแบบที่จะถูกสร้างเป็นไซต์ต้นแบบตามข้อกำหนด

ผู้ใช้สามารถสร้างต้นแบบไซต์ได้เองโดยพิจารณาตามไซต์ที่ผู้ออกแบบและพัฒนาสร้างขึ้น ซึ่งเป็นคุณลักษณะที่มีประสิทธิภาพอย่างมากในแชร์พอยต์ เนื่องจากทำให้ผู้พัฒนาสามารถสร้างโซลูชันแบบกำหนดเอง และใช้โซลูชันนั้นร่วมกันกับผู้พัฒนาอื่นๆภายในองค์กร หรือกับองค์กรภายนอกได้ นอกจากนี้ผู้พัฒนายังสามารถจัดแพคเกจไซต์และเปิดในสภาพแวดล้อมหรือโปรแกรมประยุกต์อื่นๆ เช่น ไมโครซอฟต์วิชวลสตูดิโอ และทำการเพิ่มข้อกำหนดเองในโปรแกรมดังกล่าว เมื่อผู้พัฒนาบันทึกไซต์ของผู้พัฒนาเป็นต้นแบบ นั่นคือผู้พัฒนาได้สร้าง Web Solution Package หรือ WSP ขึ้นโดย WSP เป็นแฟ้ม CAB ที่มีรายการโซลูชันที่ผู้พัฒนาสร้างจะถูกจัดเก็บไว้ในแกลเลอรีโซลูชัน สำหรับไซต์

คอลเลกชันของแชร์พอยต์จากตำแหน่งนั้น ผู้พัฒนาสามารถดาวน์โหลดสำเนาของโซลูชันหรือเปิดใช้งานได้บนเซิร์ฟเวอร์

- **ลักษณะการบันทึกไซต์ต้นแบบ**

ในการบันทึกไซต์แชร์พอยต์ของผู้พัฒนาเป็นต้นแบบ นั่นคือการบันทึกเฟรมเวิร์กโดยรวมของไซต์ที่ประกอบด้วย ลิสต์และไลบรารี มุมมองแบบฟอร์ม และเวิร์กโฟลว์ ทำให้ผู้พัฒนาสามารถรวมเนื้อหาของไซต์ในต้นแบบ เช่น เอกสารที่เก็บในไลบรารีเอกสาร ซึ่งเป็นประโยชน์ในการแสดงเนื้อหาตัวอย่างเพื่อให้ผู้ใช้สามารถเริ่มต้นใช้งานง่าย โดยต้นแบบจะรวมและสนับสนุนวัตถุส่วนอื่นๆในไซต์ อาจจะมีคุณลักษณะจำนวนหนึ่งที่ไม่ได้รับการสนับสนุน ดังแสดงในตารางที่ ข-1 เป็นข้อมูลสรุปเกี่ยวกับสิ่งที่มีและไม่มีอยู่ในโซลูชันหรือต้นแบบไซต์โดยทั่วไป

ตารางที่ ข-1 รายการคุณสมบัติของ Web Solution Package ที่ผู้ใช้นำไปใช้งาน

สิ่งที่มีอยู่ใน WSP สำหรับโซลูชันของผู้ใช้	**สิ่งที่ไม่มีอยู่ใน WSP สำหรับโซลูชันของผู้ใช้**
• รายการ • ไลบรารี • รายการภายนอก • การเชื่อมต่อแหล่งข้อมูล • มุมมองรายการและมุมมองข้อมูล • ฟอร์มแบบกำหนดเอง • เวิร์กโฟลว์ • ชนิดเนื้อหา • การกระทำแบบกำหนดเอง • การนำทาง • เพจไซต์ • เพจต้นแบบ • มอดูล • ต้นแบบเว็บ	• สิทธิ์ที่กำหนดเอง • การเรียกใช้อินสแตนซ์เวิร์กโฟลว์ • ประวัติรุ่นของรายการ • งานเวิร์กโฟลว์ที่เกี่ยวข้องกับการเรียกใช้เวิร์กโฟลว์ • ค่าเขตข้อมูลบุคคล/กลุ่ม • ค่าเขตข้อมูล Taxonomy • การประกาศเพจและการประกาศไซต์ • ไซต์ของผู้พัฒนา

ข.2. การใช้งานต้นแบบแชร์พอยต์

การบันทึกไซต์เป็นต้นแบบเป็นคุณลักษณะที่มีประสิทธิภาพสูง เนื่องจากสามารถใช้ไซต์แบบกำหนดเองในแชร์พอยต์ได้ในหลายลักษณะ ซึ่งประโยชน์ของการบันทึกไซต์เป็นต้นแบบในแชร์พอยต์รายละเอียดดังนี้

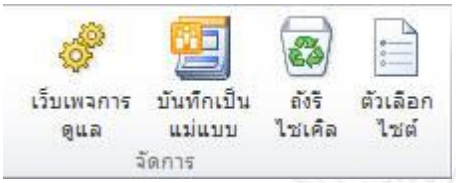
- **ไซต์แชร์พอยต์ แบบกำหนดเองสามารถปรับใช้เป็นโซลูชัน** การบันทึกและเปิดใช้งานต้นแบบในแกลเลอรีโซลูชันทำผู้ใช้งานสามารถสร้างไซต์ใหม่จากต้นแบบ ผู้พัฒนาไม่จำเป็นต้องใช้วิชวลสตูดิโอในการสร้างโซลูชันของผู้พัฒนา แต่ผู้พัฒนาจะต้องเข้าถึงเซิร์ฟเวอร์โดยตรงและเรียกใช้คำสั่งของผู้ดูแลเซิร์ฟเวอร์ โดยสามารถบันทึกไซต์เป็นต้นแบบและเปิดใช้งานต้นแบบ
- **ไซต์แชร์พอยต์ แบบกำหนดเองเป็นไซต์ที่เคลื่อนย้ายได้** ผู้พัฒนาสามารถดาวน์โหลดแฟ้ม wsp, เก็บไว้และปรับใช้ในสภาพแวดล้อมของแชร์พอยต์อื่นๆเพิ่มเติม การกำหนดไซต์เองทั้งหมดของผู้พัฒนาจะถูกเก็บไว้ในแฟ้ม ซึ่งทำให้สามารถใช้งานได้สะดวก
- **ไซต์แชร์พอยต์ แบบกำหนดเองเป็นไซต์ที่สามารถขยายได้** Web Solution Package ทำให้ผู้พัฒนาสามารถเปิดไซต์แบบกำหนดเองของผู้พัฒนาในวิชวลสตูดิโอ, ดำเนินการกำหนดเองเพื่อทำการพัฒนาเพิ่มเติมให้กับต้นแบบ แล้วทำการปรับใช้ต้นแบบนั้นกับแชร์พอยต์ ทำให้การพัฒนาไซต์แชร์พอยต์สามารถดำเนินการผ่านวงจรการพัฒนาไซต์ที่ประกอบด้วย แชร์พอยต์ ดีไซเนอร์ 2013, ไมโครซอฟต์วิชวลสตูดิโอและเบราว์เซอร์

- **การบันทึกไซต์เป็นต้นแบบ**

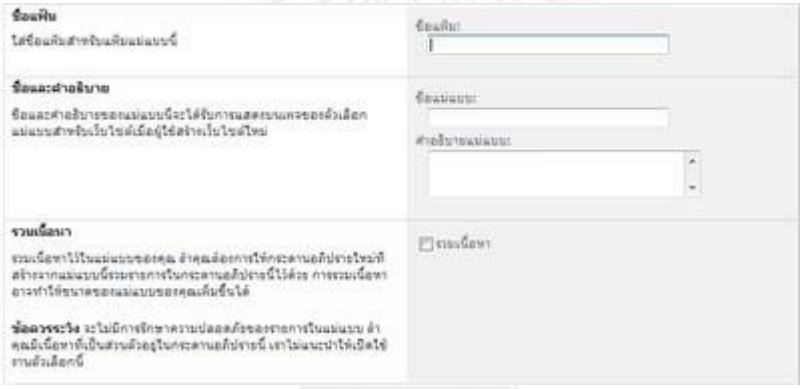
ผู้พัฒนาสามารถบันทึกไซต์เป็นต้นแบบโดยใช้ตัวเลือกต้นแบบ (บันทึกไซต์เป็นต้นแบบ) บนเพจการตั้งค่าไซต์ในแชร์พอยต์ และแชร์พอยต์ ดีไซเนอร์ 2013 เป็นการดำเนินการที่ง่ายและสะดวก เพราะมีตัวเลือกเพื่อบันทึกเป็นต้นแบบใน Ribbon ซึ่งทำให้ผู้พัฒนาไปยังเพจที่ต้องการในแชร์พอยต์ เมื่อผู้พัฒนาบันทึกต้นแบบ แฟ้มโซลูชันจะถูกสร้างและเก็บไว้ในแกลเลอรีโซลูชัน ซึ่งเป็นที่ที่ผู้พัฒนาสามารถดาวน์โหลดหรือเปิดใช้งานโซลูชันได้

การบันทึกไซต์ของผู้พัฒนาเป็นต้นแบบโดยใช้แชร์พอยต์ ดีไซเนอร์ 2013 ให้ดำเนินการขั้นตอนต่อไปนี้

1. เปิดไซต์ของผู้พัฒนาใน แชร์พอยต์ ดีไซเนอร์ 2013
2. บนแท็บไซต์ในกลุ่มจัดการ คลิกบันทึกเป็นต้นแบบ



3. การดำเนินการนี้จะนำผู้พัฒนาไปที่เพจ **บันทึกเป็นต้นแบบ** ในแชร์พอยต์



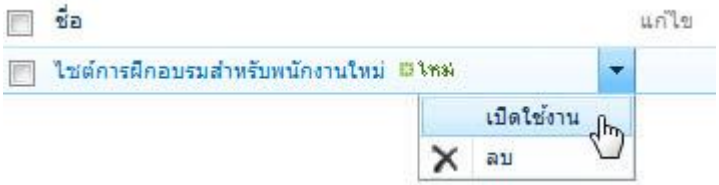
4. ระบุชื่อเพื่อนำไปใช้กับแฟ้มต้นแบบในเขตข้อมูล ชื่อแฟ้ม
5. ระบุชื่อและคำอธิบายสำหรับต้นแบบในเขตข้อมูล ชื่อต้นแบบ และคำอธิบายต้นแบบ
6. เมื่อต้องการรวมเนื้อหาของไซต์ลงในต้นแบบไซต์ ให้เลือกกล่อง รวมเนื้อหา คลิกตกลงเพื่อบันทึกต้นแบบ
7. ถ้าคอมโพเนนต์ทั้งหมดบนไซต์ถูกต้อง ต้นแบบจะถูกสร้างขึ้น และผู้พัฒนาจะเห็นข้อความที่ระบุว่า การดำเนินการเสร็จสมบูรณ์
8. เมื่อต้องการดาวน์โหลดหรือเปิดใช้งานโซลูชันจากแกลเลอรีโซลูชัน คลิกการเชื่อมโยง แกลเลอรีโซลูชันของผู้ใช้และทำตามขั้นตอนในกระบวนงานด้านล่าง
9. หรือเมื่อต้องการกลับไปยังไซต์ของผู้พัฒนา คลิกตกลง

- **การเปิดใช้งานต้นแบบไซต์ในแกลลอรีโซลูชัน**

เมื่อผู้พัฒนาได้บันทึกไซต์ของผู้พัฒนาเป็นต้นแบบแฟ้มโซลูชัน (.wsp) จะถูกสร้างและจัดเก็บไว้ในแกลเลอรีโซลูชันสำหรับไซต์คอลเลกชัน ผู้พัฒนาสามารถดาวน์โหลดหรือเปิดใช้งานโซลูชันได้

เมื่อต้องการเปิดใช้งานต้นแบบไซต์ของผู้พัฒนา ให้ดำเนินการขั้นตอนต่อไปนี้

1. เรียกดูไซต์ระดับบนสุดของไซต์คอลเลกชันของผู้พัฒนาในแชร์พอยต์
2. คลิกการกระทำในไซต์ แล้วเลือก การตั้งค่าไซต์
3. ภายใต้ แกลเลอรี คลิกเลือก โซลูชัน
4. การเปิดใช้งานโซลูชันของผู้พัฒนา คลิกเมนูแบบดรอปดาวน์ที่อยู่ด้านข้างของโซลูชันของผู้พัฒนา แล้วเลือก เปิดใช้งาน



5. บนหน้าจอการยืนยันการเปิดใช้งานโซลูชัน คลิกเลือกเมนูรายการ เปิดใช้งาน ขณะนี้โซลูชันของผู้พัฒนามีสถานะเป็นเปิดใช้งานแล้วในแกลเลอรีโซลูชัน
6. เมื่อต้องการดาวน์โหลดโซลูชัน คลิกเลือกที่ชื่อของโซลูชันในแกลเลอรีโซลูชัน
7. ในกล่องโต้ตอบ ดาวน์โหลดแฟ้ม คลิกบันทึกและเรียกดูตำแหน่งที่ตั้งที่ผู้พัฒนาต้องการบันทึกโซลูชัน

เมื่ออัปโหลดและเปิดใช้งานโซลูชันในแกลเลอรีโซลูชันแล้ว ผู้ใช้จะเห็นโซลูชันในรูปของต้นแบบที่พร้อมใช้งานบนเพจ การสร้างไซต์ในแชร์พอยต์โดยผู้พัฒนาสามารถเลือกโซลูชันและสร้างไซต์ใหม่จากโซลูชันนั้น ซึ่งจะสืบทอดคอมโพเนนต์ของไซต์ โครงสร้าง เวิร์กโฟลว์ และอื่นๆ หรือผู้พัฒนาสามารถดาวน์โหลดโซลูชันของผู้พัฒนาจากแกลเลอรีโซลูชันและปรับใช้ในสภาพแวดล้อมแชร์พอยต์อื่นๆ หรือจะเปิดขึ้นในไมโครซอฟต์วิชวลสตูดิโอที่สนับสนุน WSP

ประวัติผู้เขียนวิทยานิพนธ์

จบปริญญาตรีหลักสูตรวิทยาศาสตรบัณฑิต(วท.บ.) สาขาวิชาคณิตศาสตร์ จากมหาวิทยาลัยหอการค้าไทย ทำงานตำแหน่งพนักงานเทคโนโลยีสารสนเทศ ฝ่ายแผนงานและพัฒนาเทคโนโลยีสารสนเทศในบริษัทเอกชน และปัจจุบันกำลังศึกษาต่อ หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2554