

การตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย



นางสาว จุฑาพัคตร์ บักซี่

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

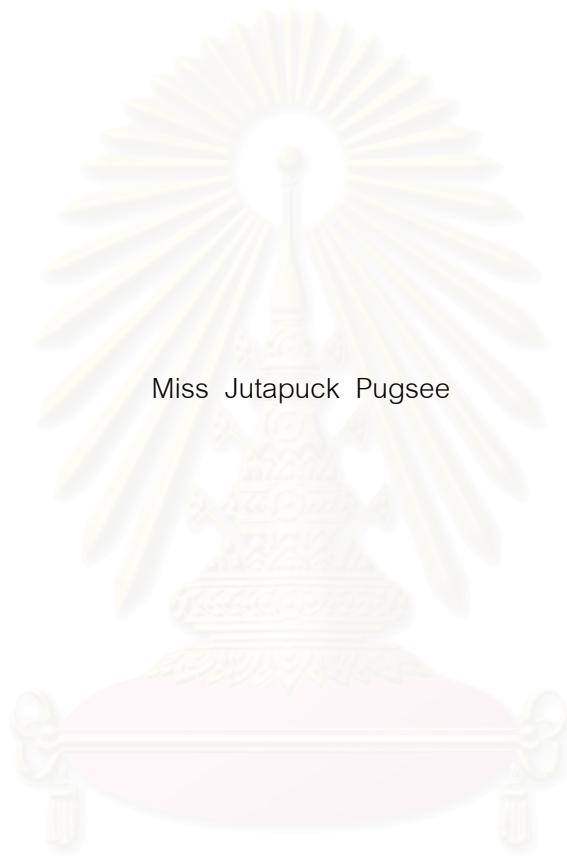
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2545

ISBN 974-17-2147-1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

INCONSISTENCY CHECKING FOR THAI SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT



Miss Jutapuck Pugsee

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2002

ISBN 974-17-2147-1

หัวข้อวิทยานิพนธ์ การตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
ภาษาไทย
โดย นางสาวจุฑาทาพัทตร์ ปักซี่
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร. วันชัย ธีรไพบูลย์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดี คณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการสอบ
(ผู้ช่วยศาสตราจารย์ สุเมธ วัชรชัยสุรพล)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.วันชัย ธีรไพบูลย์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี)

..... กรรมการ
(อาจารย์ นครทิพย์ พรหมพูล)

..... กรรมการ
(อาจารย์ เชษฐ วัฒนชัย)

จุฑาพัคตรี ปักซี่ : การตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย. (INCONSISTENCY CHECKING FOR THAI SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT) อ. ที่ปรึกษา : รองศาสตราจารย์ ดร. วันชัย รั้วโพนุลย์, 154 หน้า. ISBN 974-17-2147-1.

วิทยานิพนธ์นี้มีวัตถุประสงค์ เพื่อกำหนดแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย และพัฒนาเครื่องมือซอฟต์แวร์สนับสนุนขั้นตอนวิธีในการตรวจสอบความไม่ต้งกัน วิทยานิพนธ์นี้ใช้ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ที่มาจากแหล่งข้อมูลต่างกัน 5 แหล่ง การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้งกัน แบ่งเป็น 2 ส่วนใหญ่ โดยที่ ส่วนแรก คือ การประมวลผลภาษาธรรมชาติที่ใช้ในการเขียนเอกสาร ประกอบด้วย กระบวนการเตรียมเอกสาร การตัดคำ การกำกับหน้าที่คำ การวิเคราะห์ประโยคด้วยไวยากรณ์ เพื่อสร้างเป็นรูปแบบโครงสร้างใหม่ที่ใช้ในการตรวจสอบความไม่ต้งกัน ส่วนหลัง คือ การหาขั้นตอนวิธีในการตรวจสอบความไม่ต้งกันจากรูปแบบโครงสร้างที่ได้จากส่วนแรก

ขั้นตอนวิธีตรวจสอบความไม่ต้งกันที่ได้จากการวิจัย สามารถตรวจสอบความไม่ต้งกัน 3 ประเภทด้วยกัน คือ ประเภทที่หนึ่งความไม่ต้งกันในส่วนของข้อมูลในฐานข้อมูล ประเภทที่สองความไม่ต้งกันในความต้งกันด้านหน้าที่เดียวกัน ซึ่งประกอบด้วย ฟังก์ชันการทำงานที่ซ้ำซ้อนในความต้งกันด้านหน้าที่เดียวกัน ข้อมูลนำเข้าและข้อมูลนำออกของแต่ละฟังก์ชันการทำงานไม่ต้งกัน ข้อมูลนำเข้าและข้อมูลนำออกของฟังก์ชันการทำงานที่ทำงานต่อเนื่องกันไม่ต้งกัน ประเภทที่สามความไม่ต้งกันระหว่างความต้งกันด้านหน้าที่ที่ต่างหน้าที่กัน ซึ่งประกอบด้วย ความต้งกันด้านหน้าที่ที่ซ้ำซ้อนกัน คำที่หมายถึงกลุ่มข้อมูลเดียวกันในฐานข้อมูล แต่ประกอบด้วยข้อมูลย่อยไม่ต้งกัน ในความต้งกันด้านหน้าที่ที่ต่างกัน นอกเหนือจากนั้นยังมีการทดสอบประสิทธิภาพของขั้นตอนวิธีตรวจสอบความไม่ต้งกันและเครื่องมือซอฟต์แวร์ โดยนำมาใช้ตรวจสอบความไม่ต้งกันในตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ สรุปผลได้ว่า เครื่องมือซอฟต์แวร์สามารถตรวจสอบความไม่ต้งกันที่ระบุไว้ในงานวิจัยได้อย่างถูกต้อง

ภาควิชา วิศวกรรมคอมพิวเตอร์.....

ลายมือชื่อนิสิต

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์.....

ลายมือชื่ออาจารย์ที่ปรึกษา

ปีการศึกษา 2545.....

4370255621 : MAJOR COMPUTER SCIENCE

KEY WORD: SOFTWARE REQUIREMENTS SPECIFICATION / REQUIREMENTS ENGINEERING / NATURAL LANGUAGE PROCESSING / INCONSISTENCY

JUTAPUCK PUGSEE : INCONSISTENCY CHECKING FOR THAI SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT , THESIS ADVISOR : ASSOCIATE PROFESSOR WANCHAI RIVEPIBOON, Ph.D., 154 pp. ISBN 974-17-2147-1.

The objective of this thesis is to design methodologies for inconsistency checking in Thai software requirements specification document and implement software tool for inconsistency checking. Thai software requirements specification documents are used as examples. The design and implementation for inconsistency checking are separated into two main sections: the first section is natural language processing for textual analysis. This section consists of preprocessing, word segmentation, part-of-speech tagging and sentence analysis. The result of this part is new structures which are used for inconsistency checking. The second section is finding methodologies for inconsistency checking by using new constructed structures.

Methodologies of inconsistency checking in this thesis can examine three types of software inconsistency. First type of software inconsistency is inconsistency of data in database. Second type of software inconsistency is inconsistency of intra-functional requirements which consists of repeated functions in the same functional requirements, inconsistent inputs and outputs of each function, inconsistent inputs and outputs of the sequential functions. Third type of software inconsistency is inconsistency of inter-functional requirements which consists of repeated functional requirements, words mean the same composite data in database, but the compositions of composite data are different. In addition, methodologies of inconsistency checking and the software tool are tested by utilizing for checking inconsistency in samples of software requirements specification documents. The results show that the software tool can find all inconsistencies which are identified in this research.

DepartmentComputer Engineering..... Student's signature

Field of studyComputer Science..... Advisor's signature

Academic year2002.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลืออย่างดียิ่งจาก รองศาสตราจารย์ ดร.วันชัย ธีรไพบุลย์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้เสียสละเวลาให้คำแนะนำเกี่ยวกับแนวทางการวิจัย และข้อคิดเห็นที่มีประโยชน์มากมาย เพื่อประกอบการวิจัยของข้าพเจ้ามาโดยตลอด ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ สุเมธ วิษระชัยสุรพล ผู้ช่วยศาสตราจารย์ ดร. พรศิริ หมั่นไชยศรี อาจารย์ นครทิพย์ พร้อมพูล และอาจารย์ เชษฐ พัฒนอินทรีย์ คณะกรรมการวิทยานิพนธ์ ที่ท่านได้กรุณาให้คำแนะนำและข้อชี้แนะ ในการทำวิจัย รวมถึงการตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณคณาจารย์ท่านอื่น ๆ ทั้งในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และสถาบันอื่น ๆ ที่ให้ความรู้และคำแนะนำในการทำวิจัย

ขอขอบพระคุณอาจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์ รวมถึงบริษัทและกลุ่มผู้ผลิตซอฟต์แวร์ ที่เอื้อเฟื้อตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์สำหรับทำการวิจัย

ขอขอบคุณ NECTEC ที่พัฒนาและเผยแพร่โปรแกรมที่เกี่ยวข้องกับการประมวลผลภาษาธรรมชาติสำหรับภาษาไทย

ขอขอบคุณ ทูนกานูจนาภิเษกของบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย และทุนพัฒนาอาจารย์ของทบวงมหาวิทยาลัย ที่สนับสนุนด้านทุนการศึกษาแก่ข้าพเจ้าจนสำเร็จการศึกษา

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ (SE-LAB) และห้องปฏิบัติการอื่นๆ ที่ได้ให้คำแนะนำและกำลังใจแก่ข้าพเจ้า ตลอดเวลาที่ศึกษาในภาควิชาวิศวกรรมคอมพิวเตอร์แห่งนี้

ท้ายที่สุด ข้าพเจ้าใคร่ขอกราบขอบพระคุณบิดา มารดา พี่ ๆ และน้อง ๆ ในครอบครัวของข้าพเจ้า ที่สนับสนุนและให้กำลังใจแก่ข้าพเจ้าเสมอมาจนสำเร็จการศึกษา

จุฑาพัทธ์ ปักซี่

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ขั้นตอนในการดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 โครงสร้างวิทยานิพนธ์.....	4
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง.....	6
2.2 งานวิจัยที่เกี่ยวข้อง.....	15
3 การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	21
3.1 แนวคิดการตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	21
3.2 การออกแบบขั้นตอนวิธีการประมวลผลภาษาธรรมชาติในเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	23
3.3 การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	40

บทที่

หน้า

4 การออกแบบและพัฒนาเครื่องมือในการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนด ความต้องการซอฟต์แวร์.....	50
4.1 การวิเคราะห์และออกแบบ.....	50
4.2 การพัฒนาเครื่องมือซอฟต์แวร์.....	67
4.3 การใช้งานเบื้องต้น.....	73
5 การทดสอบและประเมินผลเครื่องมือ.....	74
5.1 การตรวจสอบความไม่ตรงกันในเอกสารตัวอย่างที่ 1.....	75
5.2 การตรวจสอบความไม่ตรงกันในเอกสารตัวอย่างที่ 2.....	93
5.3 การตรวจสอบความไม่ตรงกันในเอกสารตัวอย่างที่ 3.....	102
5.4 ผลการประเมิน.....	108
6 สรุปผลการวิจัย ปัญหา และข้อเสนอแนะ.....	109
6.1 สรุปผลการวิจัย.....	109
6.2 ปัญหาและข้อจำกัดของงานวิจัย.....	110
6.3 ข้อเสนอแนะ.....	111
รายการอ้างอิง.....	113
ภาคผนวก.....	115
ประวัติผู้เขียนวิทยานิพนธ์.....	154

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.1	สรุปความไม่ตรงกันและกฎความตรงกันที่ใช้ในการตรวจสอบ	43
ตารางที่ 3.2	ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 1	44
ตารางที่ 3.3	ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 2	45
ตารางที่ 3.4	ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 3	47
ตารางที่ 3.4	ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 4	48
ตารางที่ 5.1	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1	75
ตารางที่ 5.2	โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 1	81
ตารางที่ 5.3	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-1	83
ตารางที่ 5.4	โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 1-1	86
ตารางที่ 5.5	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-2	88
ตารางที่ 5.6	โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 1-2	91
ตารางที่ 5.7	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 2	93
ตารางที่ 5.8	โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 2	97
ตารางที่ 5.9	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 2-1	99
ตารางที่ 5.10	โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 2-1	101
ตารางที่ 5.11	เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 3	103
ตารางที่ ค-1	หน้าที่ของคำในภาษาไทยและตัวอย่าง	120
ตารางที่ จ-1	หน้าที่การทำงานของแถบเครื่องมือ	133

สารบัญรูป

	หน้า
รูปที่ 2.1 แผ่นแบบของยูสเคส.....	12
รูปที่ 3.1 แนวความคิดการออกแบบและพัฒนาขั้นตอนวิธีในการตรวจสอบความไม่ตรงกัน..	22
รูปที่ 3.2 ขั้นตอนการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	24
รูปที่ 3.3 ขั้นตอนการวิเคราะห์ประโยค.....	29
รูปที่ 3.4 ขั้นตอนการสร้างโครงสร้างใหม่.....	36
รูปที่ 3.5 ตัวอย่างโครงสร้างใหม่ที่สร้างขึ้น.....	39
รูปที่ 3.6 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 1.....	44
รูปที่ 3.7 โครงสร้างการทำงานของความต้องการด้านหน้าที่ ตัวอย่างที่ 2 การพิมพ์ใบส่งสินค้า.....	45
รูปที่ 3.8 โครงสร้างการทำงานที่ไม่ตรงกันของฐานข้อมูล.....	46
รูปที่ 3.9 ขั้นตอนการตรวจสอบความไม่ตรงกันในความต้องการด้านหน้าที่เดียวกัน.....	46
รูปที่ 3.10 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 1.....	47
รูปที่ 3.11 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 1.....	48
รูปที่ 3.12 ขั้นตอนการตรวจสอบความไม่ตรงกันในความต้องการด้านหน้าที่ต่างหน้าที่กัน.....	49
รูปที่ 4.1 แผนภาพการทำงานของเครื่องมือซอฟต์แวร์ที่เกี่ยวข้องกับผู้ใช้.....	50
รูปที่ 4.2 แผนภาพบริบท (Context Diagram) ของเครื่องมือซอฟต์แวร์.....	51
รูปที่ 4.3 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 1 51	
รูปที่ 4.4 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (มอดูลการทำงานของเอดิเตอร์ กับ กระบวนการเตรียมเอกสาร).....	52
รูปที่ 4.5 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (การตัดคำและกำกับหน้าที่คำ กับ การวิเคราะห์ประโยค).....	52
รูปที่ 4.6 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (การสร้างรูปแบบใหม่ กับ การตรวจสอบความไม่ตรงกัน).....	53
รูปที่ 4.7 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 3 (การเลือกส่วนสำคัญของประโยคและเลือกประโยค).....	53
รูปที่ 4.8 สถาปัตยกรรมภายในของเครื่องมือซอฟต์แวร์.....	54
รูปที่ 4.9 แผนภาพโครงสร้างของเครื่องมือซอฟต์แวร์.....	55
รูปที่ 4.10 แผนภาพโครงสร้างมอดูลการทำงานของเอดิเตอร์.....	55

	หน้า
รูปที่ 4.11 แผนภาพโครงสร้างมอดูลขั้นตอนการเตรียมเอกสาร.....	55
รูปที่ 4.12 แผนภาพโครงสร้างมอดูลขั้นตอนการตัดคำและการกำกับหน้าที่คำ.....	56
รูปที่ 4.13 แผนภาพโครงสร้างมอดูลขั้นตอนการวิเคราะห์ประโยค.....	56
รูปที่ 4.14 แผนภาพโครงสร้างมอดูลขั้นตอนการสร้างรูปแบบใหม่.....	56
รูปที่ 4.15 แผนภาพโครงสร้างมอดูลการตรวจสอบความไม่ต้องกัน.....	56
รูปที่ 4.16 ตัวอย่างฐานข้อมูลค่านามและค่าเฉพาะ.....	57
รูปที่ 4.17 ตัวอย่างฐานข้อมูลคำกริยา.....	58
รูปที่ 4.18 ตัวอย่างฐานข้อมูลค่านามแหล่งที่มาหรือแหล่งปลายทางของข้อมูล.....	59
รูปที่ 4.19 หน้าจอการทำงานหลักของเครื่องมือซอฟต์แวร์.....	60
รูปที่ 4.20 แถบเอดิเตอร์ ในหน้าจอการทำงานหลัก.....	61
รูปที่ 4.21 แถบรีซีล ในหน้าจอการทำงานหลัก.....	61
รูปที่ 4.22 แถบแกรมมา (ลักษณะข้อความ) ในหน้าจอการทำงานหลัก.....	62
รูปที่ 4.23 แถบแกรมมา (ลักษณะโครงสร้างรูปต้นไม้) ในหน้าจอการทำงานหลัก.....	62
รูปที่ 4.24 แถบคอนเทคในหน้าจอการทำงานหลัก.....	63
รูปที่ 4.25 แถบอินคอนซิสเทนซีในหน้าจอการทำงานหลัก.....	63
รูปที่ 4.26 แถบรีซีลในหน้าจอการเก็บข้อมูล.....	65
รูปที่ 4.27 แถบคอนเทคในหน้าจอการเก็บข้อมูล.....	65
รูปที่ 4.28 แถบเวิร์ดในหน้าจอการเก็บข้อมูล.....	66
รูปที่ 4.29 แถบดาต้า ในหน้าจอการเก็บข้อมูล.....	66
รูปที่ 4.30 หน้าจอแสดงข้อความผิดพลาด เมื่อเอกสารไม่ตรงตามมาตรฐาน.....	68
รูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน.....	74
รูปที่ 5.2 แถบเมนู Open SRS เปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่.....	76
รูปที่ 5.3 หน้าจอสำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ที่ต้องการ.....	76
รูปที่ 5.4 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ด้านหน้าที่ตัว อย่างที่ 1.....	77
รูปที่ 5.5 แถบเมนู Cut Word.....	77
รูปที่ 5.6 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ.....	78
รูปที่ 5.7 แถบเมนู Show New Structure.....	78

สารบัญรูป (ต่อ)

ฎ

	หน้า
รูปที่ 5.8 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์.....	79
รูปที่ 5.9 ออบชั่น Tree.....	79
รูปที่ 5.10 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ โครงสร้างรูปต้นไม้.....	80
รูปที่ 5.11 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	80
รูปที่ 5.12 แถบเมนู Check Inconsistency.....	82
รูปที่ 5.13 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน.....	82
รูปที่ 5.14 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัว อย่าง ที่ 1-1.....	84
รูปที่ 5.15 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	85
รูปที่ 5.16 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน.....	87
รูปที่ 5.17 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัว อย่าง ที่ 1-2.....	89
รูปที่ 5.18 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	90
รูปที่ 5.19 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน.....	92
รูปที่ 5.20 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัว อย่าง ที่ 2.....	94
รูปที่ 5.21 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ.....	95
รูปที่ 5.22 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์.....	96
รูปที่ 5.23 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	96
รูปที่ 5.24 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน.....	98
รูปที่ 5.25 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัว อย่าง ที่ 2-1.....	99
รูปที่ 5.26 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	100
รูปที่ 5.27 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน.....	102
รูปที่ 5.28 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัว อย่าง ที่ 3.....	105
รูปที่ 5.29 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ.....	105
รูปที่ 5.30 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์.....	106
รูปที่ 5.31 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	107

สารบัญรูป (ต่อ)

ฐ

	หน้า
รูปที่ จ-1 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แอปเอดิเตอร์.....	125
รูปที่ จ-2 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แอปรีซัล.....	126
รูปที่ จ-3 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แอปแกรมมา.....	126
รูปที่ จ-4 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แอปคอนเทค.....	127
รูปที่ จ-5 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – อินคอนซิสเทนซี.....	127
รูปที่ จ-6 เมนู File.....	128
รูปที่ จ-7 หน้าจอสำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ต้องการ.....	128
รูปที่ จ-8 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์.....	129
รูปที่ จ-9 หน้าจอถามความต้องการในการบันทึกก่อนการปิดเอกสาร.....	129
รูปที่ จ-10 เมนู Run.....	129
รูปที่ จ-11 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ.....	130
รูปที่ จ-12 ออบซัน Tree.....	130
รูปที่ จ-13 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ โครงสร้างรูปต้นไม้.....	131
รูปที่ จ-14 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ ลักษณะข้อความ.....	131
รูปที่ จ-15 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่.....	132
รูปที่ จ-16 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ต้งกัน.....	132
รูปที่ จ-17 หน้าจอสำหรับใส่ชื่อเพิ่มข้อมูลที่ต้องการบันทึกเอกสาร.....	133
รูปที่ จ-18 หน้าจอส่วนเก็บข้อมูล – แอปรีซัล.....	134
รูปที่ จ-19 หน้าจอส่วนเก็บข้อมูล – แอปคอนเทค.....	135
รูปที่ จ-20 หน้าจอส่วนเก็บข้อมูล – แอปเวิร์ด.....	135
รูปที่ จ-21 หน้าจอส่วนเก็บข้อมูล – แอปดาต้า.....	136
รูปที่ ฉ-1 มอดูลการตรวจสอบรูปแบบของเอกสาร.....	137
รูปที่ ฉ-2 มอดูลการแก้ไขเปลี่ยนแปลงข้อความในเอกสารโดยอัตโนมัติ.....	138
รูปที่ ฉ-3 มอดูลการรับข้อมูลของผลลัพธ์จากโปรแกรมตัดคำ.....	138
รูปที่ ฉ-4 มอดูลการแก้ไขหน้าที่คำ.....	140
รูปที่ ฉ-5 มอดูลการกำหนดขอบเขตของประโยค.....	141
รูปที่ ฉ-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค.....	142
รูปที่ ฉ-7 มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมให้โครงสร้างใหม่สมบูรณ์ขึ้น.....	148
รูปที่ ฉ-8 มอดูลการตรวจสอบความไม่ต้งกัน.....	152

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากการออกแบบและพัฒนาซอฟต์แวร์แต่ละระบบขึ้นมา นั้น มีผู้ที่เกี่ยวข้องหลายฝ่ายด้วยกัน จึงทำให้เกิดปัญหาสำคัญที่พบได้บ่อย นั่นคือ ความเข้าใจที่ไม่ตรงกันระหว่างผู้ใช้ (Users) กับผู้พัฒนาซอฟต์แวร์ ดังนั้นจึงมีการสร้างเอกสารที่ทำให้ผู้พัฒนาซอฟต์แวร์สามารถพัฒนาซอฟต์แวร์ได้ตรงตามความต้องการของผู้ใช้ เอกสารที่สำคัญเป็นอันดับแรกในขั้นตอนการพัฒนาซอฟต์แวร์นั่นก็คือ “เอกสารข้อกำหนดความต้องการซอฟต์แวร์” (Software Requirements Specification Document – SRS Document) [1] ซึ่งอธิบายหน้าที่และการทำงานของซอฟต์แวร์ ถึงแม้ว่าจะมีการสร้างมาตรฐานของเอกสารนี้หลายมาตรฐานด้วยกัน เช่น IEEE Std 830-1993, ESA PSS-05-0, NASA-DID-P200, DoD-STD-498 แต่สำหรับเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นภาษาไทยซึ่งใช้กันในประเทศยังไม่มีมาตรฐานที่แน่นอน มีเพียงงานวิจัย [2] ที่ทำการหามาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เหมาะสมที่สุด ในการนำมาประยุกต์ใช้ภายในประเทศ โดยมาตรฐานนี้จะกำหนดถึงข้อมูลต่าง ๆ เกี่ยวกับซอฟต์แวร์ที่ควรเขียนในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ แต่ในส่วนของเนื้อหา ภาษาที่ใช้เขียน รวมถึงคำศัพท์ที่ใช้ นั้น ยังขึ้นกับความสามารถและความเข้าใจของผู้เขียน ที่จะทำให้ออกสารนั้นทำความเข้าใจได้ง่าย มีความถูกต้อง (Correctness) และความสมบูรณ์ (Completeness) เพราะภาษาที่ใช้เขียนยังคงเป็นภาษาธรรมชาติ (Natural Language) ซึ่งอาจมีความกำกวม (Ambiguity) และความคลุมเครือ (Vagueness) เกิดขึ้นได้ [3]

ผลที่เกิดขึ้นถ้าข้อกำหนดความต้องการซอฟต์แวร์ (SRS) ไม่มีความถูกต้องและมีความสมบูรณ์ไม่เพียงพอ ทำให้เกิดปัญหาในการทำความเข้าใจเอกสาร เช่น ไม่เที่ยงตรง (Imprecision) มีส่วนที่ซ้ำซ้อน (Redundancy) กำกวม (Ambiguity) ไม่ตรงกัน (Inconsistency) และไม่สมบูรณ์ (Incompleteness) ซึ่งจะทำให้การพัฒนาซอฟต์แวร์ไม่สามารถประสบความสำเร็จได้ [1] [4] การแก้ปัญหาข้างต้นทางหนึ่ง คือการเขียนข้อกำหนดความต้องการซอฟต์แวร์ด้วยภาษาเชิงรูปนัย (Formal Language) [5] เช่น ภาษาเซต (Z) วิดีเอ็ม (Vienna Development Method-VDM) แต่ก็ทำให้เกิดปัญหาตามมา ก็คือข้อกำหนดความต้องการซอฟต์แวร์ไม่สามารถทำความเข้าใจได้ง่าย เนื่องจากภาษาเชิงรูปนัยมีวากยสัมพันธ์ (Syntax) เฉพาะของแต่ละภาษา ต้องมีการเรียนรู้ศึกษารูปแบบวากยสัมพันธ์ จึงจะสามารถทำความเข้าใจได้ การแก้ปัญหาอีกแนวทางหนึ่งก็คือ การตรวจสอบความถูกต้องและความสมบูรณ์จากการจำลองแบบเชิงแนวคิด (Modeling) [6] เช่น

แผนภาพการไหลของข้อมูล (Data Flow Diagram - DFD) แผนภาพเอนทิตีและความสัมพันธ์ (Entity-relationship Diagram - ERD) วิธีนี้จะมีความยุ่งยากที่ต้องแปลงข้อความในข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นภาษารวมชาติ ให้ออกมาในรูปของแบบจำลองการทำงานและแบบจำลองข้อมูลเชิงแนวคิดก่อนจึงจะทำการตรวจสอบได้

ผู้วิจัยจึงมีแนวความคิดที่จะทำการตรวจสอบหาส่วนที่ไม่ต้องกัน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ซึ่งไม่ได้อยู่ในรูปภาษาเชิงรูปนัยและแบบจำลองเชิงแนวคิด ปัญหาสำคัญที่ผู้วิจัยให้ความสนใจคือ ความไม่ต้องกัน เนื่องจากปัญหานี้ทำให้ซอฟต์แวร์ที่พัฒนาขึ้นไม่สามารถทำงานได้อย่างถูกต้อง และเป็นปัญหาที่สามารถตรวจสอบได้อย่างชัดเจนแม้จะอยู่ในรูปของภาษารวมชาติ ต่างจากความกำกวม ความซ้ำซ้อน และความไม่สมบูรณ์ที่ตรวจสอบได้ยาก เนื่องมาจากคุณสมบัติของภาษารวมชาติที่บางครั้งมีความกำกวมในตัวของมันเอง มีการใช้คำซ้ำซ้อน เพื่อขยายความทำให้ผู้อ่านสามารถเข้าใจได้อย่างถูกต้องมากขึ้น มีการละคำบางคำออกไป เพราะผู้อ่านและผู้เขียนมีความเข้าใจตรงกันอยู่แล้ว สิ่งเหล่านี้ทำให้ไม่สามารถหากฎหรือหลักการเพื่อมาตัดสินได้ว่าเกิดความกำกวม ซ้ำซ้อน หรือไม่สมบูรณ์ขึ้นจริง งานวิจัยนี้มีการประยุกต์ใช้การประมวลผลภาษารวมชาติ (Natural Language Processing - NLP) การตัดคำ (Word Segmentation) การกำกับหน้าที่คำ (Part-of-speech Tagging) เพื่อทำการแปลงจากภาษารวมชาติ ให้เป็นรูปแบบใหม่ที่สามารถใช้เทคนิคในการตรวจสอบความไม่ต้องกัน เพื่อให้ข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยมีความถูกต้อง สามารถทำความเข้าใจและนำไปวิเคราะห์ต่อไปได้ง่าย ซึ่งจะทำให้เกิดการพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพ มีความถูกต้องและตรงต่อความต้องการของผู้ใช้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้องกัน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย และพัฒนาเครื่องมือซอฟต์แวร์สนับสนุนขั้นตอนวิธีในการตรวจสอบความไม่ต้องกัน

1.3 ขอบเขตของการวิจัย

1. ศึกษาตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ที่มาจากแหล่งข้อมูลหรือหน่วยงานที่ต่างกันอย่างน้อย 5 แหล่ง แล้วนำมาประยุกต์ให้ตรงตามมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เลือกใช้ รวมถึงมีการแบ่งกลุ่ม

เอกสารเป็น 2 กลุ่มใหญ่ คือ เอกสารที่ถูกต้อง (ไม่มีความไม่ตรงกันในเอกสาร) และ เอกสารที่ไม่ถูกต้อง (มีความไม่ตรงกันในเอกสาร)

2. ตรวจสอบความไม่ตรงกันส่วนความต้องการด้านหน้าที่ ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยการประมวลผลภาษาธรรมชาติให้อยู่ในรูปโครงสร้างใหม่ ซึ่งประกอบด้วย ยูสเคส (Use Case) แหล่งที่มา (Source) ข้อมูลนำเข้า (Input) การทำงาน (Action) ข้อมูลนำออก (Output) และ แหล่งปลายทาง (Destination) ซึ่งสามารถใช้เทคนิคในการตรวจสอบความไม่ตรงกันได้
3. พัฒนาเครื่องมือซอฟต์แวร์ (Tool) เพื่อสนับสนุนการประมวลผลภาษาธรรมชาติ รวมถึงการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ให้สามารถทำความเข้าใจได้ง่ายและมีความถูกต้อง
4. ความไม่ตรงกันส่วนความต้องการด้านหน้าที่ ที่สามารถตรวจสอบจากโครงสร้างใหม่ที่สร้างขึ้น ตามขอบเขตการวิจัยข้อที่ 2 ได้แก่
 - 4.1 ความไม่ตรงกันของข้อมูลในฐานข้อมูล (Database)
 - 4.2 ความไม่ตรงกันของฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกในความต้องการด้านหน้าที่เดียวกัน (Intra-functional Requirements)
 - 4.3 ความไม่ตรงกันของฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกระหว่างความต้องการด้านหน้าที่ที่ต่างหน้าที่กัน (Inter-functional Requirements)
5. รูปแบบโครงสร้างประโยคภาษาธรรมชาติ ที่ใช้ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่จะนำมาทำการตรวจสอบความไม่ตรงกันในงานวิจัยนี้ ต้องใช้รูปแบบโครงสร้างประโยคที่ถูกต้องตามหลักไวยากรณ์ภาษาไทย มีรายละเอียดส่วนที่สำคัญของความต้องการด้านหน้าที่ครบถ้วน ชัดเจน ไม่มีความกำกวม สำหรับในประโยคความรวมหรือประโยคความซ้อน ต้องมีรูปแบบประโยคที่ไม่ซับซ้อน มีการใช้ตัวเชื่อมประโยค (คำสันธาน) ไม่เกินกว่า 3 ตัวเชื่อมในประโยค

1.4 ขั้นตอนในการดำเนินการวิจัย

1. ศึกษาลักษณะมาตรฐานของข้อกำหนดความต้องการซอฟต์แวร์ที่เลือกใช้
2. ศึกษาตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์
3. ศึกษาวิธีการตัดคำและการกำกับหน้าที่คำที่เลือกมาใช้
4. ศึกษาและวิเคราะห์ลักษณะไวยากรณ์ภาษาไทย
5. ออกแบบขั้นตอนวิธีในการประมวลผลภาษาธรรมชาติในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยประยุกต์ใช้ความสามารถของโปรแกรมตัดคำและกำกับหน้าที่

คำ เพื่อสร้างรูปแบบโครงสร้างใหม่อธิบายส่วนความต้องการด้านหน้าที่ สำหรับนำไปตรวจสอบความไม่ต้อกัน

6. ออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้อกัน ส่วนความต้องการด้านหน้าที่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
7. พัฒนาเครื่องมือซอฟต์แวร์เพื่อสนับสนุนการประมวลผลภาษาธรรมชาติ และเทคนิคในการตรวจสอบความไม่ต้อกัน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์
8. ทดสอบและปรับปรุงคุณภาพของขั้นตอนวิธี ในการตรวจสอบความไม่ต้อกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ และเครื่องมือซอฟต์แวร์ให้ปฏิบัติงานได้อย่างมีประสิทธิภาพ
9. สรุปผลการวิจัย และจัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1 ได้ขั้นตอนวิธีในการตรวจสอบความไม่ต้อกัน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ทำให้ข้อกำหนดความต้องการซอฟต์แวร์เข้าใจได้ง่าย มีความถูกต้อง และเหมาะสำหรับการนำไปวิเคราะห์ต่อไป
- 2 ผู้พัฒนาซอฟต์แวร์มีแนวทางในการสร้างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ที่มีคุณภาพ สามารถทำความเข้าใจได้ง่าย และมีความถูกต้อง

1.6 โครงสร้างวิทยานิพนธ์

บทที่ 1 กล่าวถึงความเป็นมา วัตถุประสงค์ ขอบเขตการวิจัย รวมถึงขั้นตอนการวิจัย และประโยชน์ที่คาดว่าจะได้รับจากการวิจัยนี้

บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในการทำวิทยานิพนธ์

บทที่ 3 กล่าวถึงการออกแบบขั้นตอนวิธีการประมวลผลภาษาธรรมชาติและการออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้อกัน ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

บทที่ 4 กล่าวถึงรายละเอียดต่างๆ ในการพัฒนาเครื่องมือซอฟต์แวร์เพื่อตรวจสอบความไม่ต้อกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

บทที่ 5 กล่าวถึงการทดสอบ ประเมินผลเครื่องมือ และผลการทำงานจากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์

บทที่ 6 กล่าวถึงการสรุปผลของงานวิจัย ปัญหาและข้อจำกัดของงานวิจัย รวมถึงข้อเสนอแนะในการพัฒนาขั้นตอนวิธีในการประมวลผลภาษาธรรมชาติ และการตรวจสอบความไม่ต้อกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ให้มีประสิทธิภาพมากขึ้นต่อไป

ภาคผนวก ประกอบด้วย 6 ภาคผนวก คือ

ภาคผนวก ก. แผ่นแบบเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย

ภาคผนวก ข. ตัวอย่างยูสเคส

ภาคผนวก ค. หน้าที่คำในภาษาไทยและตัวอย่างคำจากคลังข้อความอรรถิคิด

ภาคผนวก ง. ไวยากรณ์ภาษาไทยและรูปแบบเอกสารส่วนความต้องการด้านหน้าที่

ภาคผนวก จ. คู่มือประกอบการใช้เครื่องมือซอฟต์แวร์ (User Manual) ตรวจสอบความ

ไม่ต้งกันในเอกสาร

ภาคผนวก ฉ. ตัวอย่างชุดคำสั่ง (Source Code) ในเครื่องมือซอฟต์แวร์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในงานวิทยานิพนธ์นี้มีทฤษฎีต่าง ๆ ที่เกี่ยวข้องดังนี้

2.1.1. ข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification - SRS) [11] [12]

ข้อกำหนดความต้องการซอฟต์แวร์เป็นรูปแบบเบื้องต้นของเอกสารรูปนัย (Formal Document) ที่ใช้ติดต่อสื่อสารระหว่างลูกค้า (Customer) กับผู้ผลิต (Supplier) ข้อกำหนดความต้องการซอฟต์แวร์มีความสำคัญ เนื่องจากในข้อกำหนดจะบอกถึงจุดประสงค์ (Purpose) และเนื้อหา (Context) ที่สำคัญสำหรับอธิบายข้อมูลเกี่ยวกับซอฟต์แวร์

ประเภทของความต้องการแบ่งเป็นหลายประเภท ได้แก่ ความต้องการด้านหน้าที่ (Functional Requirements) ความต้องการด้านประสิทธิภาพ (Performance Requirements) ความต้องการด้านการต่อประสาน (Interface Requirements) ความต้องการด้านการดำเนินการ (Operational Requirements) ความต้องการด้านทรัพยากร (Resource Requirements) ความต้องการด้านการทวนสอบ (Verification Requirements) ความต้องการด้านการยอมรับ (Acceptance Requirements) ความต้องการด้านเอกสาร (Documentation Requirements) ความต้องการด้านคุณภาพของซอฟต์แวร์ (Quality Requirements) ความต้องการด้านความปลอดภัย (Safety Requirements) ความต้องการด้านความเชื่อถือได้ (Reliability Requirements) และ ความต้องการด้านการดูแลรักษา (Maintainability Requirements)

คุณสมบัติ (Characteristics) ของข้อกำหนดความต้องการซอฟต์แวร์ที่ดี คือ

ถูกต้อง (Correct) หมายถึง ซอฟต์แวร์ทำงานได้ถูกต้องตรงตามความต้องการทุกความต้องการ แต่ปัจจุบันยังไม่มีเครื่องมือซอฟต์แวร์ หรือขั้นตอนวิธีการที่ทำให้แน่ใจในความถูกต้องของซอฟต์แวร์ได้ร้อยเปอร์เซ็นต์ ดังนั้นจึงมีทางเลือกอีกวิธีหนึ่งคือ ให้ลูกค้าหรือผู้ใช้เป็นผู้ตัดสินใจ โดยถ้าซอฟต์แวร์นั้นตรงกับความต้องการของลูกค้าหรือผู้ใช้ ก็ถือได้ว่าซอฟต์แวร์นั้นถูกต้อง

ไม่กำกวม (Unambiguous) หมายถึง แต่ละความต้องการสามารถแปลความหมายได้เพียงหนึ่งความหมายเท่านั้น ถ้าไม่สามารถหลีกเลี่ยงความกำกวมได้ ควรมีการแยกแยะข้อความเพื่ออธิบายเนื้อหาที่ถูกต้องของความต้องการนั้น

สมบูรณ์ (Complete) หมายถึง ต้องมีการอธิบายความต้องการหลักทั้งหมดที่เกี่ยวกับหน้าที่การทำงาน (Function) ประสิทธิภาพ (Performance) ข้อจำกัดการออกแบบ (Design Constraints) และการต่อประสานภายนอก (External Interface) และมีรายละเอียดดังต่อไปนี้

- 1) มีการจำกัดความการตอบสนอง (Response) ของซอฟต์แวร์ ทุกสภาวะ ทุกข้อมูลนำเข้าอย่างครบถ้วนและสมเหตุสมผล ทั้งข้อมูลนำเข้าที่สมเหตุสมผล (Valid) และไม่สมเหตุสมผล (Invalid)
- 2) มีการทำให้รายละเอียดทุก ๆ ส่วนในข้อกำหนดตรงกับมาตรฐานของซอฟต์แวร์
- 3) มีการกำกับตัวอย่างครบถ้วน (Full Labeling) และการอ้างอิง (Reference) ของตารางและรายการอ้างอิง
- 4) มีการกำหนดทุกอย่างที่สมบูรณ์ ถ้าบางส่วนยังไม่ได้จำกัดความ ถือว่าข้อกำหนดความต้องการนั้น ยังไม่สมบูรณ์

ต้องกัน (Consistent) หมายถึง คำอธิบายหลายคำอธิบาย (Multiple Descriptors) คือ คำสองคำหรือมากกว่า ที่อธิบายสิ่งเดียวกัน ต้องมีความต้องกัน โดยความต้องกันในที่นี้ หมายถึง ความต้องกันภายใน (Internal Consistency) “ข้อกำหนดความต้องการซอฟต์แวร์มีความต้องกัน ก็ต่อเมื่อ ไม่มีส่วนย่อย (Subset) ของความต้องการใดที่อธิบายแล้วเกิดความขัดแย้งกัน (Conflict)” ชนิดของความขัดแย้งแบ่งเป็น 3 กลุ่ม คือ

กลุ่มที่ 1 คุณสมบัติเฉพาะของวัตถุในโลกความเป็นจริง (Real-world Objects) ได้แก่ การอธิบายความต้องการทางกายภาพ (Physical Requirements) มีการอธิบายวัตถุไม่ตรงกัน เช่น “ความต้องการหนึ่งให้รูปแบบรายงานเป็นตารางแต่อีกความต้องการหนึ่งให้เป็นข้อความ” ความต้องการหนึ่งกำหนดให้ไฟเป็นสีฟ้า ในขณะที่อีกความต้องการหนึ่งกำหนดให้ไฟเป็นสีเขียว

กลุ่มที่ 2 ความขัดแย้งเชิงตรรกะ (Logical Conflict) หรือความขัดแย้งที่ขึ้นกับเวลา (Temporal Conflict) ระหว่างสองการทำงาน ได้แก่ การอธิบายความต้องการด้านหน้าที่ มีการอธิบายคุณสมบัติของฟังก์ชันขัดแย้งกัน เช่น “ความต้องการหนึ่งกำหนดให้การทำงาน X เกิดหลังการทำงาน A และ (AND) B กับอีกความต้องการหนึ่งกำหนดให้การทำงาน X เกิดหลังการทำงาน A หรือ (OR) B” ความต้องการหนึ่งกำหนดให้โปรแกรมรับข้อมูลนำเข้าสองตัว กับอีกความต้องการหนึ่งให้รับข้อมูลนำเข้าจากทั้งสองตัว หรือจากตัวใดตัวหนึ่งก็ได้”

กลุ่มที่ 3 ความต้องการสองความต้องการหรือมากกว่าที่อธิบายสิ่งเดียวกัน แต่ใช้คำหรือข้อความต่างกัน

ความสำคัญ และ/หรือ เสถียรภาพเรียงลำดับ (Ranked for Importance and/or Stability) หมายถึง แต่ละความต้องการมีตัวแยกแยะความสำคัญหรือเสถียรภาพของความต้องการ เนื่องจากความต้องการในซอฟต์แวร์ไม่ได้มีความสำคัญเท่ากันหมด บางความต้องการมีความจำเป็น ซอฟต์แวร์ต้องสามารถทำงานในส่วนนั้นได้ ในขณะที่บางความต้องการเป็นสิ่งที่ต้องการเพิ่มเติมเท่านั้น

ทวนสอบได้ (Verifiable) หมายถึง ทุกความต้องการในข้อกำหนดความต้องการซอฟต์แวร์ต้องสามารถทวนสอบได้ การทวนสอบได้มีขั้นตอนวิธีในการหาค่าประสิทธิภาพ (Finite Cost-effective Method) ซึ่งสามารถทำได้โดยคนหรือเครื่อง เพื่อตรวจสอบว่าซอฟต์แวร์ที่ได้ตรงกับความต้องการ ตัวอย่างความต้องการที่ไม่สามารถทวนสอบได้ คือประกอบด้วยคำ “ดี (Well)” “ดี (Good)” “บ่อย ๆ (Usually)” เพราะไม่สามารถจำกัดความความหมายของคำเหล่านั้นได้

ดัดแปรได้ (Modifiable) หมายถึง การเปลี่ยนแปลงความต้องการที่เกิดขึ้น เพื่อให้เกิดความสมบูรณ์ ความตึงกัน และให้ง่ายต่อการทำความเข้าใจ ไม่มีผลกระทบต่อโครงสร้าง (Structure) หรือรูปแบบ (Style) ของข้อกำหนดความต้องการซอฟต์แวร์ ลักษณะของคุณสมบัติดัดแปรได้ คือ

- 1) มีโครงสร้างที่เชื่อมโยงกัน เช่น สารบัญ ดัชนี การอ้างอิง
- 2) ไม่ซ้ำซ้อน (ความต้องการเดียวกันไม่ปรากฏซ้ำในข้อกำหนดความต้องการซอฟต์แวร์)
- 3) แสดงความต้องการแต่ละความต้องการแยกกัน ไม่มีการรวมความต้องการหลายอย่างไว้ในความต้องการเดียวกัน

ตามรอยได้ (Traceable) หมายถึง ความสามารถตามรอยได้ว่าความต้องการ ณ ปัจจุบัน มีจุดกำเนิด (Origin) อยู่ที่ไหน มีการอ้างอิงมาจากไหน เมื่อมีการปรับปรุงเปลี่ยนแปลง สามารถรู้ว่าใครทำ ทำเมื่อไร และสถานะไหนนำไปสู่การปรับปรุงเปลี่ยนแปลง การตามรอยได้แบ่งเป็น 2 ชนิด คือ

- 1) การตามรอยถอยหลัง (Backward Traceability) คือแต่ละความต้องการมีการอ้างอิงถึงแหล่งที่มา
- 2) การตามรอยไปข้างหน้า (Forward Traceability) คือแต่ละความต้องการมีการกำหนดชื่อหรือหมายเลขอ้างอิงที่มีเพียงหนึ่งเดียวเท่านั้น

2.1.2 วิศวกรรมความต้องการ (Requirements Engineering) [5]

เป็นกระบวนการเกี่ยวกับการพัฒนาความต้องการ (Requirements) ให้มีความถูกต้องและความสมบูรณ์ ประกอบด้วย การเก็บรวบรวมข้อมูล (Eliciting) การทำความเข้าใจ (Understanding) การวิเคราะห์ (Analysis) และการสร้างเอกสารความต้องการ (Documentation)

ปัญหาส่วนใหญ่ในเรื่องความต้องการ คือ ความต้องการที่ได้ไม่ใช่ความต้องการที่แท้จริงของลูกค้าต่อระบบ ความต้องการมีความไม่ตรงกันหรือไม่สมบูรณ์ มีการเสียค่าใช้จ่ายสูงถ้าเกิดการเปลี่ยนแปลงความต้องการหลังจากมีการตกลงกันแล้ว วิศวกรรมความต้องการทำให้เกิดการค้นพบ การสร้างเอกสาร และการดูแลกลุ่มของความต้องการสำหรับระบบคอมพิวเตอร์

กระบวนการวิศวกรรมความต้องการ (Requirements Engineering Process)

- 1) การเก็บรวบรวมข้อมูล (Requirements Elicitation) มีได้หลายทาง ได้แก่ การสัมภาษณ์ (Interviewing) การพิจารณาสิ่งแวดล้อม (Scenarios) การสร้างต้นแบบ (Prototyping) และ การสังเกต (Participant Observation) ซึ่งมีขั้นตอนการทำงานคือ การทำความเข้าใจโดเมนโปรแกรมประยุกต์ (Application Domain Understanding) การทำความเข้าใจปัญหา (Problem Understanding) การทำความเข้าใจขั้นตอนการทำงาน (Business Understanding) การเข้าใจความต้องการและข้อจำกัดของระบบ รวมถึงความต้องการของผู้ที่ระบบมีผลกระทบด้วย
- 2) การวิเคราะห์ (Requirements Analysis) มีการใช้วิธีการวิเคราะห์ตามรายการ (Checklist) และวิเคราะห์ตามเมตริกซ์ในเชิงโต้ตอบ (Interaction Metrics)
- 3) การเจรจาตกลง (Requirements Negotiation) คือ การอภิปรายความขัดแย้ง (Discussing Conflicts) การแก้ปัญหาคความขัดแย้ง (Resolve Requirements Conflicts) และ การกำจัดความต้องการที่ซ้อนทับกัน (Remove Requirements Overlaps)
- 4) การสร้างเอกสาร (Requirements Documentation)
- 5) การตรวจสอบความสมเหตุสมผล (Requirements Validation) ตรวจสอบความตรงกัน ความสมบูรณ์ และความแม่นยำ (Accuracy)

สรุปข้อมูลนำเข้าและผลลัพธ์ที่ได้จากกระบวนการวิศวกรรมความต้องการ

ข้อมูลนำเข้า คือ เอกสารความต้องการ ความรู้ขององค์กร (Organizational Knowledge) มาตรฐานขององค์กร (Organizational Standards)

ผลลัพธ์ที่ได้ คือ เอกสารข้อกำหนดความต้องการที่มีความถูกต้องและสมบูรณ์

เทคนิคทางวิศวกรรมความต้องการ (Requirement Engineering Techniques)

- 1) การจำลองแบบการไหลของข้อมูล (Data-flow Modeling)
- 2) การจำลองแบบข้อมูลทางความหมาย (Semantic Data Models) หรือแสดงในรูปแบบของเอนทิตีและความสัมพันธ์ (Entity-relationship Model)
- 3) รูปแบบเชิงวัตถุ (Object-oriented Approaches) เช่น การจำลองแบบเชิงวัตถุ (Object Modeling) ได้แก่ วัตถุ (Object) คลาส (Class) เมธอด (Method) สาร (Message) การห่อหุ้มข้อมูล (Encapsulation) การถ่ายทอด (Inheritance) และยูเอ็มแอล (UML-Unified Modeling Language) ที่แสดงแผนภาพคลาส (Class Diagram) แผนภาพแสดงการเปลี่ยนสถานะ (State Transition Diagram)
- 4) วิธีการรูปนัย (Formal Method) ประกอบด้วย ไวยากรณ์ ความหมาย และความสัมพันธ์ มี 2 ประเภท คือ แบบจำลองตามสัญกรณ์ (Model-based notations) เช่น ภาษาเซต วิดีโอเอ็ม และกระบวนการทางพีชคณิต (Process Algebras) เช่น ซีเอสพี (Communicating Sequential Processes-CSP) ซีซีเอส (CCS) โลโตส (LOTOS) วิธีการรูปนัยนั้นมีข้อดีที่ ไวยากรณ์ ความหมาย และความสัมพันธ์ อยู่ในรูปไวยากรณ์ทางคณิตศาสตร์สามารถทวนสอบความถูกต้อง ความไม่สมบูรณ์ ความไม่ตรงกันได้ง่าย และไม่มีความกำกวมแต่ก็มีข้อเสีย คือ จุดสนใจ (Focus) เบื้องต้นสำหรับฟังก์ชันและข้อมูล รวมถึงเกณฑ์พฤติกรรม (Behavioral Aspects) ไม่สามารถแสดงได้ ไม่ได้บอกถึงตำแหน่งปัญหาของการสร้างความต้องการเริ่มต้นที่เกิดขึ้น และยังขาดแคลนเครื่องมือในการสร้างข้อกำหนดได้โดยอัตโนมัติ

การแทนความต้องการ (Requirement Representations)

จากเทคนิคทางวิศวกรรมความต้องการ สามารถเขียนแทนความต้องการ ได้เป็น 3 รูปแบบ

ดังนี้

- 1) อรูปนัย (Informal) ได้แก่ ความต้องการที่เขียนอยู่ในรูปภาษาธรรมชาติ
- 2) กึ่งรูปนัย (Semi-formal) ได้แก่ ความต้องการที่เขียนอยู่ในรูปแบบจำลองต่าง ๆ เช่น แผนภาพการไหลของข้อมูล แผนภาพเอนทิตีและความสัมพันธ์ และยูเอ็มแอล
- 3) รูปนัย (Formal) ได้แก่ ความต้องการที่เขียนอยู่ในรูปภาษาเชิงรูปนัย เช่น เซต วิดีโอเอ็ม

2.1.3 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)

เป็นความพยายามที่จะทำให้คอมพิวเตอร์สามารถเข้าใจความหมายของภาษามนุษย์ได้โดยอัตโนมัติ

ซึ่งปัญหาของการเข้าใจภาษามนุษย์เกิดจากธรรมชาติของการใช้ภาษา เช่น การละคำที่เข้าใจกันดีอยู่แล้ว ข้อความเดียวกันแต่มีความหมายได้หลายอย่างขึ้นกับคำที่อยู่รอบข้าง และภาษาธรรมชาติมีวิวัฒนาการอยู่เสมอ

ขั้นตอนในการทำความเข้าใจภาษาธรรมชาติ แบ่งได้เป็น 5 ขั้นตอน คือ

- 1) การวิเคราะห์ระดับหน่วยคำ (Morphological Analysis) การแบ่งคำ การวิเคราะห์หน่วยคำแต่ละคำ
- 2) การวิเคราะห์ระดับวากยสัมพันธ์ (Syntactic Analysis) ลำดับของคำในประโยคที่สัมพันธ์กับคำอื่น แปลงเป็นโครงสร้างประโยคที่ตรงกับไวยากรณ์
- 3) การวิเคราะห์ระดับอรรถศาสตร์ (Semantic Analysis) นำโครงสร้างประโยคที่ได้ แปลความหมายตามไวยากรณ์ ซึ่งมักจะใช้ฐานความรู้ (Knowledge Base) เข้ามาช่วยในการทำงาน
- 4) การรวมปริจเฉท (Discourse Integration) พิจารณาความหมายของประโยคโดยอ้างอิงกับประโยคข้างเคียง ความต่อเนื่องเกี่ยวข้อกันของข้อความ
- 5) การวิเคราะห์ระดับวัจนปฏิบัติศาสตร์ (Pragmatic Analysis) การแปลความหมายประโยคอีกครั้งหนึ่ง (Reinterpret) เพื่อให้ได้ความหมายที่แท้จริง

2.1.4 ยูนีค และวากยสัมพันธ์ของแผนภาพการไหลของข้อมูล (Syntax of DFD)

[13]

มีหลายแนวทางในการสร้างแผนภาพการไหลของข้อมูล เพื่อแสดงถึงกระบวนการการทำงาน (Process) ของซอฟต์แวร์ แนวทางหนึ่งคือ การสร้างยูนีค ซึ่งเป็นรูปแบบอย่างง่ายของแผนภาพการไหลของข้อมูล ดังนั้นผู้ใช้สามารถทำความเข้าใจได้ง่ายกว่า มีการนำข้อมูลจากยูนีคไปสร้างเป็นแผนภาพการไหลของข้อมูล ในส่วนคำอธิบายกระบวนการ ที่เขียนในยูนีคจะเขียนในรูปภาษาธรรมชาติ แต่มีการจัดรูปแบบ โดยแยกส่วนเป็น แหล่งที่มา ข้อมูลนำเข้า ข้อมูลนำออก และแหล่งปลายทาง แสดงดังรูปที่ 2.1 (ตัวอย่างของยูนีคแสดงในภาคผนวก ข.)

ผู้วิจัยใช้แนวคิดนี้ เป็นแนวทางเบื้องต้นในการสร้างรูปแบบสำหรับการตรวจสอบความไม่
 ต้องกันในส่วนความต้องการด้านหน้าที่

ID number :			
Use case name :			
Short description :			
Trigger :			
Type : External		Temporal	
Major Inputs		Major Outputs	
Description	Source	Description	Destination
.....
.....
.....
Major Steps Performed		Information for Steps	
1.	
2.	

รูปที่ 2.1 แผ่นแบบของยูสเคส

วากยสัมพันธ์ของแผนภาพการไหลของข้อมูล ได้แก่

วากยสัมพันธ์ของกระบวนการ ประกอบด้วย

- 1) แต่ละกระบวนการมีหนึ่งชื่อ และหนึ่งคำอธิบาย
- 2) มีข้อมูลนำเข้าอย่างน้อยหนึ่งข้อมูล
- 3) มีข้อมูลนำออกอย่างน้อยหนึ่งข้อมูล

วากยสัมพันธ์ของการไหลของข้อมูล ประกอบด้วย

- 1) แต่ละการไหลของข้อมูลมีหนึ่งชื่อ และหนึ่งคำอธิบาย
- 2) มีการไหลอย่างน้อยหนึ่งเส้นทางที่ต่อเชื่อมกับกระบวนการ

การไหลของข้อมูล มีหลักเกณฑ์สถานะ ดังต่อไปนี้

- 1) ข้อมูลต้องอยู่กับที่ จนกระทั่งมีกระบวนการทำให้ข้อมูลเคลื่อนที่
- 2) ข้อมูลห้ามมีการสูญหาย ระหว่างการเคลื่อนที่ผ่านกระบวนการ

- 3) กระบวนการไม่สามารถสร้างข้อมูลขึ้นมาเองได้ โดยไม่มีข้อมูลนำเข้า
- 4) ชื่อกระบวนการต่างกระบวนการห้ามซ้ำกัน ชื่อกระบวนการเดียวกันต้องใช้ชื่อเดียวกันในแต่ละที่ปรากฏ
- 5) ชื่อข้อมูลต่างข้อมูลห้ามซ้ำกัน ข้อมูลเดียวกันต้องใช้ชื่อเดียวกัน

ผู้วิจัยใช้ความสัมพันธ์นี้เป็นแนวทางในการตรวจสอบความไม่ต่องกันในส่วนความต้องการด้านหน้าที่

2.1.5 ความไม่ต่องกันในการพัฒนาซอฟต์แวร์ (Inconsistency in Software Development) [14]

ความไม่ต่องกัน หมายถึง สถานการณ์ (Situation) ที่การอธิบาย (Description) สองการอธิบายใด ๆ ไม่ตรงตามความสัมพันธ์ระหว่างกัน (Relationship) ซึ่งความสัมพันธ์ระหว่างกันนั้นเปรียบได้กับกฎความต่องกัน (Consistency Rule) ซึ่งสามารถตรวจสอบได้ ตัวอย่างกฎความต่องกัน เช่น ใน DFD ถ้ากระบวนการหนึ่ง แยกได้เป็นกระบวนการย่อยหลายกระบวนการ ข้อมูลนำเข้าของกระบวนการที่เป็นลูกจะเป็นข้อมูลนำเข้าเดียวกันกับกระบวนการใหญ่กระบวนการเดิม

กฎความต่องกันสามารถหาได้จากสิ่งต่อไปนี้ คือ

- 1) การจำกัดความของสัญกรณ์ (Definition of Notations)
- 2) วิธีการในการพัฒนา (Development Methods) เช่น วิธีการในการหาสัญกรณ์ การใช้สัญกรณ์
- 3) แบบจำลองกระบวนการในการพัฒนา (Development Process Model) เช่น ข้อจำกัดการกำหนดขั้นตอนการพัฒนา ข้อจำกัดการนำเข้า ข้อจำกัดการส่งออก ในแต่ละขั้นตอน
- 4) ความบังเอิญเฉพาะ (Local Contingencies) บางครั้งความต่องกันที่เฉพาะเจาะจง (Specific Consistency) เกิดขึ้นระหว่างการอธิบายที่เฉพาะเจาะจง (Specific Description)
- 5) โดเมนของแอปพลิเคชัน (Application Domain) ในแต่ละโดเมนมีข้อจำกัดเฉพาะของโดเมน

ประเภทของความไม่ต่องกันสามารถแบ่งได้ในหลายรูปแบบ ได้แก่ การแบ่งประเภทตามการไม่เป็นไปตามกฎความต่องกันแต่ละกฎ การแบ่งประเภทตามการทำงานที่เป็นสาเหตุให้เกิดความไม่ต่องกัน และการแบ่งประเภทตามผลกระทบของความไม่ต่องกันที่เกิดขึ้น

การวินิจฉัยความไม่ตรงกัน (Diagnosing Inconsistency) ประกอบด้วย

- 1) การหาตำแหน่งความไม่ตรงกัน (Locating) เช่น ส่วนที่พบการอธิบายที่ทำให้เกิดความไม่ตรงกัน
- 2) การหาสาเหตุของความไม่ตรงกัน (Identifying the Cause) เช่น ความไม่ตรงกันเกิดจากการไม่เป็นไปตามกฎ เนื่องจากข้อมูลมีการสูญหาย ทำให้เกิดความไม่เข้าใจ
- 3) การจัดกลุ่มความไม่ตรงกัน (Classifying) เพื่อสามารถจัดการกับความไม่ตรงกันได้ อย่างเหมาะสม

ผู้วิจัยใช้ขั้นตอนเหล่านี้ เป็นแนวทางในการตรวจสอบความไม่ตรงกันในส่วนความต้องการด้านหน้าที่ และสร้างกฎในการตรวจสอบ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องจะแบ่งได้เป็น 4 ด้าน คือ มาตรฐานของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เลือกใช้เป็นมาตรฐาน การตัดคำและการกำกับหน้าที่คำภาษาไทยที่เลือกใช้งานวิจัยที่มีการประมวลผลภาษาธรรมชาติ และเทคนิคการตรวจหาความไม่ตรงกัน

มาตรฐานของเอกสารข้อกำหนดความต้องการซอฟต์แวร์

2.2.1 การพัฒนาแนวทางสำหรับการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ (Development a Guideline for Preparing Software Requirements Specification Document) [2] นำฝน อัสวเมชิน (Assawamekin N.)

นำฝน อัสวเมชิน ได้ทำวิจัยเกี่ยวกับการหามาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เหมาะสมกับการใช้ภายในประเทศ ซึ่งมีการศึกษามาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ต่าง ๆ ได้แก่ IEEE Std 830-1993, DoD-STD-498, ESA PSS-05-0 และ NASA-DID-P200 โดยนำมาเปรียบเทียบข้อดีข้อเสียของแต่ละมาตรฐาน เพื่อเป็นแนวทางเบื้องต้นในการจัดทำเอกสารข้อกำหนดความต้องการซอฟต์แวร์แล้วทำการวิจัยโดยการออกแบบสอบถามไปยังหน่วยงานต่าง ๆ ทั้งหน่วยงานราชการ รัฐวิสาหกิจ และบริษัทเอกชน ซึ่งสอบถามถึงข้อมูลเกี่ยวกับองค์ประกอบของผู้ตอบ ข้อมูลเกี่ยวกับผู้ตอบ และข้อมูลเกี่ยวกับโครงการ เพื่อนำข้อมูลจากการตอบแบบสอบถามมาวิเคราะห์ว่า เนื้อหาส่วนใดควรมีอยู่ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ และมีหลักเกณฑ์ทางสถิติมาใช้ในการวิเคราะห์ข้อมูลที่ได้จากแบบสอบถาม รวมทั้งข้อมูลที่เลือกจากการนำส่วนดีของมาตรฐานต่าง ๆ มารวมกัน และปรับปรุงให้เหมาะสมกับการใช้ภายในประเทศ

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ แผ่นแบบ (Template) ของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทั้งภาษาอังกฤษและภาษาไทย ที่แบ่งเป็น 3 ระดับ คือ ระดับเริ่มต้น (Beginner) ระดับปานกลาง (Intermediate) และระดับผู้เชี่ยวชาญ (Expert) ดังนั้นในงานวิจัยของผู้ทำวิจัยจึงเลือกแผ่นแบบของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ในระดับผู้เชี่ยวชาญ เป็นมาตรฐานของข้อกำหนดความต้องการซอฟต์แวร์ที่จะทำการวิจัย (รายละเอียดแสดงในภาคผนวก ก.)

การตัดคำและการกำกับหน้าที่คำภาษาไทย

2.2.2 การตัดคำไทยโดยใช้คุณลักษณะ (Feature-based Thai Segmentation) [7] ไพศาล เจริญพรสวัสดิ์ (Charoenpornswat P.)

ไพศาล เจริญพรสวัสดิ์ ได้ทำวิจัยการตัดคำโดยลักษณะเลือกแบบเหมือนมากที่สุด (Maximal Matching) การกำหนดหน้าที่คำด้วยโมเดลไตรแกรม (Part-of-speech Trigram Model) และใช้การเรียนรู้ของเครื่อง (Machine Learning) คือ ริปเปอร์ (Ripper-repeated incremental pruning to produce error reduction) ที่เป็นการเรียนรู้จากตัวอย่างต่าง ๆ ที่ป้อนให้ แล้วทำการสรุปกฎต่าง ๆ ขึ้นมาเองและวินโนว์ (Winnow) ที่เป็นการเรียนรู้จากตัวอย่างต่าง ๆ ที่ป้อนให้เช่นกัน แต่กฎที่ได้จะอยู่ในรูปของโครงข่าย (Window Network) โดยเมื่อเริ่มการตัดคำใช้การตัดคำเลือกแบบเหมือนมากที่สุด เป็นการตัดคำโดยใช้พจนานุกรม ซึ่งจะทำการตัดคำที่เป็นไปได้ในทุก ๆ แบบออกมา จากนั้นจะมีการพิจารณาหน้าที่ของคำ เข้ามาช่วยในการแก้ปัญหาคำกำกวม ซึ่งมีการใช้ค่าทางสถิติมาพิจารณาความต่อเนื่องของคำ ค่าทางสถิตินั้นหาได้จากโมเดลไตรแกรมนั้นเอง ทำให้ผลลัพธ์ของการตัดคำที่ได้มีความถูกต้องมากขึ้น รวมถึงมีการใช้การเรียนรู้ของเครื่อง ริปเปอร์ และวินโนว์ เข้าช่วยในการคำนวณหาค่าคุณลักษณะต่าง ๆ ของคำ เพื่อนำเอาคุณลักษณะต่าง ๆ ที่ได้จากการเรียนรู้มาช่วยในการพิจารณาตัดคำ

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ โปรแกรมตัดคำภาษาไทย ซึ่งสามารถบอกหน้าที่ของคำหลังจากที่ตัดคำเรียบร้อยแล้ว ผู้วิจัยจึงเลือกตัวโปรแกรมนี้เป็นส่วนตัดคำในขั้นตอนขั้นต้นของการประมวลผลภาษาธรรมชาติ และนำคำที่ตัดได้ไปทำการประมวลผลต่อไป

งานวิจัยที่มีการประมวลผลภาษาธรรมชาติ

2.2.3 แกรมมาไลเซอร์ : เครื่องมือวิศวกรรมซอฟต์แวร์ที่ใช้คอมพิวเตอร์ช่วยตามหลักการวิเคราะห์ข้อความ (The Grammalizer: A CASE Tool Based on Textual Analysis) [8] Jeroen Hoppenbrouwers, Willem-Jan van den Heuvel, Stijn Hoppenbrouwers, Hans Weigand and Olga de Troyer

Jeroen Hoppenbrouwers และคณะได้ทำงานวิจัยอธิบายการออกแบบและพัฒนา (Implementation) แกรมมาไลเซอร์ ซึ่งเป็นเครื่องมือวิศวกรรมซอฟต์แวร์ที่ใช้คอมพิวเตอร์ช่วยสำหรับการวิเคราะห์ข้อความของเอกสาร ตามหลักของการวิเคราะห์รูปร่างลักษณะไวยากรณ์ (Morphosyntactic NL Analysis) และคิสส์ เมทริออด (KISS Method) แนวคิดของ KISS หลังจากการแปลภาษาธรรมชาติ จะวิเคราะห์ข้อความจากประโยคที่ประกอบด้วย คำนาม (Noun) คำกริยา (Verb) และคำอื่น ๆ ออกมาเป็น ประธาน (Subject) สาร (Message) กรรม (Object) และ

การกระทำ (Action) ลักษณะของ แกรมมาไลเซอร์ เป็นเครื่องมือวิศวกรรมซอฟต์แวร์ที่ใช้คอมพิวเตอร์ช่วยในระดับสูง (Upper CASE Tool) มีสถาปัตยกรรม (Architecture) 2 ส่วน คือ ตัวกำกับหน้าที่คำ (Tagger) กับ ตัววิเคราะห์ (Analyzer) โดยใช้ CORBA เป็นตัวติดต่อสื่อสารในระบบไคลเอนท์/เซิร์ฟเวอร์ (Client/Server) มีกระบวนการทำงาน คือ รับข้อมูลนำเข้าที่เป็นข้อความผ่านทางเครือข่าย (Network) แยกข้อความเป็นส่วนที่เกี่ยวข้อง (Relevant) กับไม่เกี่ยวข้อง (Irrelevant) แล้วทำการกำกับหน้าที่คำตามแนวคิด KISS สร้างแทค (Tags) และแผ่นแบบ (Template) ที่เป็นแม่แบบ (Paradigm) ของ ข้อกำหนด เพื่อนำไปสร้างเป็นแบบจำลองต่อไป

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ เครื่องมือวิศวกรรมซอฟต์แวร์ที่ใช้คอมพิวเตอร์ช่วยที่เรียกว่า แกรมมาไลเซอร์ สามารถรับข้อมูลนำเข้าเป็นข้อความ แล้ววิเคราะห์ออกมาสร้างเป็นแบบจำลองเชิงแนวคิด ส่วนของงานวิจัยนี้ที่ผู้วิจัยนำมาประยุกต์ใช้คือแนวคิดในการกำหนดหน้าที่ของคำและวิเคราะห์ข้อความในประโยค

2.2.4 รายงาน เรื่อง “การแนะนำการจัดจำพวกทั่วไปและการหาสภาพที่เป็นส่วนจำเพาะของการแปลส่วนสำคัญจากข้อความในเนื้อหาของวิศวกรรมซอฟต์แวร์ (Introducing Genericity and Modularity of Textual Scenario Interpretation in the Context of Requirements Engineering)” [9] ของ ESPRIT Reactive Long Term Research CREWS (Cooperative Requirements Engineering With Scenarios) Camille Ben Achour และ Colette Rolland

Camille Ben Achour และ Colette Rolland ได้ทำวิจัยเพื่อจำกัดความเรื่องหน้าที่และข้อจำกัดของการแปลภาษาธรรมชาติในกระบวนการวิศวกรรมความต้องการ (Requirements Engineering Process) โดยมีการพัฒนาเครื่องมือในการแปลภาษาธรรมชาติ ซึ่งมีความสามารถที่สำคัญ 2 ส่วน คือ ส่วนแรกเป็นการจัดจำพวกทั่วไป (Genericity) และส่วนที่สองเป็นการหาสภาพที่เป็นส่วนจำเพาะ (Modularity) งานวิจัยนี้มีการกล่าวถึงกระบวนการของการแปลภาษาทางด้านภาษาศาสตร์ที่สำคัญแบ่งเป็น 3 ระดับ คือ ระดับคำศัพท์ (Lexical Level) ระดับไวยากรณ์ (Syntactic Level) และระดับความหมาย (Semantic Level) ซึ่งงานวิจัยนี้เน้นในระดับการแปลความหมาย โดยสร้างไวยากรณ์ของเคส (Case Grammar) หรือเรียกว่า เคสความหมาย (Semantic Cases) เช่น ตัวควบคุมการทำงาน (Agent) วัตถุ (Object) แหล่งที่มา (Source) แหล่งปลายทาง (Destination) มาใช้เป็นกฎในการแปลความหมายจากกริยาที่แสดงการทำงาน (Action Verbs) และสร้างรูปแบบความหมายของประโยค (Sentence Semantic Pattern) เช่น ประโยคที่เป็นข้อจำกัด (Constraint) ประโยคที่เป็นลำดับ (Sequence) หรือประโยคที่เป็นตัวกระตุ้นการทำงาน (Trigger) ซึ่งรูปแบบและกฎที่สร้างขึ้นอยู่ในรูปภาษาเชิงตรรกะ (Logic Language)

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ เครื่องมือในการแปลภาษาธรรมชาติเป็นภาษาทางตรรกะ และสามารถเปรียบเทียบความไม่ตรงกันระหว่างส่วนเนื้อหา (Context) กับแผนภาพลำดับการทำงาน (Sequence Diagram) ได้ โดยตรวจสอบจากภาษาทางตรรกะ ที่ได้มาจากการแปลภาษาธรรมชาติและแผนภาพลำดับการทำงาน ส่วนของงานวิจัยนี้ที่ผู้วิจัยนำมาประยุกต์ใช้คือ แนวคิดในการแปลภาษาธรรมชาติและการวิเคราะห์ความหมายของภาษาธรรมชาติ

2.2.5 วิธีการตัดใจความจากเอกสารภาษาไทย (Information Extraction Strategies for Thai Documents) [15] Rattasit Sukhahuta และ Dan Smith

Rattasit Sukhahuta และ Dan Smith ได้ทำวิจัยโดยออกแบบและพัฒนาระบบสำหรับตัดใจความที่ต้องการออกมา จากเอกสารจำนวนมาก ความหมายของการตัดใจความ คือ การอ่านเอกสารอย่างคร่าว ๆ (Skim) แล้วดึงข้อมูลที่เกี่ยวข้องกับคำสำคัญ (Trigger Term) ออกมาจากข้อความในเอกสารที่มีข้อความจำนวนมาก ข้อมูลที่ต้องการอาจอยู่ในรูปเป็นประโยคหรือวลี วิธีดังกล่าวมีการกำหนดโดเมนของคำสำคัญ เพื่อให้ทราบว่าถ้าคำสำคัญเป็นคำนี้ ควรดึงข้อมูลอะไรออกมาบ้าง ระบบที่พัฒนาขึ้นจากงานวิจัยนี้ประกอบด้วย การทำงานในส่วนย่อย ๆ หลายส่วน ได้แก่ การตัดคำ (Word Tokenization) การกำกับหน้าที่คำ การวิเคราะห์โครงสร้างพื้นฐานประโยค (Surface Structure Analysis) ทำการประมวลผล โดยอาศัยฐานความรู้ คือ พจนานุกรม (Dictionary) คลังข้อความ (Corpus) ไวยากรณ์ของภาษา การคำนวณค่าทางสถิติโดยใช้ไบแกรมโมเดล (Bigram Model) และไตรแกรมโมเดล (Trigram Model) เพื่อสร้างเอกสารให้อยู่ในรูปแบบต้นไม้โครงสร้างวลี (Phrase Structure Tree) แล้วทำการตัดใจความที่ต้องการจากต้นไม้โครงสร้างวลี รวมถึงมีการประเมินประสิทธิภาพด้านความถูกต้องของผลลัพธ์ที่ได้ จากการตัดคำ การกำกับหน้าที่คำ และข้อมูลที่ต้องการ

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ ระบบตัดใจความ ที่สามารถดึงข้อมูลที่ต้องการออกจากเอกสารจำนวนมาก และมีขนาดใหญ่ ได้อย่างรวดเร็วและมีความถูกต้อง สิ่งที่ผู้วิจัยนำมาประยุกต์ใช้จากงานงานชิ้นนี้ คือ ไวยากรณ์ภาษาไทย และการออกแบบระบบ ซึ่งคล้ายคลึงกับแนวคิดของผู้วิจัย เป็นอย่างมาก

เทคนิคการตรวจหาความไม่ตรงกัน

2.2.6 วิธีการตรวจหาและแปลความหมายของความไม่ตรงกันในข้อกำหนดความต้องการซอฟต์แวร์ (A Detecting and Interpreting Method of the Inconsistency of Software Requirements Specification) [1] Hideaki Sugimoto and Atsushi Ohnishi

งานวิจัยนี้มีการสร้างแบบจำลองกรอบของความต้องการ (Requirement Frame Model) ได้แก่ กรอบที่หนึ่งคือ กรอบคำนาม (Noun Frame) ที่อธิบายชนิดและความหมายของคำนาม (Type and Meaning) เช่น คำประเภท “คน (Human)” มีความหมาย คือ “วัตถุที่กระทำ (Active Object) และวัตถุภายนอก (External Object)” คำประเภท “ฟังก์ชัน” มีความหมาย คือ “วัตถุที่กระทำและวัตถุภายใน (Internal Object)” คำประเภท “แฟ้มข้อมูล (File)” มีความหมาย คือ “วัตถุที่ถูกกระทำ (Passive Object) ของกลุ่มข้อมูล (Information Set)” กรอบที่สองคือ กรอบกรณี (Case Frame) ที่ประกอบด้วย คำนาม คำกริยา และกรณี (Cases) ตัวอย่างของกรณี เช่น ตัวควบคุมการทำงาน (Agent) จุดหมาย (Goal) แหล่งที่มา การดำเนินการ (Operation) มีการสร้างแนวคิดหลัก (Concepts) ขึ้นมาเพื่อหาความสัมพันธ์ของคำกริยากับเคส มีทั้งหมด 16 แนวคิด ตัวอย่างเช่น การไหลของข้อมูล ต้องมีแหล่งที่มา ตัวควบคุมการทำงาน จุดหมาย และ เครื่องมือ (Instrument) กรอบสุดท้ายคือ กรอบฟังก์ชัน (Function Frame) ที่อธิบายประโยชน์ของความ ต้องการให้สอดคล้องกับแนวคิดของกรอบกรณี มีทั้งหมด 12 กรอบ ตัวอย่างเช่น การสร้างข้อมูล (Data Creation) การประมวลผลข้อมูล (Data Processing) การไหลของข้อมูล (Data Flow) มีการพัฒนาเครื่องมือซอฟต์แวร์ทางด้านภาษาความต้องการที่เป็นข้อความ (Text-based Requirements Language) ชื่อว่า X-JRDL เป็นภาษาในการเขียนความต้องการที่สามารถแปลงให้อยู่ในรูปของกรอบกรณีได้ง่าย หลังจากนั้นจะเปรียบเทียบความไม่ตรงกันในประโยคเดียวกัน และความไม่ตรงกันระหว่างแต่ละประโยค จากแบบจำลองกรอบทั้ง 3 กรอบ แล้วใช้ความน่าจะเป็นเลือกว่าประโยคที่ถูกต้องควรเป็นประโยคใด

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ วิธีตรวจหาและแปลความหมายของความไม่ตรงกันในข้อกำหนดความต้องการซอฟต์แวร์ ส่วนของงานวิจัยนี้ที่ผู้วิจัยนำมาประยุกต์ใช้คือ แนวคิดวิธีการตรวจหาความไม่ตรงกัน

2.2.7 การจัดการความไม่ตรงกันในข้อกำหนดความต้องการซอฟต์แวร์จากหลายๆ มุมมอง (Inconsistency Handling in Multiperspective Specifications) [10]
โดย Anthony C.W. Finkelstein, Dov Gabbay, Anthony Hunter, Jeff Kramer and Bashar Nuseibeh

งานวิจัยนี้มีการนำเอามุมมอง (View Point) แต่ละมุมมองของข้อกำหนดความต้องการซอฟต์แวร์มาตรวจสอบความไม่ตรงกันระหว่างแต่ละมุมมอง ซึ่งในแต่ละมุมมองจะประกอบด้วย การแทน (Representation) การพัฒนา (Development) และ ข้อกำหนด (Specification) ในส่วนของข้อกำหนดจะอยู่ในรูปของ 5 ส่วนย่อย ได้แก่ แหล่งที่มา ข้อมูลนำเข้า การทำงาน ข้อมูลนำออก และแหล่งปลายทาง หรือเรียกรวมกันว่า ตารางการทำงาน (Actions Tables) งานวิจัยนี้คิดกฎในการตรวจสอบความไม่ตรงกันระหว่างต่างมุมมอง ซึ่งประยุกต์มาจากคอร์เมทรีอิด (COntrolled Requirements Expression – CORE) แล้วทำการแปลงจากมุมมองต่าง ๆ ให้เป็นรูปแบบของภาษาเชิงตรรกะที่สร้างขึ้น เพื่อทำการตรวจสอบความไม่ตรงกันตามกฎหมายที่กำหนด

ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ วิธีการจัดการความไม่ตรงกันของข้อกำหนดความต้องการซอฟต์แวร์จากหลายๆ มุมมอง ส่วนของงานวิจัยนี้ที่ผู้วิจัยนำมาประยุกต์ใช้คือ ใช้กฎการตรวจสอบความไม่ตรงกันในส่วนของข้อกำหนด เป็นแนวทางในการตรวจสอบความไม่ตรงกันในส่วนความต้องการด้านหน้าที่

บทที่ 3

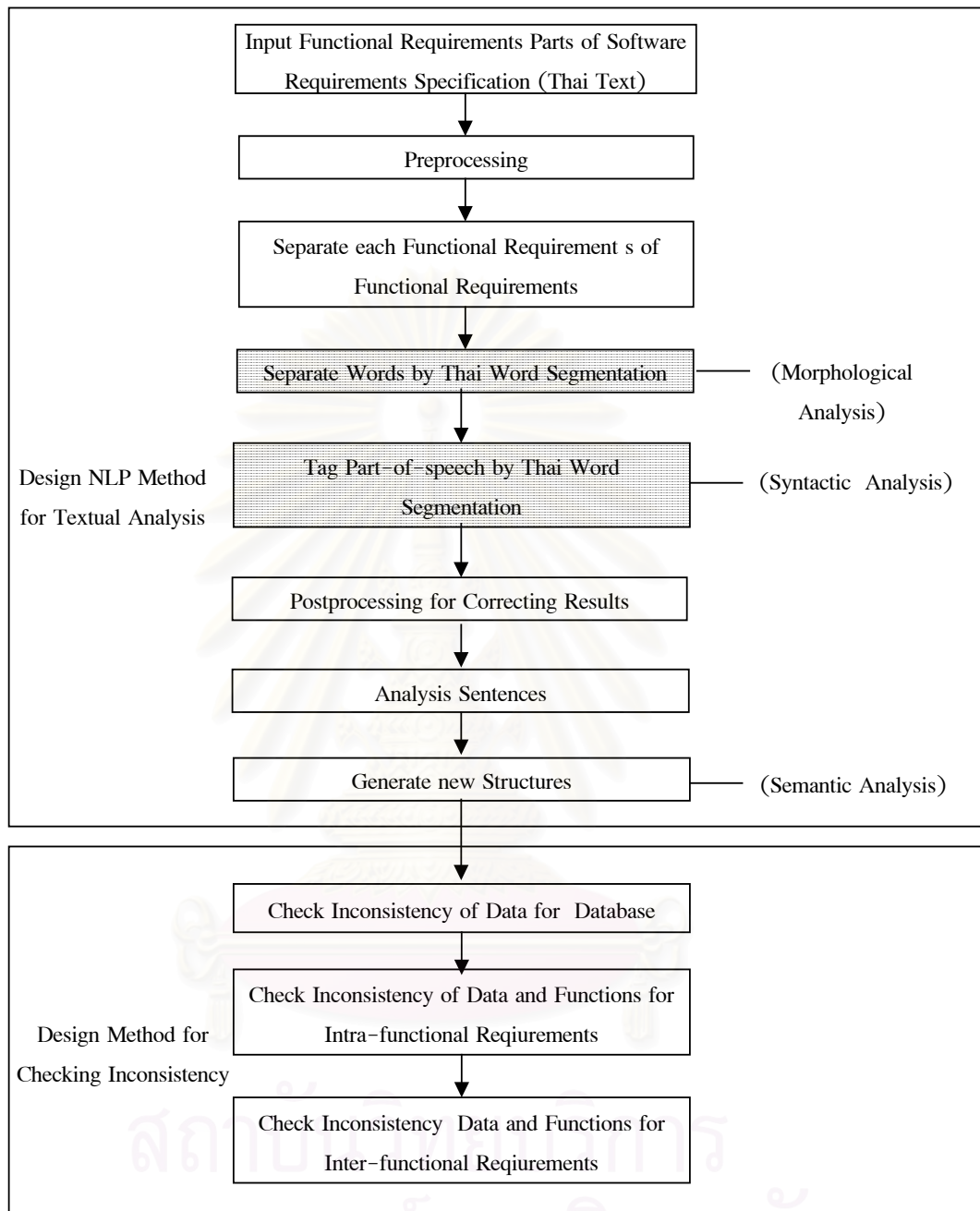
การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนด ความต้องการซอฟต์แวร์

3.1 แนวคิดการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

งานวิจัยนี้มีแนวคิดว่าการตรวจสอบความไม่ตรงกันจากตัวเอกสารที่เขียนด้วยภาษาธรรมชาติโดยตรงนั้นสามารถทำได้ยาก เพราะลักษณะการเขียนเอกสาร การใช้โครงสร้างประโยค และการใช้ภาษาของแต่ละเอกสารมีความแตกต่างกันค่อนข้างมาก ดังนั้นจึงต้องมีการประมวลผลภาษาธรรมชาติในตัวเอกสาร ให้อยู่ในรูปแบบโครงสร้างแบบใหม่ตามที่งานวิจัยนี้กำหนดขึ้น แล้วทำการตรวจสอบความไม่ตรงกันจากโครงสร้างใหม่นั้น แนวคิดการออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ แสดงดังรูปที่ 3.1 แบ่งการออกแบบขั้นตอนวิธีเป็น 2 ส่วน คือ

- 1) ขั้นตอนวิธีการประมวลผลภาษาธรรมชาติในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ในส่วนความต้องการด้านหน้าที่ ซึ่งประกอบด้วย กระบวนการเตรียมเอกสาร (Preprocessing) การตัดคำภาษาไทย การกำกับหน้าที่คำ การวิเคราะห์ประโยค (Sentence Analysis) และการสร้างโครงสร้างใหม่เพื่ออธิบายความต้องการด้านหน้าที่
- 2) ขั้นตอนวิธีการตรวจสอบความไม่ตรงกันของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ในส่วนความต้องการด้านหน้าที่ ซึ่งประกอบด้วย การตรวจสอบความไม่ตรงกันของข้อมูลในฐานข้อมูล ความไม่ตรงกันของฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกในความต้องการด้านหน้าที่เดียวกัน ความไม่ตรงกันของฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกระหว่างความต้องการด้านหน้าที่ที่ต่างหน้าที่กัน

การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ตรงกัน ในส่วนขั้นตอนวิธีการประมวลผลภาษาธรรมชาติในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ จะอธิบายอย่างละเอียดในหัวข้อ 3.2 และส่วนขั้นตอนวิธีการตรวจสอบความไม่ตรงกันของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ในส่วนความต้องการด้านหน้าที่ จะอธิบายอย่างละเอียดในหัวข้อ 3.3 โดยเรียงลำดับตามขั้นตอนวิธีที่ออกแบบในรูปที่ 3.1



รูปที่ 3.1 แนวความคิดการออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ตรงกัน

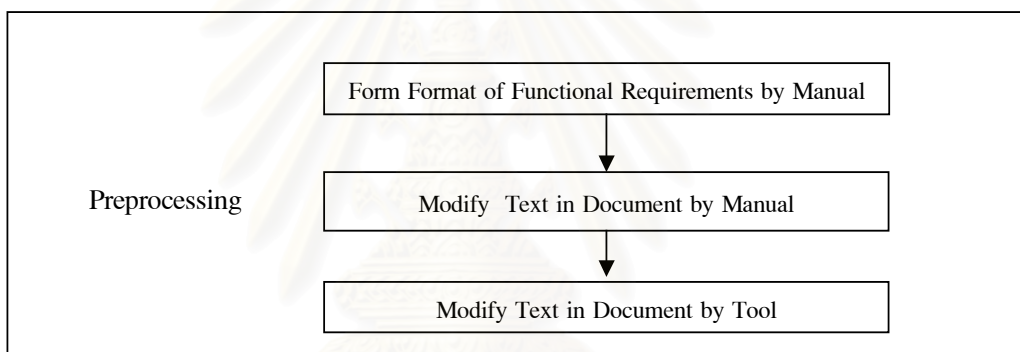
3.2 การออกแบบขั้นตอนวิธีการประมวลผลภาษาธรรมชาติ ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

3.2.1 กระบวนการเตรียมเอกสาร

ในส่วนนี้เป็นการนำเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เฉพาะส่วนความต้องการด้านหน้าที่มาทำการจัดรูปแบบเอกสาร โครงสร้างประโยค ลักษณะข้อความที่ใช้ในการเขียนเอกสาร ให้เหมาะสมสำหรับการนำไปประมวลผลในขั้นถัดไป ประกอบด้วย 3 ขั้นตอน (แสดงดังรูปที่ 3.2) คือ

- 1) การจัดรูปแบบเอกสารให้ตรงตามมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยที่งานวิจัยนี้เลือกใช้ด้วยมือ (Form Format of Functional Requirements by Manual) ตามงานวิจัยที่เกี่ยวข้องในหัวข้อ 2.2.1 [2] (ตัวอย่างแบบเอกสารแสดงในภาคผนวก ก.) ซึ่งมีการอธิบายความต้องการด้านหน้าที่แต่ละหน้าที่ออกเป็นส่วนต่าง ๆ 5 ส่วนด้วยกัน คือ ชื่อความต้องการด้านหน้าที่ (Name) บทนำ (Introduction) ข้อมูลนำเข้า กระบวนการ (Processing) และข้อมูลนำออก เหตุผลในการเลือกใช้มาตรฐานเอกสารนี้ เนื่องจากมาตรฐานเอกสารภาษาไทยนี้มีรูปแบบเหมาะสมสำหรับการแปลงเป็นยูสเคสตามทฤษฎีที่เกี่ยวข้อง 2.1.4 [13] (ตัวอย่างยูสเคสในภาคผนวก ข.) ซึ่งเป็นการอธิบายฟังก์ชันการทำงานของซอฟต์แวร์ด้วยภาษาธรรมชาติ ก่อนที่จะนำไปสร้างเป็นแผนภาพการไหลของข้อมูล และมีความสอดคล้องกับโครงสร้างใหม่ที่งานวิจัยนี้จะสร้างขึ้นเพื่อใช้อธิบายแทนความต้องการด้านหน้าที่ที่เป็นข้อความภาษาธรรมชาติ (รายละเอียดจะอธิบายในหัวข้อ 3.2.4 การสร้างรูปแบบโครงสร้างใหม่)
- 2) การแก้ไขเปลี่ยนแปลงข้อความในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้วยมือ (Modify Text by Manual) เพื่อให้มีโครงสร้างประโยคที่ถูกต้อง ลักษณะการเขียนเอกสารไม่ซับซ้อน ไม่มีการใช้คำขยายมากเกินไปในประโยค และมีข้อมูลแสดงส่วนที่สำคัญของความต้องการด้านหน้าที่อย่างครบถ้วน ตัวอย่างเช่น ในส่วนของข้อมูลนำเข้า ต้องมีประโยคที่แสดงถึงข้อมูลนำเข้าและแหล่งที่มาของข้อมูล ในส่วนของข้อมูลนำออกต้องมีประโยคที่แสดงถึงข้อมูลนำออกและแหล่งปลายทางของข้อมูล
- 3) การแก้ไขเปลี่ยนแปลงข้อความในเอกสารข้อกำหนดความต้องการซอฟต์แวร์โดยอัตโนมัติด้วยเครื่องมือซอฟต์แวร์ (Modify Text by Tool) เพื่อกำจัดตัวอักษร (Characters) บางตัวที่มีผลต่อโปรแกรมตัดคำภาษาไทย ได้แก่

- การเว้นวรรคระหว่างคำกับเครื่องหมายไม้มยมก (๗) จะมีการแก้ไขเปลี่ยนแปลงข้อความโดยกำจัดเว้นวรรคระหว่างคำกับไม้มยมกออกไป เพื่อให้ผลลัพธ์ที่ได้จากโปรแกรมตัดคำภาษาไทยนั้น ตัดคำที่อยู่หน้าเครื่องหมายไม้มยมกกับเครื่องหมายไม้มยมกรวมเป็นคำเดียวกัน ตัวอย่างเช่น ข้อความเดิม คือ “ต่าง ๆ” ผลลัพธ์ของการตัดคำจากข้อความเดิม จะแยกออกมาเป็น 2 คำ ได้แก่ “ต่าง” และ “ๆ” ซึ่งไม่ถูกต้อง จะมีการแก้ไขเป็นข้อความใหม่ คือ “ต่างๆ” ซึ่งหลังจากตัดคำจะได้เป็นคำคำเดียว คือ “ต่างๆ”
- การเว้นวรรคโดยการใช้ตัวอักษรแท็บ (Tab) จะมีการแก้ไขเปลี่ยนแปลงข้อความโดยเปลี่ยนตัวอักษรแท็บให้เป็นการเว้นวรรคธรรมดา เพื่อให้ผลลัพธ์ที่ได้จากโปรแกรมตัดคำภาษาไทยนั้น สามารถตรวจสอบได้ว่าตัวอักษรนี้เป็นการเว้นวรรค ไม่ใช่ตัวอักษรแท็บที่โปรแกรมตัดคำภาษาไทยเห็นเป็นคำ



รูปที่ 3.2 ขั้นตอนการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์

3.2.2 การตัดคำภาษาไทยและการกำกับหน้าที่คำ

ในส่วนนี้เป็นการนำเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ในส่วนความต้องการด้านหน้าที่ซึ่งเป็นแฟ้มข้อมูลชนิดข้อความ (.txt) ที่ผ่านกระบวนการเตรียมเอกสารแล้ว เข้ามาทำการประมวลผล โดยแยกเป็นแต่ละหน้าที่ความต้องการ แต่ละส่วนของหน้าที่ แต่ละประโยค และแต่ละคำจากเอกสารข้อกำหนดความต้องการซอฟต์แวร์ และทำการกำกับหน้าที่คำ ซึ่งเป็นการประมวลผลวากยสัมพันธ์ทางภาษา (Syntactic Processing) เพื่อกำหนดว่าแต่ละคำในประโยคทำหน้าที่เป็นคำใด เช่น คำนาม คำกริยา คำคุณศัพท์ (Adjective) คำกริยาวิเศษณ์ (Adverb) คำสันธาน (Conjunction) คำบุพบท (Preposition) ฯลฯ โดยใช้โปรแกรมตัดคำภาษาไทย [7] [17] มาช่วยในการตัดคำและกำกับหน้าที่ของคำ

โปรแกรมตัดคำที่เลือกมาใช้ในการวิจัย คือ โปรแกรม SWATH [17] พัฒนาโดยนักวิจัยของ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ซึ่งสามารถกำกับหน้าที่ของคำ ให้โดยอัตโนมัติหลังจากที่มีการตัดคำเรียบร้อยแล้ว ผลลัพธ์ที่ได้จากการตัดคำของ SWATH จะเป็น คำศัพท์แยกเป็นคำ ๆ พร้อมทั้งหน้าที่ของคำกำกับไว้ แต่พบว่ามีส่วนที่ผลลัพธ์ไม่ถูกต้อง เกิด ข้อผิดพลาดไม่สามารถนำผลลัพธ์ที่ได้ไปประมวลผลในขั้นต่อไปทันที ดังนั้นจำเป็นต้องมีการคิดขั้นตอนวิธีที่เข้ามาช่วยให้การตัดคำและกำกับหน้าที่คำมีความถูกต้องมากขึ้น

หมายเหตุ การปรับปรุงแก้ไขผลลัพธ์ของการตัดคำนั้น ในการทำงานจริงจะทำหลังจากที่ผ่านขั้นตอนการกำกับหน้าที่คำแล้ว เนื่องจากการตัดคำและการกำกับหน้าที่คำเป็นการใช้ โปรแกรมที่มีผู้พัฒนาไว้แล้วมาใช้ซึ่งเป็นโปรแกรมเดียวกัน ดังนั้นลักษณะการทำงานจะส่งข้อมูลนำ เข้าซึ่งเป็นข้อความต่อกันยาว ๆ เข้าสู่โปรแกรม SWATH แล้วได้ผลลัพธ์สุดท้ายเป็นข้อมูลนำออกก็ เป็นคำแต่ละคำพร้อมหน้าที่คำกำกับไว้ หลังจากนั้นจึงนำผลลัพธ์ที่ได้มาทำการปรับปรุงแก้ไขให้มีความถูกต้องมากขึ้น

โปรแกรมตัดคำภาษาไทย ในส่วนการตัดคำ มีรายละเอียด คือ

- 1) มีอัลกอริทึมในการตัดคำ 4 วิธีด้วยกัน คือ การเลือกคำยาวที่สุดที่พบในพจนานุกรม (Longest Matching) การเลือกคำที่เหมือนที่สุดที่จะเกิดขึ้นได้ (Maximal Matching) การใช้ค่าทางสถิติที่คำนวณจากโปรแกรมโมเดล และการใช้คุณลักษณะในภาษาไทย เข้ามาช่วยในการตัดคำ (Feature-based Thai)
- 2) มีคลังข้อความ และพจนานุกรมฝังอยู่ในตัวโปรแกรม คลังข้อความที่ใช้มีชื่อว่า ออริคิด (Orchid) [16]

โปรแกรมตัดคำภาษาไทย ในส่วนการกำกับหน้าที่คำ (Tagger) มีรายละเอียด คือ

- 1) มีอัลกอริทึมในการกำกับหน้าที่คำ คือ ใช้ค่าทางสถิติที่คำนวณจากโปรแกรมโมเดล และการใช้คุณลักษณะในภาษาไทยเข้ามาช่วยในการตัดคำ (Feature-based Thai)
- 2) สำหรับคลังข้อความออริคิด ซึ่งฝังอยู่ในโปรแกรม จะมีหน้าที่ของคำ (Tag) ทั้งหมด 47 หน้าที่ (รายละเอียดในภาคผนวก ค.) แบ่งเป็น กลุ่มหน้าที่ประเภทคำนาม กลุ่มหน้าที่ประเภทคำสรรพนาม กลุ่มหน้าที่ประเภทคำกริยา กลุ่มหน้าที่ประเภทคำสันธาน กลุ่มหน้าที่ประเภทคำบุพบท กลุ่มหน้าที่ประเภทคำกริยาช่วย (Auxiliary) กลุ่มหน้าที่ประเภทคำกริยาวิเศษณ์ และกลุ่มหน้าที่ประเภทคำอื่น ๆ เช่น คำอุทาน (Interjection) คำลงท้ายประโยคเพื่อตอบรับ (Ending for Affirmative Sentence) คำลงท้ายประโยคเพื่อถามความคิดเห็น (Ending for Interrogative Sentence) คำปฏิเสธ (Negator) เครื่องหมายวรรคตอน (Punctuation)

ปัญหาข้อผิดพลาดของโปรแกรมตัดคำ SWATH และวิธีการปรับปรุงแก้ไขให้มีความถูกต้องมากขึ้น ได้แก่

- 1) ปัญหาในเรื่องคำสำคัญ (Keyword) ที่แสดงถึงส่วนต่าง ๆ ของความต้องการด้านหน้าที่ ได้แก่คำว่า “ความต้องการด้านหน้าที่” “ข้อมูลนำเข้า” “ข้อมูลนำออก” ถูกตัดแยกเป็นคำย่อยหลายคำ และมีการกำกับหน้าที่คำเหล่านั้นแยกเป็นแต่ละคำทำให้ไม่สามารถตรวจสอบได้ว่าข้อความหรือประโยคที่ทำการประมวลผลนั้น อยู่ในส่วนใดของความต้องการด้านหน้าที่ วิธีการปรับปรุงแก้ไข คือ
 - ตรวจสอบว่าคำเดิมของคำย่อยเหล่านั้นก่อนเข้าสู่โปรแกรมตัดคำภาษาไทย ตรงกับคำเฉพาะที่แสดงถึงส่วนต่าง ๆ ของความต้องการด้านหน้าที่หรือไม่ และในบรรทัดนั้นของเอกสารมีแต่คำเฉพาะที่แสดงถึงส่วนต่าง ๆ ของความต้องการด้านหน้าที่เท่านั้น แล้วทำการแก้ไขผลลัพธ์ที่ออกมาจากโปรแกรมตัดคำ โดยระบุว่าคำเหล่านั้นเป็นคำสำคัญที่แสดงถึงส่วนต่าง ๆ ของความต้องการด้านหน้าที่ โดยคำเดิมจะไม่ถูกแยกออกเป็นคำย่อย
 - แก้ไขการกำกับหน้าที่คำเป็นคำสำคัญในการแยกส่วนความต้องการด้านหน้าที่ (สร้างหน้าที่คำใหม่ขึ้นมา ไม่กำกับหน้าที่ของคำตามหน้าที่คำในคลังข้อความ)
- 2) ปัญหาในเรื่องคำศัพท์ที่เป็นคำเฉพาะ (Specific Word) โปรแกรมตัดคำยังไม่สามารถตัดคำเป็นคำที่ถูกต้องได้ คำเฉพาะถูกแยกออกเป็นคำย่อยหลาย ๆ คำ และมีการกำกับหน้าที่คำเหล่านั้นแยกเป็นแต่ละคำ วิธีการปรับปรุงแก้ไข คือ
 - ใช้ข้อมูลคำในฐานะข้อมูลที่เก็บคำศัพท์เฉพาะมาช่วยในการตัดคำ เพื่อให้สามารถรวมคำเฉพาะที่ถูกแยกออกเป็นคำย่อยให้กลับมาเป็นคำเดิมได้
 - แก้ไขการกำกับหน้าที่คำให้คำที่รวมเป็นคำเฉพาะคำเดิมนั้น มีหน้าที่เป็นคำวิสามานยนาม (Proper Noun) ซึ่งหมายถึงคำนามเฉพาะทั่ว ๆ ไป
- 3) ปัญหาในเรื่องคำศัพท์ที่เป็นคำภาษาต่างประเทศ (Foreign Language) โปรแกรมตัดคำยังไม่สามารถตัดคำเป็นคำที่ถูกต้องได้ คำภาษาต่างประเทศบางคำที่เกิดจากคำที่มากกว่าหนึ่งคำมารวมกัน เช่น หมายเลขโทรศัพท์ (Telephone Number) ถูกแยกออกเป็นคำย่อยหลายคำ วิธีการปรับปรุงแก้ไข คือ
 - เนื่องจากไม่มีการเก็บฐานข้อมูลคำศัพท์ภาษาต่างประเทศ การรวมคำประเภทนี้จะมองในลักษณะเดียวกับปัญหาในเรื่องคำนามที่เป็นกลุ่มคำ คืออาศัยไวยากรณ์เข้ามาช่วยในการรวมคำ (หัวข้อ 3.2.3.4) ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป และการแก้ไขหน้าที่ของคำ

- 4) ปัญหาในเรื่องคำนามที่เป็นกลุ่มคำ (Compound Noun) ไม่สามารถตัดคำให้ได้ ผลลัพธ์ที่ถูกต้อง กลุ่มคำเหล่านั้นถูกตัดแยกเป็นคำนามหลาย ๆ คำ และมีการกำกับหน้าที่คำเหล่านั้นแยกแต่ละคำ ทำให้เมื่อมีการวิเคราะห์ไวยากรณ์ของประโยค ประโยคที่ถูกต้อง กลายเป็นประโยคที่มีโครงสร้างไม่ตรงตามหลักไวยากรณ์ มีวิธีการปรับปรุงแก้ไข คือ
- จะแก้ไขในขั้นตอนการประมวลผลขั้นต่อไป ซึ่งอาศัยไวยากรณ์เข้ามาช่วยในการรวมคำที่ควรเป็นคำเดียวกันกลับเข้าด้วยกัน และการแก้ไขหน้าที่ของคำ จะทำหลังจากการรวมคำแล้ว ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป (หัวข้อ 3.2.3.4)
- 5) ปัญหาในเรื่องคำกริยาที่เป็นกลุ่มคำ (Compound Verb) ไม่สามารถตัดคำให้ได้ ผลลัพธ์ที่ถูกต้อง กลุ่มคำเหล่านั้นถูกตัดแยกเป็นคำกริยาหลาย ๆ คำ และมีการกำกับหน้าที่คำเหล่านั้นแยกแต่ละคำ ทำให้เกิดปัญหาคัดค้านกับคำนามที่เป็นกลุ่มคำคือ ประโยคมีโครงสร้างไม่ตรงตามหลักไวยากรณ์ มีวิธีการปรับปรุงแก้ไข คือ
- จะแก้ไขในขั้นตอนการประมวลผลขั้นต่อไป ซึ่งอาศัยไวยากรณ์เข้ามาช่วยในการรวมคำที่ควรเป็นคำเดียวกันกลับเข้าด้วยกัน และการแก้ไขหน้าที่ของคำ จะทำหลังจากการรวมคำแล้ว ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป (หัวข้อ 3.2.3.4)
- 6) ปัญหาในเรื่องคำนามที่เกิดจาก คำกริยาที่นำหน้าด้วย คำว่า “การ” หรือ “ความ” โปรแกรมจะตัดแยกคำนามคำเดิมออกมาเป็น 2 ส่วน คือ ส่วนของคำนำหน้า (“การ” หรือ “ความ”) โดยกำกับหน้าที่เป็นคำนำหน้าคำนาม (Nominal Prefix) และส่วนคำกริยาที่ตามหลัง โดยกำกับหน้าที่เป็นคำกริยา ผลลัพธ์ที่เกิดขึ้นทำให้คำนามคำเดิมหายไป และประโยคที่ได้ไม่ตรงตามหลักไวยากรณ์ มีวิธีการปรับปรุงแก้ไข คือ
- จะแก้ไขในขั้นตอนการประมวลผลขั้นต่อไป ซึ่งอาศัยไวยากรณ์เข้ามาช่วยในการรวมคำที่ควรเป็นคำเดียวกันกลับเข้าด้วยกัน และการแก้ไขหน้าที่ของคำ จะทำหลังจากการรวมคำแล้ว ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป (หัวข้อ 3.2.3.4)
- 7) ปัญหาในเรื่องคำนามที่เกิดจาก คำกริยาที่นำหน้าด้วย คำว่า “ผู้” โปรแกรมจะตัดแยกคำนามคำเดิมออกมาเป็น 2 ส่วน คือ ส่วนของคำนำหน้า (“ผู้”) โดยกำกับหน้าที่เป็นคำบุรุษสรรพนาม (Personal Pronoun) และส่วนคำกริยาที่ตามหลัง โดยกำกับหน้าที่เป็นคำกริยา มีวิธีการปรับปรุงแก้ไข คือ

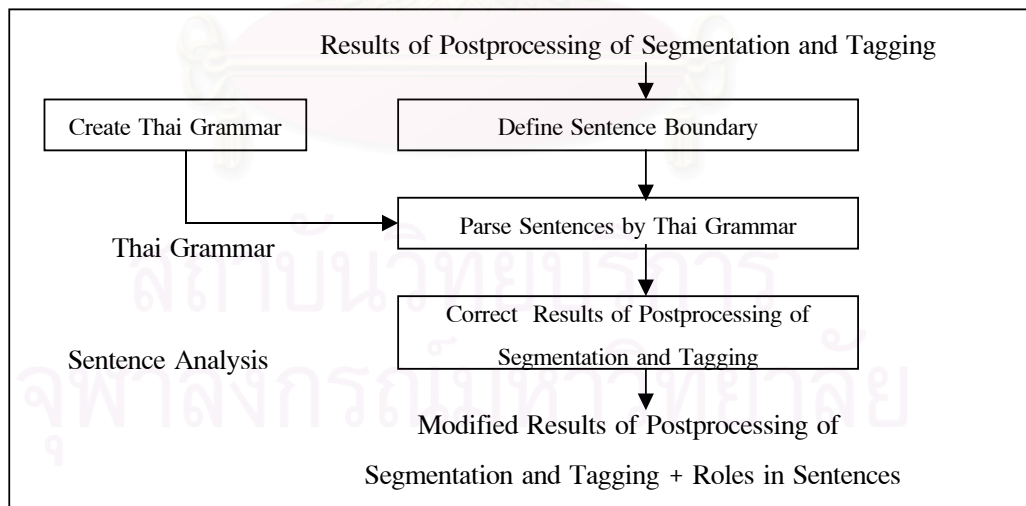
- จะแก้ไขในขั้นตอนการประมวลผลขั้นต่อไป ซึ่งอาศัยไวยากรณ์เข้ามาช่วยในการรวมคำที่ควรเป็นคำเดียวกันกลับเข้าด้วยกัน และการแก้ไขหน้าทีของคำ จะทำหลังจากการรวมคำแล้ว ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป (หัวข้อ 3.2.3.4)
- 8) ปัญหาในเรื่องลักษณะคำไทยบางคำ ที่เขียนเหมือนกันแต่มีหลายความหมายและทำหน้าที่ต่างกัน ตัวอย่างเช่น “ที่ทำงาน” อาจหมายถึงสถานที่ทำงาน หรือคำว่า “ที่” เป็นคำแทนคำอื่นบอกว่าสิ่งนั้นทำงาน โปรแกรมตัดคำมักจะตัดคำว่าที่ทำงานในความหมายแรกกลายเป็นความหมายที่สอง มีวิธีการปรับปรุงแก้ไข คือ
- จะแก้ไขในขั้นตอนการประมวลผลขั้นต่อไป ซึ่งอาศัยไวยากรณ์เข้ามาช่วยในการรวมคำที่ควรเป็นคำเดียวกันกลับเข้าด้วยกัน และการแก้ไขหน้าทีของคำ จะทำหลังจากการรวมคำแล้ว ซึ่งจะกล่าวถึงในการประมวลผลขั้นถัดไป (หัวข้อ 3.2.3.4)
- 9) ปัญหาในเรื่องการกำกับหน้าที่คำของเครื่องหมายวรรคตอนที่ได้พบได้ในภาษาไทยหรือเครื่องหมายพิเศษอื่น ๆ เช่น วงเล็บ (“(”, “)”) ยัติภังค์ (“-”) โปรแกรมตัดคำจะกำกับหน้าที่เป็นคำวิสามานยนาม ไม่มีการแยกหน้าที่ออกมาว่าเป็นเครื่องหมายวรรคตอน มีวิธีการปรับปรุงแก้ไข คือ
- ตรวจสอบผลลัพธ์ที่ได้จากโปรแกรมการตัดคำว่าคำนามที่ได้ คือเครื่องหมายวรรคตอนหรือไม่ และทำการแก้ไขหน้าที่คำให้กลายเป็นเครื่องหมายวรรคตอน
 - ตรวจสอบผลลัพธ์ที่ได้จากโปรแกรมการตัดคำว่าคำนามที่ได้ คือเครื่องหมายยัติภังค์ ที่ใช้การละข้อความว่าไม่มีรายละเอียดของข้อมูลหรือไม่ ตัวอย่างเช่น การเขียนความต้องการด้านหน้าที่อาจไม่มีส่วนบพหน้า สามารถใช้เครื่องหมาย “-” เพื่อแสดงว่าไม่มีข้อมูลในส่วนนั้น ทำการแก้ไขการกำกับหน้าที่คำให้กลายเป็น “แดช (DASH)” (สร้างหน้าที่คำใหม่ขึ้นมา ไม่กำกับหน้าที่ของคำตามหน้าที่คำในคลังข้อความ)
- 10) ปัญหาในเรื่องการกำกับหน้าที่คำของคำนาม โปรแกรมตัดคำจะกำกับหน้าที่คำของคำนามได้ไม่ชัดเจนว่าเป็นคำนามประเภทใด โดยผลลัพธ์ที่ได้จากการกำกับหน้าที่คำของโปรแกรมตัดคำมักจะกำกับหน้าที่คำนามออกมาเป็นวิสามานยนามเสมอ ปัญหานี้ยังไม่มีวิธีการปรับปรุงแก้ไข เนื่องจากฐานข้อมูลของคำนามที่งานวิจัยนี้สร้างขึ้นยังไม่มีการกำกับหน้าที่คำไว้ ดังนั้นมีผลต่อเนื่องทำให้การประมวลผลในขั้นการสร้างโครงสร้างใหม่ จะไม่มีการให้ความสำคัญกับประเภทของคำนาม โดยจะพิจารณาคำนามทุกประเภท ไม่มีการแยกแยะประเภทคำนามที่ย่อยลงไป

3.2.3 การวิเคราะห์ประโยค

การวิเคราะห์ประโยค เพื่อหาคำหรือวลีได้ในประโยคทำหน้าที่เป็นส่วนสำคัญของประโยค เช่น คำนามหรือนามวลีที่เป็นส่วนประธานหรือส่วนกรรมของประโยค คำกริยาหรือกริยา วลีที่เป็นส่วนกริยา คำหรือวลีที่ทำหน้าที่เพียงแคเป็นส่วนขยายของประโยค เช่น คำคุณศัพท์ คำกริยาวิเศษณ์ คุณศัพท์วลี กริยาวิเศษณ์วลี หรือทำหน้าที่เป็นคำเชื่อม เช่น คำสันธาน คำบุพบท ซึ่งจะทำให้ได้ใจความของประโยคออกมา รวมถึงพิจารณาลักษณะการใช้รูปแบบโครงสร้างประโยคของภาษาธรรมชาติว่าตรงตามหลักไวยากรณ์ไทยหรือไม่

ผลลัพธ์ในส่วนนี้ จะเห็นโครงสร้างประโยคทั้งหมดในเอกสาร โครงสร้างของวลี หน้าที่ของคำและวลีต่าง ๆ ในประโยค เพื่อเป็นข้อมูลในการตัดสินใจเลือกคำหรือวลี ที่แสดงถึงส่วนสำคัญของความต้องการด้านหน้าที่ แต่หากเอกสารนั้นมีประโยคที่ไม่ตรงตามหลักไวยากรณ์จะมีข้อความแสดงผลว่าโครงสร้างประโยคผิดพลาด และไม่สามารถทำงานต่อไปได้

การประมวลผลในขั้นนี้จะนำผลลัพธ์ที่ได้จากการตัดคำและกำกับหน้าคำที่แก้ไขเรียบร้อยแล้ว มาวิเคราะห์ส่วนต่าง ๆ ของประโยคโดยใช้หลักไวยากรณ์เข้ามาช่วย แบ่งการทำงานเป็น 4 ส่วน (แสดงดังรูปที่ 3.3) คือ การสร้างกฎไวยากรณ์ภาษาไทย การกำหนดขอบเขตประโยค การวิเคราะห์ประโยคด้วยไวยากรณ์ภาษาไทย และการรวมคำและแก้ไขหน้าที่คำโดยใช้ไวยากรณ์เข้ามาช่วย



รูปที่ 3.3 ขั้นตอนการวิเคราะห์ประโยค

3.2.3.1 การสร้างกฎไวยากรณ์ภาษาไทย

ในส่วนนี้มีการสร้างกฎไวยากรณ์ภาษาไทย สำหรับนำไปวิเคราะห์หาส่วนต่าง ๆ ของประโยค ไวยากรณ์ที่งานวิจัยนี้สร้างและปรับปรุงประยุกต์มาจากงานวิจัยที่เกี่ยวข้อง 2.2.5 [15] โดยมีการปรับปรุงให้มีความซับซ้อนน้อยลง และเพิ่มเติมบางส่วน เพื่อให้เหมาะสมกับโครงสร้างประโยคของการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ได้แก่

- 1) การลดกฎไวยากรณ์ เพื่อให้ไวยากรณ์มีความง่ายขึ้น มี 2 ลักษณะ คือ
 - การลดกฎไวยากรณ์เดิมที่ไม่พบในลักษณะการเขียนเอกสารออกไป
 - การลดหน้าที่ของคำที่นำมาวิเคราะห์ให้น้อยลง โดยการตัดหน้าที่คำที่ไม่เกี่ยวข้องกับการเขียนเอกสาร เช่น คำอุทาน คำลงท้ายประโยคเพื่อตอบรับ คำลงท้ายประโยคเพื่อถามความคิดเห็น เนื่องจากคำเหล่านี้จะไม่พบในเอกสาร และตัดกฎไวยากรณ์ที่เกี่ยวข้องกับหน้าที่คำเหล่านั้นออกไป
- 2) การรวมกฎไวยากรณ์เดิม โดยลดความสำคัญของหน้าที่คำที่ไม่ใช่ส่วนสำคัญในการอธิบายความต้องการด้านหน้าที่ของซอฟต์แวร์ เช่น ไม่ว่าจะป็นกริยาช่วยที่นำหน้าหรือตามหลัง ก็รวมถือเป็นคำประเภทกริยาช่วยทั้งหมด
- 3) การเพิ่มกฎไวยากรณ์ เพื่อให้ไวยากรณ์มีความยืดหยุ่น เหมาะกับลักษณะการเขียนเอกสาร เช่น การละประธานของประโยค การละคำบุพบท

ไวยากรณ์ที่สร้างและปรับปรุงขึ้นนี้ นอกจากทำการเพิ่ม ลด เปลี่ยนแปลงกฎไวยากรณ์แล้ว ยังมีการเปลี่ยนแปลงลักษณะการเขียนกฎไวยากรณ์ เพื่อให้ตรงตามความต้องการ และสามารถทำงานได้กับขั้นตอนการวิเคราะห์ประโยคในงานวิจัยนี้

ลักษณะไวยากรณ์ (รายละเอียดในภาคผนวก ง.) ที่สร้างและปรับปรุงขึ้นใหม่ คือ

- 1) ไวยากรณ์ที่สร้างขึ้น เป็นลักษณะไวยากรณ์การแจقس่วนจากล่างขึ้นบน (Bottom-up) เพราะเหมาะกับลักษณะการทำงานที่นำผลลัพธ์ที่ได้จากการตัดคำ ซึ่งเกิดจากการตัดข้อความในเอกสารออกมาเป็นคำ ๆ แล้วนำมารวมกันใหม่ เพื่อสร้างเป็นโครงสร้างประโยค
- 2) การจัดลำดับกฎไวยากรณ์ จะเรียงลำดับกฎไวยากรณ์ที่มีส่วนประกอบของประโยคยาวไว้ก่อนกฎไวยากรณ์ที่มีส่วนประกอบของประโยคสั้น เพื่อให้สามารถสร้างประโยคที่เป็นโครงสร้างที่สมบูรณ์ขึ้นมาได้ เนื่องจากลำดับของไวยากรณ์มีผลต่อผลลัพธ์ที่ได้ ซึ่งก็คือโครงสร้างประโยคและความเร็วของการวิเคราะห์ประโยค ถ้าจัดลำดับกฎไวยากรณ์ที่มีส่วนประกอบของประโยคสั้นไว้ก่อน เมื่อทำการวิเคราะห์หา

ส่วนประกอบของประโยค อาจได้ผลลัพธ์ที่เป็นประโยคที่ไม่สมบูรณ์ตามโครงสร้างประโยคจริง หรือต้องเสียเวลาส่วนหนึ่งในการเปลี่ยนกฎไวยากรณ์ที่เลือก เพื่อให้ได้โครงสร้างประโยคที่สมบูรณ์

- 3) ลำดับและประเภทของสัญลักษณ์ (Symbol) ในแต่ละกฎไวยากรณ์ กฎไวยากรณ์แต่ละกฎจะเขียนสัญลักษณ์ปลายทาง (Terminal Symbol) อยู่เป็นลำดับแรกในแต่ละกฎ เนื่องจากลำดับและประเภทของสัญลักษณ์ในกฎไวยากรณ์ มีผลต่อความเร็วในการวิเคราะห์ประโยค เนื่องจากถ้าพบสัญลักษณ์ที่ไม่ใช่สัญลักษณ์ปลายทาง (Non-terminal Symbol) การทำงานต้องวิเคราะห์กฎไวยากรณ์อื่น ๆ ต่อไปจนกระทั่งพบสัญลักษณ์ปลายทาง แล้วจึงกลับมาวิเคราะห์สัญลักษณ์ถัดไปในกฎเดิมต่อ ทำให้ใช้เวลาในการประมวลผลมากขึ้น
- 4) จำนวนของสัญลักษณ์ที่ไม่ใช่สัญลักษณ์ปลายทาง จะมีจำนวนน้อยที่สุด เนื่องจากจำนวนของสัญลักษณ์ที่ไม่ใช่สัญลักษณ์ปลายทางมีผลต่อความเร็วในการวิเคราะห์ไวยากรณ์ ยิ่งมีสัญลักษณ์ที่ไม่ใช่สัญลักษณ์ปลายทางมาก การวิเคราะห์ไวยากรณ์จะใช้เวลาเพิ่มขึ้น เพราะต้องเสียเวลาในการวิเคราะห์กฎไวยากรณ์อื่น ๆ ต่อไป จนกระทั่งพบสัญลักษณ์ปลายทาง

3.2.3.2 การกำหนดขอบเขตของประโยค

ผลลัพธ์ที่ได้จากการตัดคำและการกำกับหน้าที่คำก่อนเข้าสู่ขั้นตอนการวิเคราะห์ประโยคด้วยไวยากรณ์ภาษาไทยนั้น จะต้องมีการหาขอบเขตของประโยคที่แท้จริง ซึ่งในงานวิจัยนี้ขอบเขตของประโยคจะกำหนดตามเว้นวรรคระหว่างประโยคในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นข้อมูลนำเข้า ซึ่งเมื่อเอกสารที่เป็นข้อมูลนำเข้าผ่านการตัดคำและกำกับหน้าที่ของคำแล้ว ส่วนที่เป็นการเว้นวรรคระหว่างประโยคและการเว้นวรรคระหว่างคำจะมองเห็นเป็นอักขระการเว้นวรรคเหมือนกัน รวมถึงเครื่องหมายวรรคตอนต่าง ๆ ซึ่งอาจเป็นเครื่องหมายวรรคตอนที่อยู่ระหว่างคำ หรือระหว่างประโยคก็ได้ ดังนั้นต้องมีขั้นตอนการหาขอบเขตประโยคที่แท้จริง โดยการกำจัดเว้นวรรคหรือเครื่องหมายวรรคตอนที่ไม่ใช่จุดสิ้นสุดของประโยคออกไป

การเว้นวรรคหรือเครื่องหมายวรรคตอนที่ใช้ในการเขียนเอกสาร ซึ่งไม่ใช่จุดสิ้นสุดของประโยค ประกอบด้วย

- 1) เว้นวรรคหรือเครื่องหมายวรรคตอนที่อยู่ระหว่างคำกับตัวเลข ใช้การตรวจสอบจากตัวอักขระที่อยู่หน้าและหลังเว้นวรรคหรือเครื่องหมายวรรคต่อนั้น
- 2) เว้นวรรคหรือเครื่องหมายวรรคตอนที่อยู่ระหว่างคำภาษาอังกฤษ ใช้การตรวจสอบจากตัวอักขระที่อยู่หน้าและหลังเว้นวรรคหรือเครื่องหมายวรรคต่อนั้น

- 3) เว้นวรรคระหว่างคำภาษาไทยกับคำภาษาอังกฤษที่ขึ้นต้นด้วยอักษรตัวเล็ก แสดงว่าคำภาษาอังกฤษนั้นอยู่ภายในประโยค ใช้การตรวจสอบจากตัวอักษรที่อยู่หน้าและหลังเว้นวรรคนั้น
- 4) เว้นวรรคระหว่างคำภาษาไทยกับคำภาษาอังกฤษที่ขึ้นต้นด้วยอักษรตัวใหญ่ ใช้การตรวจสอบจากตัวอักษรที่อยู่หน้าและหลังเว้นวรรคนั้น และระยะห่างจากคำนั้นไปยังคำกริยาในประโยค เนื่องจากคำภาษาอังกฤษที่ขึ้นต้นด้วยอักษรตัวใหญ่ อาจเป็นการขึ้นต้นประโยคที่แท้จริง หรือคำภาษาอังกฤษนั้นเป็นชื่อเฉพาะจึงเขียนขึ้นต้นด้วยอักษรตัวใหญ่เสมอ ดังนั้นจะใช้การคำนวณระยะห่างจากคำกริยาเข้ามาตัดสินด้วยว่าเป็นเว้นวรรคระหว่างประโยคหรือไม่ โดยถ้าเว้นวรรคนั้นอยู่หลังคำภาษาอังกฤษแต่หน้าคำภาษาไทย และคำภาษาอังกฤษนั้นอยู่ใกล้คำกริยาตัวหลังแสดงว่าคำภาษาอังกฤษเป็นคำขึ้นต้นประโยคใหม่ หรือถ้าเว้นวรรคนั้นอยู่หลังคำภาษาไทยแต่หน้าคำภาษาอังกฤษ และคำภาษาอังกฤษนั้นอยู่ใกล้คำกริยาตัวหน้าแสดงว่าคำภาษาอังกฤษไม่ใช่คำขึ้นต้นประโยคใหม่
- 5) เว้นวรรคที่อยู่ระหว่างเครื่องหมายวรรคตอนกับตัวอักษรหรือตัวเลข ใช้การตรวจสอบจากตัวอักษรที่อยู่หน้าและหลังเว้นวรรคนั้น

นอกเหนือจากการใช้เว้นวรรคระหว่างประโยคที่เขียนในเอกสารที่เป็นข้อมูลนำเข้าเป็นขอบเขตของประโยคแล้ว อาจมีการกำจัดเว้นวรรคที่มากับเอกสารที่เป็นข้อมูลนำเข้านั้นออกไป ถ้าลักษณะการเขียนเอกสารมีการเว้นวรรคในส่วนที่ไม่ใช่จุดสิ้นสุดของประโยคที่แท้จริง ได้แก่

- 1) เว้นวรรคที่อยู่ระหว่างเครื่องหมายวรรคตอนกับตัวอักษรหรือตัวเลข ใช้การตรวจสอบจากตัวอักษรที่อยู่หน้าและหลังเว้นวรรคนั้น
- 2) เว้นวรรคระหว่างสันธานวลีกับประโยคเดิมที่วลีนั้นทำหน้าที่ขยาย จะใช้การตรวจสอบจากหน้าที่คำที่อยู่หน้าเว้นวรรคนั้น

หากลักษณะการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นข้อมูลนำเข้ามีลักษณะการเว้นวรรคที่ไม่ถูกต้อง เช่น การเว้นวรรคในส่วนที่ไม่ใช่จุดสิ้นสุดของประโยคที่แท้จริง หรือไม่มีการเว้นวรรคเมื่อเป็นจุดสิ้นสุดประโยค จะใช้การแก้ปัญหาโดยให้ลักษณะไวยากรณ์ที่สร้างและปรับปรุงขึ้นในหัวข้อ 3.2.3.1 นั้นมีความยืดหยุ่นเพียงพอในการรองรับลักษณะประโยคดังกล่าว แต่ถ้ามีลักษณะการเว้นวรรคประโยคที่ไม่ถูกต้องมากเกินไป ในขั้นตอนการวิเคราะห์ประโยค จะไม่สามารถทำงานต่อไปได้ เนื่องจากผลลัพธ์ของการทำงานจะแสดงผลว่าประโยคในการเขียนเอกสารไม่ถูกต้องตามกฎไวยากรณ์

3.2.3.3 การวิเคราะห์ส่วนประกอบของประโยคด้วยไวยากรณ์ภาษาไทย

ในส่วนนี้เป็นการวิเคราะห์ส่วนประกอบของประโยค เพื่อให้ทราบว่าประโยคประกอบด้วยวลีหรือคำประเภทใดบ้าง โดยใช้ไวยากรณ์เข้ามาช่วยในการวิเคราะห์ รวมทั้งทำให้ทราบว่าวลีหรือคำในประโยคเหล่านั้นทำหน้าที่ใดของประโยค โดยใช้ลำดับในประโยคและประเภทของวลีหรือคำที่ได้จากการวิเคราะห์ด้วยไวยากรณ์เป็นตัวพิจารณา

การประมวลผลในขั้นตอนนี้ใช้ไวยากรณ์ที่สร้างและปรับปรุงขึ้นในหัวข้อ 3.2.3.1 เป็นตัววิเคราะห์กับหน้าที่ของคำที่ปรับปรุงแก้ไขแล้วจากการกำกับคำ และเมื่อได้ผลลัพธ์ของโครงสร้างประโยค จะทำการเชื่อมโยงจากหน้าที่คำกลับไปยังคำเดิม ทำให้ทราบว่าคำแต่ละคำเป็นส่วนประกอบอยู่ในวลีใดของประโยค และทำหน้าที่เป็นส่วนใดของประโยค

3.2.3.4 การรวมคำและแก้ไขหน้าที่คำโดยใช้ไวยากรณ์

ในส่วนนี้จะเป็นการปรับปรุงแก้ไขผลลัพธ์ของการตัดคำและกำกับหน้าที่คำ โดยใช้ข้อมูลจากการวิเคราะห์ส่วนประกอบของประโยคด้วยไวยากรณ์เข้ามาช่วย

การรวมคำนามที่เป็นกลุ่มคำและแก้ไขหน้าที่คำ มีรายละเอียดดังต่อไปนี้

- 1) เป็นคำที่พบในฐานข้อมูลคำศัพท์ โดยใช้การเปรียบเทียบแบบรูป (Pattern Matching)
- 2) ถ้าไม่ใช่คำที่มีในฐานข้อมูลคำศัพท์ จะสร้างเงื่อนไขในการพิจารณา ว่าควรจะมีการรวมคำย่อยหลายคำให้เป็นคำเดียวกันหรือไม่ เงื่อนไขที่นำมาพิจารณา คือ
 - 2.1) คำย่อยหลาย ๆ คำที่เรียงมาติดกันนั้น ไม่มีเครื่องหมายวรรคตอนหรือการเว้นวรรคระหว่างคำย่อยเหล่านั้น
 - 2.2) เป็นไปตามเงื่อนไข 2.1) และมีหน้าที่ของคำเป็นคำนามประเภทเดียวกันทั้งหมด
 - เป็นส่วนประกอบในนามวลีซึ่งเป็นวลีเดียวกัน จะถือว่าเป็นคำนามที่เป็นกลุ่มคำ ตัวอย่างคำเช่น หมายเลขบัตรเครดิต
 - เป็นส่วนประกอบของนามวลีหลายวลีต่อกัน จะใช้คำกริยาหรือกริยาวลีที่อยู่ใกล้เคียงช่วยในการพิจารณาว่าเป็นคำนามที่เป็นกลุ่มคำหรือไม่
 - 2.3) เป็นไปตามเงื่อนไข 2.1) และมีหน้าที่ของคำคำแรกเป็นคำนามตามมาด้วยคำที่มีหน้าที่คำเป็นคำกริยาเพียงคำเดียว และตามด้วยคำที่มีหน้าที่เป็นคำนามประเภทเดียวกับคำนามแรก และเป็นส่วนประกอบในนามวลีซึ่งเป็นวลีเดียวกัน จะถือว่าเป็นคำนามที่เป็นกลุ่มคำ ตัวอย่างคำเช่น ไบส่งสินค้า

- 2.4) เป็นไปตามเงื่อนไข 2.1) และมีหน้าที่ของคำคำแรกเป็นคำบุพบทตามมาด้วย คำที่มีหน้าที่คำเป็นคำกริยาเพียงคำเดียว จะถือว่าเป็นคำนามที่เป็นกลุ่มคำ ตัวอย่างคำเช่น ที่ทำงาน
- 3) หลังจากรวมกลุ่มคำแล้ว จะมีการกำหนดหน้าที่คำใหม่ ให้ตรงกับหน้าที่ของคำใหม่ที่เกิดขึ้น ซึ่งส่วนใหญ่จะกำกับหน้าที่ตามคำนามคำแรกที่เป็นคำขึ้นต้นกลุ่มคำ

การรวมคำกริยาที่เป็นกลุ่มคำและแก้ไขหน้าที่คำ มีรายละเอียดดังต่อไปนี้

- 1) เป็นคำที่พบในฐานข้อมูลคำศัพท์ โดยใช้การเปรียบเทียบแบบรูป (Pattern Matching)
- 2) ถ้าไม่ใช่คำที่มีในฐานข้อมูลคำศัพท์ จะสร้างเงื่อนไขในการพิจารณา ว่าควรจะทำกรรวมคำย่อยหลายคำให้เป็นคำเดียวกันหรือไม่ เงื่อนไขที่นำมาพิจารณา คือ
 - 2.1) คำย่อยหลาย ๆ คำที่เรียงมาติดกันนั้น ไม่มีเครื่องหมายวรรคตอนหรือการเว้นวรรคระหว่างคำย่อยเหล่านั้น
 - 2.2) เป็นไปตามเงื่อนไข 2.1) และมีหน้าที่ของคำเป็นคำกริยาหรือคำกริยาช่วย และเป็นส่วนประกอบในกริยวลีซึ่งเป็นวลีเดียวกัน จะถือว่าเป็นคำกริยาที่เป็นกลุ่มคำ
- 3) หลังจากรวมกลุ่มคำแล้ว จะมีการกำหนดหน้าที่คำใหม่ ให้ตรงกับหน้าที่ของคำใหม่ที่เกิดขึ้น โดยปัญหาในการพิจารณาหน้าที่คำคือ กลุ่มคำกริยาใหม่นี้เป็นได้ทั้งคำกริยาที่แสดงการกระทำ (Active Verb) หรือคำกริยาที่แสดงสถานะ (Stative Verb) หรืออาจเป็นแค่กริยาช่วยก็ได้ ดังนั้นจึงต้องพิจารณาจากลักษณะเฉพาะของภาษาไทย และลักษณะเฉพาะของคำ เพื่อกำหนดหน้าที่คำใหม่ให้ถูกต้อง

การรวมคำกริยาที่ขึ้นต้นด้วยคำว่า “การ” หรือ “ความ” มีรายละเอียดดังต่อไปนี้

- 1) ใช้หลักของแบบรูป โดยตรวจสอบว่าเป็นคำว่า “การ” หรือ “ความ” ขึ้นต้น และตามมาด้วยคำกริยา
 - 1.1) ขึ้นต้นด้วยคำว่า “ความ” จากการพิจารณาตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์พบว่า กริยาที่ตามหลังคำว่าความ ส่วนใหญ่จะไม่ใช่วลีที่แสดงการกระทำ แต่เป็นกริยาที่แสดงสถานะ ดังนั้นจะเปลี่ยนหน้าที่คำเป็นคำกริยาที่แสดงสถานะ เพราะไม่ได้เป็นส่วนสำคัญที่แสดงถึงความต้องการด้านหน้าที่ของซอฟต์แวร์

- 1.2) ขึ้นต้นด้วยคำว่า “การ” จากการพิจารณาตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์พบว่า กริยาที่ตามหลังคำว่ากร บางครั้งเป็นส่วนสำคัญของความต้องการด้านหน้าที่จึงคงหน้าที่คำเป็นคำกริยาที่แสดงการกระทำไว้
- 2) ใช้การสร้างไวยากรณ์ในหัวข้อ 3.2.3.1 ให้รองรับว่า คำนามสามารถเกิดจากการรวมกันระหว่าง คำที่มีหน้าที่เป็นคำนำหน้าคำนามกับคำกริยาได้
- 3) ไม่มีการแก้ไขหน้าที่คำที่รวมแล้ว เนื่องจากคำกริยาอาจเป็นส่วนสำคัญของความต้องการด้านหน้าที่ จึงยังคงหน้าที่คำคำนั้นเป็นคำกริยาแสดงการกระทำไว้

การรวมคำกริยาที่ขึ้นต้นด้วยคำว่า “ผู้” มีรายละเอียดดังต่อไปนี้

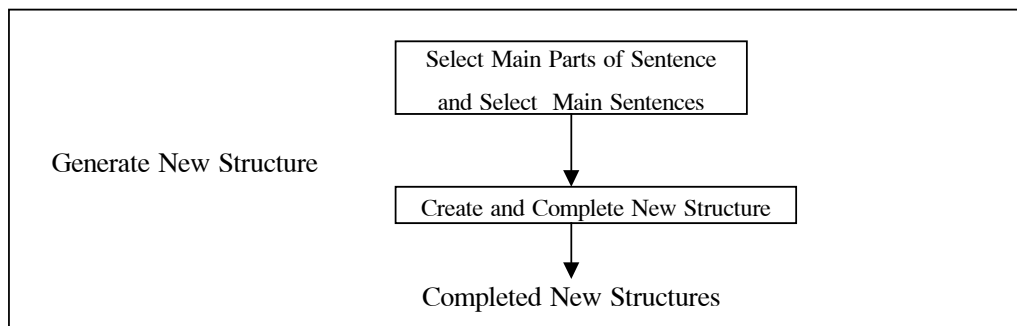
- 1) ใช้หลักของแบบรูป โดยตรวจสอบว่าเป็นคำว่า “ผู้” ขึ้นต้น และตามมาด้วยคำกริยา
- 2) ใช้การสร้างไวยากรณ์ในหัวข้อ 3.2.3.1 ให้รองรับว่า คำนามสามารถเกิดจากการรวมกันระหว่าง คำที่มีหน้าที่เป็นบุรุษสรรพนามกับคำกริยาได้
- 3) แก้ไขหน้าที่คำที่รวมแล้วเป็นคำนาม

3.2.4 การสร้างรูปแบบโครงสร้างใหม่

รูปแบบโครงสร้างใหม่ที่สร้างขึ้นในงานวิจัยนี้ สำหรับการอธิบายความต้องการด้านหน้าที่แทนภาษาธรรมชาติ มีแนวคิดมาจากทฤษฎีที่เกี่ยวข้อง 2.1.4 [13] ซึ่งเป็นการอธิบายฟังก์ชันการทำงานของซอฟต์แวร์ด้วยภาษาธรรมชาติ แต่มีการแยกเป็นส่วนต่าง ๆ คือ ข้อมูลนำเข้า แหล่งที่มา ข้อมูลนำออก แหล่งปลายทาง และขั้นตอนการทำงาน ประยุกต์เข้ากับงานวิจัยที่เกี่ยวข้องหัวข้อ 2.2.3 [8] ที่มีการพิจารณาประธาน กรรม การกระทำจากข้อความในประโยค และงานวิจัยที่เกี่ยวข้องหัวข้อ 2.2.4 [9] ที่มีการสร้างเศษความหมายจากกริยาที่แสดงการทำงานในประโยค จนได้โครงสร้างที่แสดงให้เห็นหน้าที่การทำงานของซอฟต์แวร์อย่างชัดเจน ประกอบด้วยส่วนสำคัญ 6 ส่วน คือ ยูสเคส แหล่งที่มา ข้อมูลนำเข้า การทำงาน ข้อมูลนำออก และ แหล่งปลายทาง

ขั้นตอนในการสร้างโครงสร้างใหม่ ประกอบด้วยการทำงาน 2 ขั้นตอน แสดงดังรูปที่ 3.4 คือ ขั้นตอนแรกเป็นการเลือกส่วนสำคัญของประโยคและประโยคที่แสดงถึงฟังก์ชันการทำงานของความต้องการด้านหน้าที่เพื่อนำมาสร้างเป็นโครงสร้างใหม่ และขั้นตอนที่สองเป็นการสร้างโครงสร้างใหม่และเพิ่มเติมข้อมูลเพื่อให้โครงสร้างใหม่มีความสมบูรณ์มากขึ้น

ผลลัพธ์สุดท้ายที่ได้จากการประมวลผลในขั้นนี้ คือ ความต้องการด้านหน้าที่ทุกหน้าที่ถูกอธิบายแทนด้วยโครงสร้างใหม่ ที่แสดงให้เห็นถึงฟังก์ชันการทำงานที่ชัดเจนของซอฟต์แวร์ โดยแต่ละฟังก์ชันแสดงส่วนสำคัญทั้ง 6 ส่วนครบถ้วนตามที่กล่าวข้างต้น ความต้องการด้านหน้าที่หนึ่งหน้าที่นี้อาจอธิบายแทนด้วยโครงสร้างใหม่ que แสดงถึงฟังก์ชันการทำงานได้มากกว่าหนึ่งฟังก์ชัน



รูปที่ 3.4 ขั้นตอนการสร้างโครงสร้างใหม่

3.2.4.1 การเลือกส่วนสำคัญของประโยคและประโยคที่จะนำไปสร้างเป็นโครงสร้างใหม่

ในส่วนนี้จะทำการพิจารณาผลลัพธ์ที่ได้จากการประมวลผลขั้นต่าง ๆ ก่อนหน้าขั้นตอนนี้ทั้งหมด เพื่อให้การหาคำสำคัญ สำหรับการสร้างโครงสร้างใหม่ที่ใช้อธิบายแทนความต้องการด้านหน้าที่ของซอฟต์แวร์ มีความถูกต้อง ครบถ้วนให้มากที่สุด ส่วนประกอบของประโยคที่ควรให้ความสำคัญ ได้แก่ ส่วนของกริยาวลี นามวลี และบุพบทวลี เพราะส่วนเหล่านี้มักจะบอกถึงการทำงานของความต้องการด้านหน้าที่ ข้อมูลที่เกี่ยวข้องกับการทำงานนั้น และแหล่งที่มาหรือแหล่งปลายทางของข้อมูล ตามลำดับ เริ่มต้นด้วยการพิจารณาคำกริยาที่ขึ้นต้นกริยาวลี ซึ่งเป็นคำที่แสดงการกระทำเป็นอันดับแรก เพราะความต้องการด้านหน้าที่จะแสดงหน้าที่การทำงานของซอฟต์แวร์ ถัดจากนั้นจึงมาพิจารณาคำนามที่เกี่ยวข้องกับคำกริยานั้น ซึ่งคาดว่าอาจเป็นข้อมูลนำเข้าหรือข้อมูลนำออก สุดท้ายจะพิจารณาคำบุพบท คู่กับคำนามที่เกี่ยวข้องกับบุพบทนั้น เพราะอาจจะแสดงถึงแหล่งที่มาหรือแหล่งปลายทางของข้อมูล

การพิจารณาคำกริยา ว่าเป็นกริยาที่แสดงการทำงานของความต้องการด้านหน้าที่หรือไม่ พิจารณาดังต่อไปนี้

- 1) คำกริยานั้นเป็นคำกริยาแสดงการกระทำ และเป็นกริยาแท้ในประโยค
- 2) คำกริยานั้นพบว่าเป็นคำที่ซ้ำกันในหลายส่วนของความต้องการด้านหน้าที่ เช่น พบทั้งในส่วนคำนำ และส่วนกระบวนการ
- 3) ให้ความสำคัญกับคำกริยาที่พบในส่วนกระบวนการ มากกว่าคำกริยาที่พบในส่วนข้อมูลนำเข้า และข้อมูลนำออก (กำหนดค่าน้ำหนัก (Weight) ของแต่ละคำที่พบในแต่ละส่วนต่างกัน)

การพิจารณาคำนาม ว่าเป็นคำนามที่เป็นข้อมูล ซึ่งเกี่ยวข้องกับการทำงานของความต้องการด้านหน้าที่หรือไม่ พิจารณาดังต่อไปนี้

- 1) เป็นคำนามที่ตามหลังคำกริยาที่แสดงกระบวนการการทำงานของความต้องการด้านหน้าที่ เพราะมักจะเป็นข้อมูลที่เกี่ยวข้องกับการทำงานนั้น

- 2) ให้ความสำคัญกับค่านามที่พบในส่วนข้อมูลนำเข้าและส่วนข้อมูลนำออก มากกว่าค่านามที่พบในส่วนอื่น

การพิจารณาค่านาม **ว่าเป็นแหล่งที่มาหรือแหล่งปลายทางของข้อมูลที่เกี่ยวข้องกับการทำงานหรือไม่** พิจารณาดังต่อไปนี้

- 1) มีการใช้ฐานข้อมูลคำศัพท์เข้ามาช่วย สำหรับคำที่จะเป็นแหล่งที่มาและแหล่งปลายทางของข้อมูลเสมอ เช่น ฐานข้อมูล ผู้ใช้ ลูกค้า
- 2) ใช้การพิจารณาร่วมกับคำบุพบทที่เกี่ยวข้อง

การพิจารณาลำดับการทำงานที่ต่อเนื่องกัน พิจารณาดังต่อไปนี้

- 1) โครงสร้างใหม่ที่ได้เรียงลำดับตามการทำงานที่ต่อเนื่องกันในแต่ละความต้องการด้านหน้าที่ โดยเรียงลำดับตามประโยคที่ใช้ในการเขียนเอกสาร
- 2) โครงสร้างใหม่ที่ได้เรียงลำดับของความต้องการด้านหน้าที่ต่างกันต่อกัน โดยเรียงลำดับตามการเขียนเอกสารข้อกำหนดความต้องการซอฟต์แวร์

การกำหนดการถ่วงน้ำหนัก เพื่อใช้เป็นตัวช่วยในการสร้างโครงสร้างใหม่ มีรายละเอียดดังต่อไปนี้

- 1) กำหนดให้คำกริยาที่พบในประโยคส่วนกระบวนการของความต้องการด้านหน้าที่มีน้ำหนักมากที่สุด (ค่าน้ำหนักเท่ากับ 4) ในส่วนบทนำ และชื่อมีน้ำหนักน้อยลงลดหลั่นลงมาตามลำดับ (ค่าน้ำหนักเท่ากับ 3 และ 2) และในส่วนข้อมูลนำเข้าและข้อมูลนำออก มีค่าน้ำหนักน้อยที่สุดเท่ากัน (ค่าน้ำหนักเท่ากับ 1)
- 2) พิจารณาการซ้ำซ้อนกันของคำกริยาและส่วนค่านามของประโยคที่พบในส่วนอื่น ๆ กับประโยคในส่วนกระบวนการ โดยถ้ามีการซ้ำซ้อนกันจะเพิ่มน้ำหนักของประโยคในส่วนกระบวนการด้วยการบวกค่าน้ำหนักของประโยคเดียวกันที่พบในส่วนอื่นเข้าไป และเปลี่ยนน้ำหนักของประโยคที่พบในส่วนอื่นนั้นเป็นศูนย์ สำหรับป้องกันการเลือกประโยคซ้ำซ้อน
- 3) พิจารณาค่านามหรือนามวลีที่พบในส่วนข้อมูลนำเข้าและข้อมูลนำออกของความต้องการด้านหน้าที่เพิ่มเติม ถึงแม้ว่าค่านามหรือวลีนั้นจะไม่อยู่ในส่วนประกอบของประโยค เนื่องจากการศึกษาจากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ พบว่า บางครั้งการเขียนเอกสารในส่วนข้อมูลนำเข้าและข้อมูลนำออกนั้น ไม่เขียนอยู่ในรูปของประโยค แต่จะเขียนในลักษณะค่านามหรือนามวลีที่เป็นข้อมูลแต่ละข้อมูลต่อเนื่องกันไป โดยใช้เว้นวรรคเป็นตัวแบ่งแต่ละข้อมูล

การพิจารณาความหมายคำ เพื่อใช้เป็นตัวช่วยในการสร้างโครงสร้างใหม่
พิจารณาดังต่อไปนี้

- 1) หน้าที่ของคำกริยา คำกริยาที่พิจารณาเป็นคำกริยาแสดงการกระทำ หรือเป็นเพียงกริยาที่บอกสถานะ คำนามที่เกี่ยวข้องกับคำกริยานี้ ควรเป็นข้อมูลนำเข้าที่ทำให้เกิดการ ทำงาน หรือเป็นข้อมูลนำออกที่เป็นผลลัพธ์ของการทำงาน ซึ่งพิจารณาจากฐานข้อมูลคำศัพท์คำกริยาที่เป็นฐานความรู้
- 2) มีการตรวจสอบคำนามที่แสดงถึงข้อมูลได้ว่า บางข้อมูลอาจเป็นส่วนประกอบของอีกข้อมูลหนึ่ง เช่น รหัสลูกค้า เป็นองค์ประกอบของข้อมูลลูกค้า
- 3) มีการตรวจสอบได้ว่า คำภาษาไทยและคำภาษาอังกฤษได้มีความหมายเดียวกันได้

หมายเหตุ ผลลัพธ์ที่ถูกต้องของการพิจารณาความหมายคำ ต้องอาศัยฐานข้อมูลที่เป็นฐานความรู้ที่มีความถูกต้อง สมบูรณ์ รายละเอียดของฐานข้อมูลศัพท์ และฐานข้อมูลคำศัพท์ที่เป็นฐานความรู้จะอธิบายในบทที่ 4 การพัฒนาเครื่องมือซอฟต์แวร์

หลังจากทำการเลือกส่วนสำคัญของประโยค จะได้ส่วนต่าง ๆ ของประโยคซึ่งแสดงถึงการกระทำ ข้อมูลที่ถูกกระทำ แหล่งที่มาหรือแหล่งปลายทางของข้อมูล ซึ่งประโยคที่ได้ทั้งหมดจะถูกเลือกเพื่อนำมาสร้างเป็นโครงสร้างใหม่ ซึ่งจะพิจารณาจากการถ่วงน้ำหนักของแต่ละประโยค โดยที่ประโยคที่มีน้ำหนักไม่เป็นศูนย์จะถูกเลือกมาสร้างเป็นโครงสร้างใหม่และพิจารณา รวมกับความหมายคำที่กล่าวในข้างต้น

3.2.4.2 การสร้างโครงสร้างใหม่และเพิ่มเติมข้อมูลเพื่อให้โครงสร้างใหม่ สมบูรณ์มากขึ้น

ในส่วนนี้จะนำประโยคที่ถูกเลือกแล้ว มาสร้างเป็นโครงสร้างใหม่ ผลลัพธ์ที่ได้คือ

- 1) ส่วนกริยาของประโยค จะกลายเป็นส่วนการทำงานของโครงสร้างใหม่
- 2) ส่วนคำนามที่เกี่ยวข้องกับคำกริยาของประโยค จะกลายเป็นส่วนข้อมูลนำเข้าหรือข้อมูลนำออกของโครงสร้างใหม่ ขึ้นกับการตรวจสอบกับฐานข้อมูลคำศัพท์คำกริยาที่เป็นฐานข้อมูลความรู้บอกลักษณะของกริยาตัวนั้น โดยถ้าไม่พบคำกริยานั้นในฐานข้อมูลจะได้คำนามนั้นเป็นข้อมูลนำออก เนื่องจากการศึกษาจากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ คำนามที่ตามหลังคำกริยานั้น มักเป็นข้อมูลที่ถูกกระทำจากการทำงานนั้น
- 3) ส่วนคำนามที่แสดงถึงแหล่งที่มาหรือแหล่งปลายทางของข้อมูล จะกลายเป็นส่วนแหล่งที่มาหรือแหล่งปลายทางของโครงสร้างใหม่ ขึ้นอยู่กับความเกี่ยวข้องกับคำนามที่เป็นข้อมูลนำเข้าหรือข้อมูลนำออก

- 4) ส่วนอื่น ๆ ของโครงสร้างใหม่ที่ยังไม่สมบูรณ์จะมีการเพิ่มเติมในขั้นตอนการทำงานถัดไป

ต่อจากนั้นในส่วนนี้มีการพิจารณาเงื่อนไขต่าง ๆ เพิ่มเติม เพื่อสร้างส่วนที่ยังไม่สมบูรณ์ในโครงสร้างใหม่ให้มีความครบถ้วน สมบูรณ์ขึ้น จากรูปที่ 3.5 แสดงตัวอย่างโครงสร้างใหม่ที่สร้างขึ้น

ฟังก์ชัน 1.1 (ฟังก์ชันที่ 1 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ผู้ใช้	ชื่อลูกค้า	สร้างรหัสลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฟังก์ชัน 1.2

ฟังก์ชัน 1.2 (ฟังก์ชันที่ 2 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.1	รหัสลูกค้า ชื่อลูกค้า	เก็บข้อมูล	รหัสลูกค้า ชื่อลูกค้า	ฐานข้อมูล

รูปที่ 3.5 ตัวอย่างโครงสร้างใหม่ที่สร้างขึ้น

รายละเอียดการเพิ่มเติมส่วนต่าง ๆ ของโครงสร้างใหม่ ประกอบด้วย

- 1) หมายเลขฟังก์ชันการทำงานของโครงสร้างใหม่ มีการกำหนดตามหมายเลขความต้องการด้านหน้าที่และลำดับการทำงาน
- 2) ส่วนยูสเคสของโครงสร้างใหม่ ซึ่งเป็นส่วนที่แสดงถึงผู้ที่ดูแลควบคุมฟังก์ชันการทำงานนั้น จะกำหนดจากลำดับที่ของความต้องการด้านหน้าที่แต่ละหน้าที่
- 3) ส่วนแหล่งที่มาและแหล่งปลายทางของโครงสร้างใหม่
 - แหล่งที่มา สำหรับฟังก์ชันการทำงานแรกในแต่ละความต้องการด้านหน้าที่จะเป็นแหล่งที่มาของข้อมูลที่พบในเอกสารส่วนข้อมูลนำเข้าของความต้องการด้านหน้าที่
 - แหล่งปลายทาง สำหรับฟังก์ชันการทำงานสุดท้ายในแต่ละความต้องการด้านหน้าที่จะเป็นแหล่งปลายทางของข้อมูลที่พบในเอกสารส่วนข้อมูลนำออกของความต้องการด้านหน้าที่
 - แหล่งที่มาและแหล่งปลายทาง จะขึ้นกับคำกริยาบางคำที่ความหมายของคำ แสดงให้เห็นว่าแหล่งที่มาหรือแหล่งปลายทางจะเป็นฐานข้อมูลเสมอ ตัวอย่าง คำเช่น ค้นหา เก็บข้อมูล

- แหล่งที่มาและแหล่งปลายทาง ที่มีการทำงานต่อเนื่องกันของฟังก์ชันการทำงานในความต้องการด้านหน้าที่เดียวกัน จะกำหนดตามลำดับของฟังก์ชันการทำงาน
- 4) ส่วนข้อมูลนำเข้าและข้อมูลนำออกของโครงสร้างใหม่ ใช้ความหมายของคำกริยาและลำดับการทำงานมาช่วยในการพิจารณา
 - คำกริยาบางคำ มีความหมายว่าข้อมูลที่ถูกริยานั้นกระทำ ซึ่งเป็นข้อมูลนำออกจะเป็นข้อมูลนำเข้าด้วยเสมอ ตัวอย่างคำเช่น “เก็บ” เมื่อมีการทำงานเก็บข้อมูล ฟังก์ชันการทำงานนั้นต้องรับข้อมูลที่จะทำการจัดเก็บเข้ามาเป็นข้อมูลนำเข้าของฟังก์ชันการทำงาน ก่อนทำการจัดเก็บและกลายเป็นข้อมูลนำออกไปสู่ฐานข้อมูล
 - ฟังก์ชันการทำงานที่มีลำดับต่อเนื่องกัน ในความต้องการด้านหน้าที่เดียวกัน ถ้าข้อมูลนำเข้าหรือข้อมูลนำออกของฟังก์ชันการทำงานไม่ระบุไว้ในเอกสาร จะมีการเพิ่มเติมข้อมูลเข้าไป โดยข้อมูลนำออกของฟังก์ชันที่ทำงานก่อน จะเป็นข้อมูลนำเข้าของฟังก์ชันที่ทำงานตามมาเสมอ และในลักษณะเดียวกันข้อมูลนำเข้าของฟังก์ชันการทำงานที่ทำตามมา จะเป็นข้อมูลนำออกของฟังก์ชันที่ทำงานก่อนเสมอ
- 5) ส่วนการทำงานของโครงสร้างใหม่ จะมีการรวมคำกริยาเข้ากับค่านามข้อมูลที่ถูกกริยานั้นกระทำ เพื่อให้เห็นการทำงานชัดเจน

3.3 การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้องการ ในเอกสารข้อกำหนดความต้องการซอฟต์แวร์

จากทฤษฎีที่เกี่ยวข้อง หัวข้อ 2.1.2 ข้อกำหนดความต้องการซอฟต์แวร์ [11] มีการกล่าวถึงความต้องการในคุณสมบัติของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ดี ว่าความไม่ต้องการเกิดขึ้นเมื่อมีความต้องการใดที่อธิบายแล้วเกิดความขัดแย้ง ซึ่งความไม่ต้องการในงานวิจัยนี้จะทำให้เกิดความขัดแย้งในกลุ่มที่ 3 คือ “ความต้องการสองความต้องการหรือมากกว่าที่อธิบายสิ่งเดียวกัน แต่ใช้คำหรือข้อความต่างกัน” และสอดคล้องกับนิยามความไม่ต้องการในทฤษฎีที่เกี่ยวข้อง หัวข้อ 2.1.5 ความไม่ต้องการในการพัฒนาซอฟต์แวร์ [14] กล่าวถึงความไม่ต้องการ คือ สถานการณ์ที่การอธิบายสองการอธิบายใด ๆ ไม่ตรงตามความสัมพันธ์ระหว่างกัน ซึ่งความสัมพันธ์ระหว่างกันนั้นเปรียบได้กับกฎความต้องการ โดยที่กฎความต้องการสามารถหาได้จากแบบจำลองกระบวนการในการพัฒนาได้ ดังนั้นกฎความต้องการในงานวิจัยนี้ จะหามาจากความต้องการของโครงสร้างใหม่

ที่สร้างขึ้นหลังจากการประมวลผลภาษาธรรมชาติ ซึ่งส่วนหนึ่งอ้างอิงกับทฤษฎีที่เกี่ยวข้อง หัวข้อ 2.1.4 ยูสเคสและวากยสัมพันธ์ของแผนภาพการไหลของข้อมูล [13] โดยใช้วากยสัมพันธ์ของกระบวนการ วากยสัมพันธ์และหลักเกณฑ์สถานะการไหลของข้อมูล และกฎการตรวจสอบความไม่ต้องการในงานวิจัยที่เกี่ยวข้อง หัวข้อ 2.2.7 [10] มาช่วยในการสร้างกฎความไม่ต้องการ

ความหมายของความไม่ต้องการ ที่งานวิจัยนี้จะทำการตรวจสอบ คือ

- 1) ความไม่ต้องการของข้อมูลในฐานข้อมูล มีการดึงข้อมูลจากฐานข้อมูลขึ้นมาเพื่อแสดงผล แต่ไม่มีการเก็บข้อมูลนั้นลงฐานข้อมูล ซึ่งเป็นความไม่ต้องการที่เกิดจากการอธิบายข้อมูลในฐานข้อมูลไม่ตรงกัน ตัวอย่างเช่น
 - มีฟังก์ชันการทำงานต้องการแสดงผลข้อมูลที่อยู่ลูกค้า หลังจากการรับซื้อลูกค้าเข้ามา แต่ถ้าไม่มีฟังก์ชันการทำงานที่เก็บข้อมูลที่อยู่ลูกค้า จะไม่พบข้อมูลที่อยู่ลูกค้าในฐานข้อมูลและไม่สามารถทำการแสดงผลได้
- 2) ความไม่ต้องการในความต้องการด้านหน้าที่เดียวกัน ซึ่งเป็นความไม่ต้องการที่เกิดจากการอธิบายฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกไม่ตรงกัน หรือมีการอธิบายที่ซ้ำซ้อน ตัวอย่างเช่น
 - ในความต้องการด้านหน้าที่เดียวกันมีฟังก์ชันการทำงานที่ซ้ำกัน
 - ข้อมูลซื้อลูกค้าเป็นข้อมูลนำเข้าของฟังก์ชันการทำงานสร้างรหัสลูกค้าแต่ไม่พบการเป็นข้อมูลนำออก (ข้อมูลมีการสูญหาย)
 - ฟังก์ชันการทำงานที่ทำงานก่อนมีการส่งข้อมูลนำออก คือรหัสลูกค้าและซื้อลูกค้าไปยังฟังก์ชันการทำงานถัดไป แต่ฟังก์ชันการทำงานถัดไปมีข้อมูลนำเข้าแค่ซื้อลูกค้าเพียงอย่างเดียว (ข้อมูลนั้นเป็นข้อมูลนำออกแต่ไม่พบการเป็นข้อมูลนำเข้า)
 - ฟังก์ชันการทำงานที่ทำงานถัดไปมีข้อมูลนำเข้าจากฟังก์ชันการทำงานก่อนคือรหัสลูกค้า ซื้อลูกค้า และที่อยู่ลูกค้า แต่ฟังก์ชันการทำงานก่อนมีการส่งข้อมูลนำออก คือรหัสลูกค้าและซื้อลูกค้าไปยังฟังก์ชันการทำงานถัดไปเท่านั้น (ข้อมูลนั้นเป็นข้อมูลนำเข้าแต่ไม่พบการเป็นข้อมูลนำออก)
- 3) ความไม่ต้องการระหว่างความต้องการด้านหน้าที่ที่ต่างหน้าที่กัน ซึ่งเป็นความไม่ต้องการที่เกิดจากการอธิบายฟังก์ชันการทำงาน ข้อมูลนำเข้า ข้อมูลนำออกที่ซ้ำซ้อน หรือมีการอธิบายข้อมูลในฐานข้อมูลที่เป็นข้อมูลเดียวกันไม่ตรงกัน ตัวอย่างเช่น

- ความต้องการด้านหน้าที่ต่างหน้าที่กัน แต่มีฟังก์ชันการทำงานที่เหมือนกันทั้งหมด แสดงว่าเกิดความซ้ำซ้อนของความต้องการด้านหน้าที่
- คำที่ใช้ในการเขียนเอกสาร ที่มีความสำคัญในการแสดงถึง ข้อมูลนำเข้าและข้อมูลนำออก บางครั้งใช้คำกำกวม เกิดความไม่ต้องการกัน เมื่อพิจารณารวมทุกความต้องการด้านหน้าที่ ตัวอย่างเช่น คำว่า “ข้อมูลลูกค้า” จะไม่สามารถระบุได้ว่า ข้อมูลลูกค้าประกอบด้วยข้อมูลอะไรบ้าง บางความต้องการด้านหน้าที่อาจหมายถึง รหัสลูกค้าเพียงอย่างเดียว ในขณะที่ความต้องการด้านหน้าที่หมายถึง รหัสลูกค้า และ ชื่อลูกค้า (คำที่หมายถึงข้อมูลเดียวกัน แต่ประกอบด้วยข้อมูลย่อยต่างกัน)

กฎความต้องกัน ที่นำมาตรวจสอบความไม่ต้องการกันในงานวิจัยนี้ คือ

- กฎที่ 1 ข้อมูลทุกข้อมูลที่มีแหล่งที่มาจากรฐานข้อมูล เพื่อเป็นข้อมูลนำเข้าสู่ฟังก์ชันการทำงานใด ๆ ต้องเป็นข้อมูลนำออกจกฟังก์ชันการทำงานอื่นและมีแหล่งปลายทางคือฐานข้อมูล ใช้ในการตรวจสอบความไม่ต้องการกันข้อ 1)
- กฎที่ 2 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำเข้าของฟังก์ชันการทำงานใด ๆ จะต้องเป็นข้อมูลนำออกจกฟังก์ชันการทำงานนั้นเสมอ เนื่องจากข้อมูลต้องไม่มีการสูญหายระหว่างฟังก์ชันการทำงาน ซึ่งเป็นหลักเกณฑ์สถานะการไหลของข้อมูล ใช้ในการตรวจสอบความไม่ต้องการกันข้อ 2)
- กฎที่ 3 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำออกจกฟังก์ชันการทำงานใด ๆ ไปยังฟังก์ชันการทำงานถัดไป จะต้องเป็นข้อมูลนำเข้าของฟังก์ชันการทำงานถัดไปนั้น เนื่องจากพิจารณาการทำงานที่ต่อเนื่องกัน และประยุกต์จากกฎการตรวจสอบความไม่ต้องการกันของงานวิจัย 2.2.7 [10] ใช้ในการตรวจสอบความไม่ต้องการกันข้อ 2)
- กฎที่ 4 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำเข้าของฟังก์ชันการทำงานใด ๆ จะต้องเป็นข้อมูลนำออกจกของฟังก์ชันการทำงานก่อนหน้านั้น เนื่องจากพิจารณาการทำงานที่ต่อเนื่องกัน และประยุกต์จากกฎการตรวจสอบความไม่ต้องการกันของงานวิจัย 2.2.7 [10] ใช้ในการตรวจสอบความไม่ต้องการกันข้อ 2)
- กฎที่ 5 ฟังก์ชันการทำงานทุกฟังก์ชันในความต้องการด้านหน้าที่เดียวกัน ต้องมีฟังก์ชันการทำงานนั้นเพียงฟังก์ชันเดียวเท่านั้น เนื่องจากในความต้องการด้านหน้าที่เดียวกัน ห้ามมีฟังก์ชันการทำงานที่ซ้ำซ้อนกัน ใช้ในการตรวจสอบความไม่ต้องการกันข้อ 2)

กฎที่ 6 ความต้องการด้านหน้าที่ทุกความต้องการ ต้องมีความต้องการด้านหน้าที่เพียง ความต้องการเดียวเท่านั้น เนื่องจากห้ามมีความต้องการด้านที่ซ้ำซ้อนกัน ใช้ใน การตรวจสอบความไม่ต้องการข้อ 3)

กฎที่ 7 คำทุกคำที่หมายถึงกลุ่มข้อมูลเดียวกันในฐานข้อมูล ต้องมีข้อมูลย่อยที่เป็นส่วน ประกอบของกลุ่มข้อมูลนั้นเป็นข้อมูลย่อยเดียวกัน ใช้ในการตรวจสอบความไม่ ต้องการข้อ 3)

ตารางที่ 3.1 สรุปความไม่ต้องการกันและกฎความต้องการกันที่ใช้ในการตรวจสอบ

ความไม่ต้องการกัน	กฎความต้องการกัน
ความไม่ต้องการกันของข้อมูล ในฐานข้อมูล	กฎที่ 1 ข้อมูลทุกข้อมูลที่มีแหล่งที่มาจากรฐานข้อมูล เพื่อเป็นข้อมูลนำ เข้าสู่ฟังก์ชันการทำงานใด ๆ ต้องเป็นข้อมูลนำออกจากฟังก์ชันการ ทำงานอื่นและมีแหล่งปลายทางคือฐานข้อมูล
ความไม่ต้องการกันในความ ต้องการด้านหน้าที่เดียว กัน	กฎที่ 2 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำเข้าของฟังก์ชันการทำงานใด ๆ จะต้องเป็นข้อมูลนำออกจากฟังก์ชันการทำงานนั้นเสมอ
	กฎที่ 3 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำออกจากฟังก์ชันการทำงานใด ๆ ไปยังฟังก์ชันการทำงานถัดไป จะต้องเป็นข้อมูลนำเข้าของฟังก์ชันการ ทำงานถัดไปนั้น
	กฎที่ 4 ข้อมูลทุกข้อมูลที่เป็นข้อมูลนำเข้าของฟังก์ชันการทำงานใด ๆ จะต้องเป็นข้อมูลนำออกจากของฟังก์ชันการทำงานก่อนหน้านั้น
	กฎที่ 5 ฟังก์ชันการทำงานทุกฟังก์ชันในความต้องการด้านหน้าที่เดียว กัน ต้องมีฟังก์ชันการทำงานนั้นเพียงฟังก์ชันเดียวเท่านั้น
ความไม่ต้องการกันระหว่าง ความต้องการด้านหน้าที่ ที่ต่างหน้าที่กัน	กฎที่ 6 ความต้องการด้านหน้าที่ทุกความต้องการ ต้องมีความต้องการ ด้านหน้าที่เพียงความต้องการเดียวเท่านั้น
	กฎที่ 7 คำทุกคำที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล ต้องมีข้อมูลย่อยที่ เป็นส่วนประกอบของกลุ่มข้อมูลนั้นเป็นข้อมูลย่อยเดียวกัน

3.3.1 การตรวจสอบความไม่ต้องการกันของข้อมูลในฐานข้อมูล

ทำการตรวจสอบโครงสร้างใหม่ของฟังก์ชันการทำงานที่มีแหล่งที่มาเป็นฐานข้อมูลว่ามีข้อมูลใดเป็นข้อมูลนำเข้าบ้าง แล้วตรวจสอบกับฟังก์ชันการทำงานอื่น ๆ ทุกฟังก์ชัน ที่มีแหล่งปลายทางเป็นฐานข้อมูล ว่าข้อมูลเหล่านั้นเป็นข้อมูลที่นำออกไปสู่ฐานข้อมูลหรือไม่ โดยถ้าข้อมูลนำเข้า ที่มีแหล่งที่มาจากรฐานข้อมูล ไม่พบว่าเป็นข้อมูลนำออกที่มีแหล่งปลายทางเป็นฐานข้อมูลแล้ว จะ แสดงผลว่าเกิดความไม่ต้องการในฐานข้อมูลขึ้น และระบุฟังก์ชันการทำงานที่มีข้อมูลนั้น

ตัวอย่างความไม่ต้องการกันของฐานข้อมูล คือ

ตารางที่ 3.2 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 1

หน้าที่ที่ 1	รับข้อมูลของลูกค้าใหม่
บทบาท	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้าใหม่จากผู้ใช้
กระบวนการ	รับข้อมูลชื่อลูกค้าใหม่ แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	ชื่อลูกค้า รหัสลูกค้า ไปเก็บในฐานข้อมูล

จากข้อความในตารางที่ 3.2 นำมาสร้างเป็นโครงสร้าง แสดงดังรูปที่ 3.6

จากรูปที่ 3.6 มีลักษณะการทำงานคือ ฟังก์ชัน 1.1 รับข้อมูลชื่อลูกค้าใหม่ แล้วส่งชื่อลูกค้าใหม่ไปยังฟังก์ชันการทำงานถัดไป คือฟังก์ชัน 1.2 ต่อจากนั้นฟังก์ชัน 1.2 รับชื่อลูกค้าและสร้างรหัสลูกค้า แล้วส่งข้อมูลนำออก คือรหัสลูกค้าและชื่อลูกค้าไปยังฟังก์ชัน 1.3 เพื่อให้ฟังก์ชัน 1.3 ทำการเก็บข้อมูลลงในฐานข้อมูล

ฟังก์ชัน 1.1 (ฟังก์ชันที่ 1 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ผู้ใช้	ชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 1.2

ฟังก์ชัน 1.2 (ฟังก์ชันที่ 2 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.1	ชื่อลูกค้า	สร้างรหัสลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฟังก์ชัน 1.3

ฟังก์ชัน 1.3 (ฟังก์ชันที่ 3 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.2	รหัสลูกค้า ชื่อลูกค้า	เก็บข้อมูลลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฐานข้อมูล

รูปที่ 3.6 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 1

ตารางที่ 3.3 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 2

หน้าที่ที่ 2	การพิมพ์ใบส่งสินค้า
บทบาท	หน้าที่ในส่วนนี้ เป็นการดึงข้อมูลลูกค้าจากฐานข้อมูล แล้วส่งพิมพ์ใบส่งสินค้า ซึ่งประกอบด้วย ชื่อและที่อยู่ของลูกค้าออกทางเครื่องพิมพ์
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับชื่อลูกค้า เพื่อดึงข้อมูลที่อยู่ที่ของลูกค้าจากฐานข้อมูล แล้วพิมพ์ใบส่งสินค้า
ข้อมูลนำออก	ชื่อและที่อยู่ของลูกค้าไปยังผู้ใช้

จากข้อความในตารางที่ 3.3 นำมาสร้างเป็นโครงสร้าง แสดงดังรูปที่ 3.7

จากรูปที่ 3.7 มีลักษณะการทำงานคือ ฟังก์ชัน 2.1 รับข้อมูลชื่อลูกค้า และส่งข้อมูลชื่อลูกค้าเป็นข้อมูลนำออกไปยังฟังก์ชัน 2.2 ต่อจากนั้นฟังก์ชัน 2.2 รับข้อมูลชื่อลูกค้า และส่งเป็นข้อมูลนำออกไปยังฐานข้อมูล เพื่อเป็นคีย์ในการดึงข้อมูลจากฐานข้อมูล แล้วฟังก์ชัน 2.3 รับข้อมูลนำเข้ามาจากฐานข้อมูล เพื่อพิมพ์ใบส่งสินค้า

ฟังก์ชัน 2.1 (ฟังก์ชันที่ 1 ของหน้าที่การพิมพ์ใบส่งสินค้า)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 2	ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 2.2

ฟังก์ชัน 2.2 (ฟังก์ชันที่ 2 ของหน้าที่การพิมพ์ใบส่งสินค้า)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 2	ฟังก์ชัน 2.1	ชื่อลูกค้า	ดึงข้อมูล	ชื่อลูกค้า	ฐานข้อมูล

ฟังก์ชัน 2.3 (ฟังก์ชันที่ 3 ของหน้าที่การพิมพ์ใบส่งสินค้า)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 2	ฐานข้อมูล	ชื่อลูกค้า ที่อยู่ลูกค้า	พิมพ์ใบ ส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

รูปที่ 3.7 โครงสร้างการทำงานของความต้องการด้านหน้าที่ ตัวอย่างที่ 2
การพิมพ์ใบส่งสินค้า

ถ้าพิจารณา ฟังก์ชัน 2.3 ของความต้องการด้านหน้าที่การพิมพ์ใบส่งสินค้า ในรูปที่ 3.7 เปรียบเทียบกับโครงสร้าง ฟังก์ชัน 1.3 ที่ได้จากความต้องการด้านหน้าที่การรับข้อมูลลูกค้าใหม่ ในรูปที่ 3.6 จะเห็นได้ว่าเกิดความไม่ต้องการกันของฐานข้อมูล เพราะข้อมูลนำออกจากความต้องการ

ด้านหน้าที่การรับข้อมูลลูกค้าใหม่ (ฟังก์ชัน 1.3) ไปยังฐานข้อมูลมีเพียงชื่อและรหัสลูกค้าเท่านั้น แต่ข้อมูลที่ต้องการดึงออกจากฐานข้อมูลมีทั้งชื่อและที่อยู่ของลูกค้า (ฟังก์ชัน 2.3) ซึ่งมีผลทำให้เกิดความไม่ต้องการในการอธิบายฐานข้อมูล เนื่องจากข้อมูลที่มีเพียงรหัสลูกค้ากับชื่อลูกค้า แต่ข้อมูลที่ต้องการมีทั้งชื่อลูกค้าและที่อยู่แสดงดังรูปที่ 3.8 โดยในวงรีแสดงถึงส่วนที่เกิดความไม่ต้องการซึ่งไม่ตรงตามกฎข้อที่ 1

ฟังก์ชัน 1.3 (ฟังก์ชันที่ 3 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.2	รหัสลูกค้า ชื่อลูกค้า	เก็บข้อมูลลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฐานข้อมูล

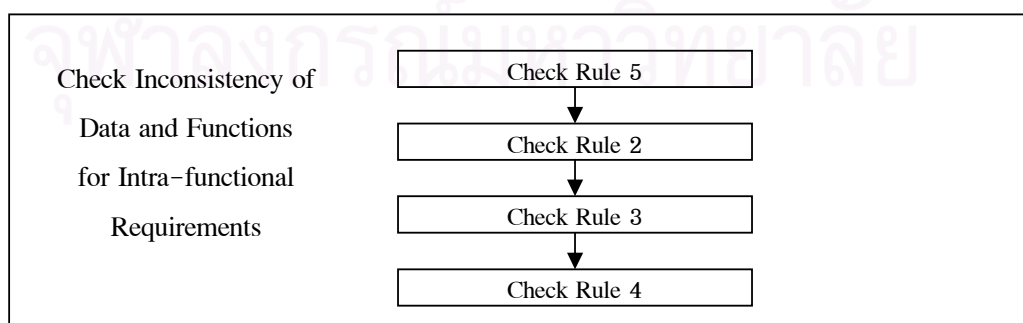
ฟังก์ชัน 2.3 (ฟังก์ชันที่ 3 ของหน้าที่การพิมพ์ใบส่งสินค้า)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 2	ฐานข้อมูล	ชื่อลูกค้า ที่อยู่ลูกค้า	พิมพ์ใบ ส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

รูปที่ 3.8 โครงสร้างการทำงานที่ไม่ต้องการกันของฐานข้อมูล

3.3.2 การตรวจสอบความไม่ต้องการกัน ในความต้องการด้านหน้าที่การทำงานเดียวกัน

ทำการตรวจสอบโครงสร้างใหม่ของฟังก์ชันการทำงานทั้งหมดที่อยู่ในความต้องการด้านหน้าที่เดียวกัน ทั้งในส่วนฟังก์ชันการทำงาน ข้อมูลนำเข้า และข้อมูลนำออก ตามกฎความต้องการกันที่กล่าวข้างต้น แสดงดังรูปที่ 3.9



รูปที่ 3.9 ขั้นตอนการตรวจสอบความไม่ต้องการกัน ในความต้องการด้านหน้าที่เดียวกัน

ตัวอย่างความไม่ต้องกันในการความต้องการด้านหน้าที่เดียวกัน เมื่อข้อมูลนั้นเป็นข้อมูลนำออกจากฟังก์ชันการทำงานก่อนหน้า แต่ไม่พบการเป็นข้อมูลนำเข้าของฟังก์ชันการทำงานถัดไปคือ

ตารางที่ 3.4 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 3

หน้าที่ที่ 1	รับข้อมูลของลูกค้าใหม่
บทบาท	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้าใหม่จากผู้ใช้
กระบวนการ	รับข้อมูลชื่อลูกค้าใหม่ แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	ชื่อลูกค้าไปเก็บในฐานข้อมูล

จากข้อความในตารางที่ 3.4 นำมาสร้างเป็นโครงสร้าง แสดงดังรูปที่ 3.10

จากรูปที่ 3.10 มีลักษณะการทำงานคือ ฟังก์ชัน 1.1 รับข้อมูลชื่อลูกค้าใหม่ แล้วส่งชื่อลูกค้าใหม่ไปยังฟังก์ชันการทำงานถัดไป คือฟังก์ชัน 1.2 ต่อจากนั้นฟังก์ชัน 1.2 รับชื่อลูกค้าและสร้างรหัสลูกค้า แล้วส่งข้อมูลนำออก คือรหัสลูกค้าและชื่อลูกค้าไปยังฟังก์ชัน 1.3 เพื่อให้ฟังก์ชัน 1.3 ทำการเก็บข้อมูลลงในฐานข้อมูล แต่ฟังก์ชัน 1.3 รับแค่ชื่อลูกค้าและเก็บลงในฐานข้อมูล ส่วนข้อมูลรหัสลูกค้าหายไป โดยในวงรีรูปที่ 3.10 แสดงถึงส่วนที่เกิดความไม่ต้องกัน ซึ่งไม่ตรงตามกฎข้อที่ 3 ฟังก์ชัน 1.1 (ฟังก์ชันที่ 1 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ผู้ใช้	ชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 1.2

ฟังก์ชัน 1.2 (ฟังก์ชันที่ 2 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.1	ชื่อลูกค้า	สร้างรหัสลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฟังก์ชัน 1.3

ฟังก์ชัน 1.3 (ฟังก์ชันที่ 3 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.2	ชื่อลูกค้า	เก็บข้อมูลลูกค้า	ชื่อลูกค้า	ฐานข้อมูล

รูปที่ 3.10 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 3

ตัวอย่างความไม่ต้อกันในการต้องการด้านหน้าที่เดียวกัน เมื่อข้อมูลนั้นเป็นข้อมูลนำเข้าของฟังก์ชันที่ทำงานถัดไป แต่ไม่พบการเป็นข้อมูลนำออกของฟังก์ชันการทำงานก่อนหน้า คือ

ตารางที่ 3.5 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 4

หน้าที่ที่ 1	รับข้อมูลของลูกค้าใหม่
บทบาท	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้าใหม่จากผู้ใ้
กระบวนการ	รับข้อมูลชื่อลูกค้าใหม่ แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	รหัสลูกค้า ชื่อลูกค้า ที่อยู่ลูกค้า ไปเก็บในฐานข้อมูล

จากข้อความในตารางที่ 3.5 นำมาสร้างเป็นโครงสร้าง แสดงดังรูปที่ 3.11

จากรูปที่ 3.11 มีลักษณะการทำงานคือ ฟังก์ชัน 1.1 รับข้อมูลชื่อลูกค้าใหม่ แล้วส่งชื่อลูกค้าใหม่ไปยังฟังก์ชันการทำงานถัดไป คือฟังก์ชัน 1.2 ต่อจากนั้นฟังก์ชัน 1.2 รับชื่อลูกค้าและสร้างรหัสลูกค้า แล้วส่งข้อมูลนำออก คือรหัสลูกค้าและชื่อลูกค้าไปยังฟังก์ชัน 1.3 เพื่อให้ฟังก์ชัน 1.3 ทำการเก็บข้อมูลลงในฐานข้อมูล แต่ฟังก์ชัน 1.3 มีการรับข้อมูลทั้งรหัสลูกค้า ชื่อลูกค้า และที่อยู่ลูกค้า เก็บลงในฐานข้อมูล เห็นได้ว่าข้อมูลที่อยู่เกิดมาจากข้อมูลที่ส่งออกจากฟังก์ชัน 1.2 โดยในวงรีรูปที่ 3.11 แสดงถึงส่วนที่เกิดความไม่ต้อกัน ซึ่งไม่ตรงตามกฎข้อที่ 4

ฟังก์ชัน 1.1 (ฟังก์ชันที่ 1 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ผู้ใ้	ชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 1.2

ฟังก์ชัน 1.2 (ฟังก์ชันที่ 2 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.1	ชื่อลูกค้า	สร้างรหัสลูกค้า	รหัสลูกค้า ชื่อลูกค้า	ฟังก์ชัน 1.3

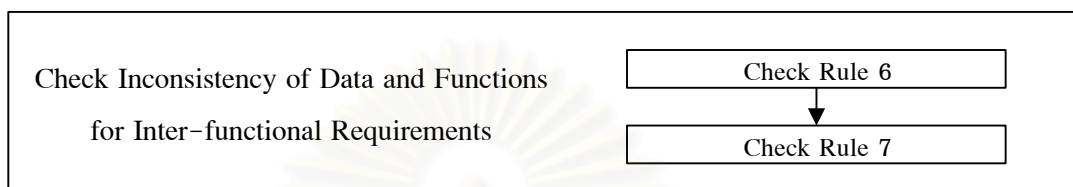
ฟังก์ชัน 1.3 (ฟังก์ชันที่ 3 ของหน้าที่การรับข้อมูลลูกค้าใหม่)

Use case	Source	Input	Action	Output	Destination
หน้าที่ที่ 1	ฟังก์ชัน 1.2	รหัสลูกค้า ชื่อลูกค้า ที่อยู่ลูกค้า	เก็บข้อมูลลูกค้า	รหัสลูกค้า ชื่อลูกค้า ที่อยู่ลูกค้า	ฐานข้อมูล

รูปที่ 3.11 โครงสร้างการทำงานของความต้องการด้านหน้าที่ตัวอย่างที่ 4

3.3.3 การตรวจสอบความไม่ตรงกันในความต้องการด้านหน้าที่การทำงานต่างกัน

ในขั้นตอนนี้จะทำการตรวจสอบโครงสร้างใหม่ของฟังก์ชันการทำงานทั้งหมดที่อยู่ในความต้องการด้านหน้าที่เดียวกัน ทั้งในส่วนฟังก์ชันการทำงาน ข้อมูลนำเข้า และข้อมูลนำออกว่ามีความซ้ำซ้อนกันหรือไม่ และทำการตรวจสอบค่าที่ใช้ในการเขียนเอกสาร แสดงดังรูปที่ 3.12



รูปที่ 3.12 ขั้นตอนการตรวจสอบความไม่ตรงกันในความต้องการด้านหน้าที่ต่างหน้าที่กัน

สำหรับการตรวจสอบค่าที่ใช้ในการเขียนเอกสาร จะตรวจสอบจากส่วนประกอบของประโยคที่เป็นผลลัพธ์จากขั้นตอนการทำงานในหัวข้อ 3.2.4.1 ที่มีส่วนประกอบของประโยคซึ่งเป็นคำนามหรือนามวลีที่มีความเกี่ยวข้องกับฐานข้อมูล หรือมีคำกริยาที่มีความเกี่ยวข้องกับฐานข้อมูล มาเปรียบเทียบกับข้อมูลที่ปรากฏในโครงสร้างใหม่ที่งานวิจัยนี้สร้างขึ้นในส่วนข้อมูลนำเข้าหรือข้อมูลนำออก โดยถ้าข้อมูลไม่ตรงกัน จะแสดงผลว่าความหมายของข้อมูลที่แทนด้วยคำคำนั้นในฟังก์ชันการทำงานที่พบ หมายถึงข้อมูลย่อยใดบ้าง เนื่องจากข้อมูลที่ปรากฏในโครงสร้างใหม่จะเป็นข้อมูลที่เห็นส่วนประกอบของข้อมูลที่ชัดเจนทั้งหมดแล้ว เพราะพิจารณาจากสิ่งต่าง ๆ รวมกันทั้งคำที่เป็นส่วนประกอบของประโยคและลำดับฟังก์ชันการทำงาน

ตัวอย่างของความไม่ตรงกันประเภทนี้ พิจารณาจากตารางที่ 3.2 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 1 และตารางที่ 3.3 ตัวอย่างความต้องการด้านหน้าที่ ตัวอย่างที่ 2 แสดงให้เห็นว่าการเขียนเอกสารบางครั้งใช้การละข้อมูลย่อยไว้ โดยเขียนรวมกันเป็นคำคำเดียวที่หมายถึงข้อมูลย่อยทั้งหมด ได้แก่คำว่า “ข้อมูลลูกค้า” แต่ความหมายที่แท้จริงที่แสดงถึงข้อมูลย่อยในข้อมูลลูกค้าของแต่ละความต้องการด้านหน้าที่ไม่ใช่ข้อมูลย่อยเดียวกัน โดยในตัวอย่างที่ 1 ข้อมูลลูกค้าจะหมายถึง ชื่อและรหัสของลูกค้า แต่ในตัวอย่างที่ 2 จะหมายถึง ชื่อและที่อยู่ของลูกค้า

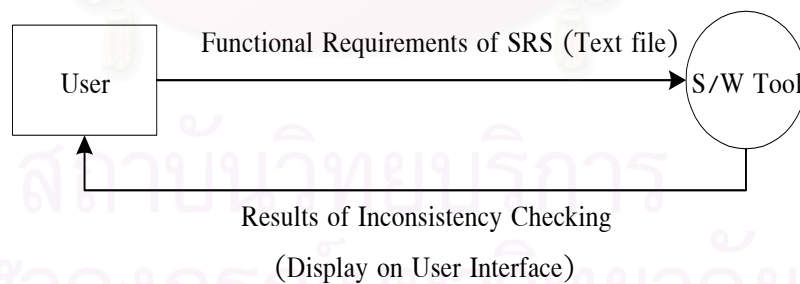
บทที่ 4

การออกแบบและพัฒนาเครื่องมือในการตรวจสอบความไม่ตรงกันในเอกสาร ข้อกำหนดความต้องการซอฟต์แวร์

การออกแบบและพัฒนาเครื่องมือซอฟต์แวร์ในการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เป็นการนำขั้นตอนวิธีที่ออกแบบในบทที่ 3 มาสร้างเป็นเครื่องมือที่สามารถทำงานได้โดยอัตโนมัติ โดยรับข้อมูลนำเข้าเป็นแฟ้มข้อมูลชนิดข้อความของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เฉพาะส่วนความต้องการด้านหน้าที่ แบ่งการออกแบบและพัฒนาเครื่องมือเป็น 3 ส่วน คือ ส่วนมอดูล (Module) การทำงานภายในเครื่องมือซอฟต์แวร์ ส่วนฐานข้อมูล และส่วนต่อประสานกับผู้ใช้ (User Interface)

4.1 การวิเคราะห์และออกแบบ

ในส่วนนี้จะประกอบด้วย การวิเคราะห์และออกแบบเครื่องมือซอฟต์แวร์ การออกแบบฐานข้อมูล และการออกแบบส่วนต่อประสานกับผู้ใช้ ซึ่งจะอธิบายรายละเอียดในหัวข้อย่อยถัดไป การทำงานของเครื่องมือซอฟต์แวร์ที่เกี่ยวข้องกับผู้ใช้เครื่องมือซอฟต์แวร์ในภาพรวม แสดงดังรูปที่ 4.1 ข้อมูลนำเข้าของเครื่องมือซอฟต์แวร์ คือแฟ้มข้อมูลชนิดข้อความของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ส่วนความต้องการด้านหน้าที่ เมื่อเครื่องมือซอฟต์แวร์ทำการประมวลผลเพื่อตรวจสอบความไม่ตรงกันในเอกสารแล้ว จะแสดงผลลัพธ์ของการตรวจสอบความไม่ตรงกันทางหน้าจอของเครื่องมือซอฟต์แวร์

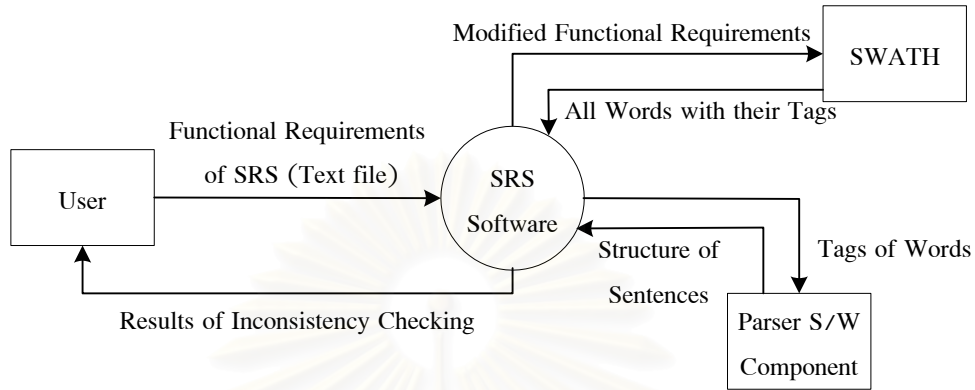


รูปที่ 4.1 แผนภาพการทำงานของเครื่องมือซอฟต์แวร์ที่เกี่ยวข้องกับผู้ใช้

4.1.1 การวิเคราะห์และออกแบบเครื่องมือซอฟต์แวร์

การวิเคราะห์และออกแบบการทำงานของเครื่องมือซอฟต์แวร์ จะได้ลักษณะการทำงานโดยรวมของมอดูลในเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ 4.2 โดยข้อมูลนำเข้าของเครื่องมือซอฟต์แวร์ เป็นเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ส่วนความต้องการด้านหน้าที่ ในรูปแฟ้ม

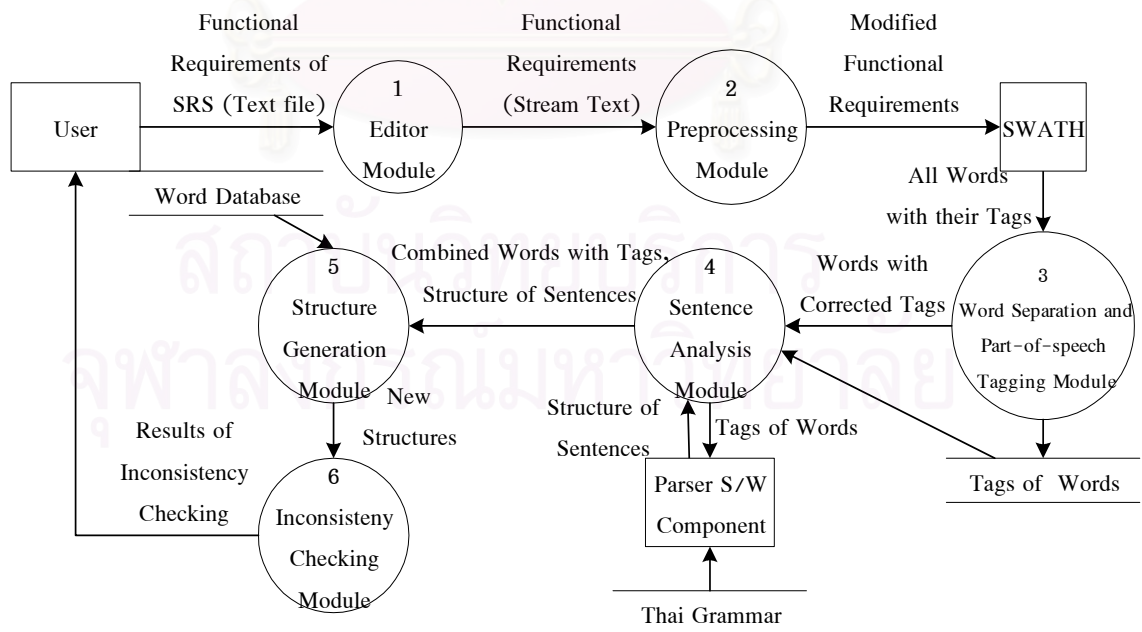
ข้อมูลชนิดข้อความ โดยมีการใช้โปรแกรม SWATH เข้ามาช่วยในการตัดคำและกำกับหน้าที่คำ การใช้ส่วนประกอบของซอฟต์แวร์ช่วยในการวิเคราะห์ด้วยไวยากรณ์ ผลลัพธ์ของการทำงานขั้นสุดท้าย คือ ผลของการตรวจสอบความไม่ตรงกันในเอกสาร



รูปที่ 4.2 แผนภาพบริบท (Context Diagram) ของเครื่องมือซอฟต์แวร์

4.1.1.1 การวิเคราะห์เครื่องมือซอฟต์แวร์

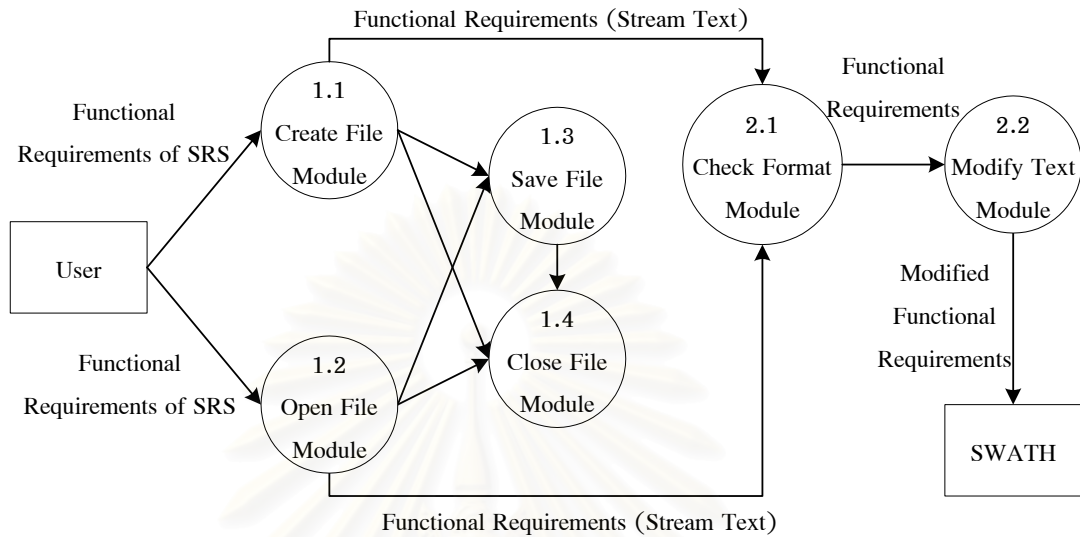
การวิเคราะห์การทำงานภายในของเครื่องมือซอฟต์แวร์ ประกอบด้วย การประมวลผลในขั้นต่าง ๆ แสดงดังรูปที่ 4.3 ได้แก่ การรับข้อมูลนำเข้า กระบวนการเตรียมเอกสาร การตัดคำ และกำกับหน้าที่คำ การวิเคราะห์ประโยค การสร้างรูปแบบโครงสร้างใหม่ และนำรูปแบบโครงสร้างใหม่ที่ได้มาตรวจสอบความไม่ตรงกัน



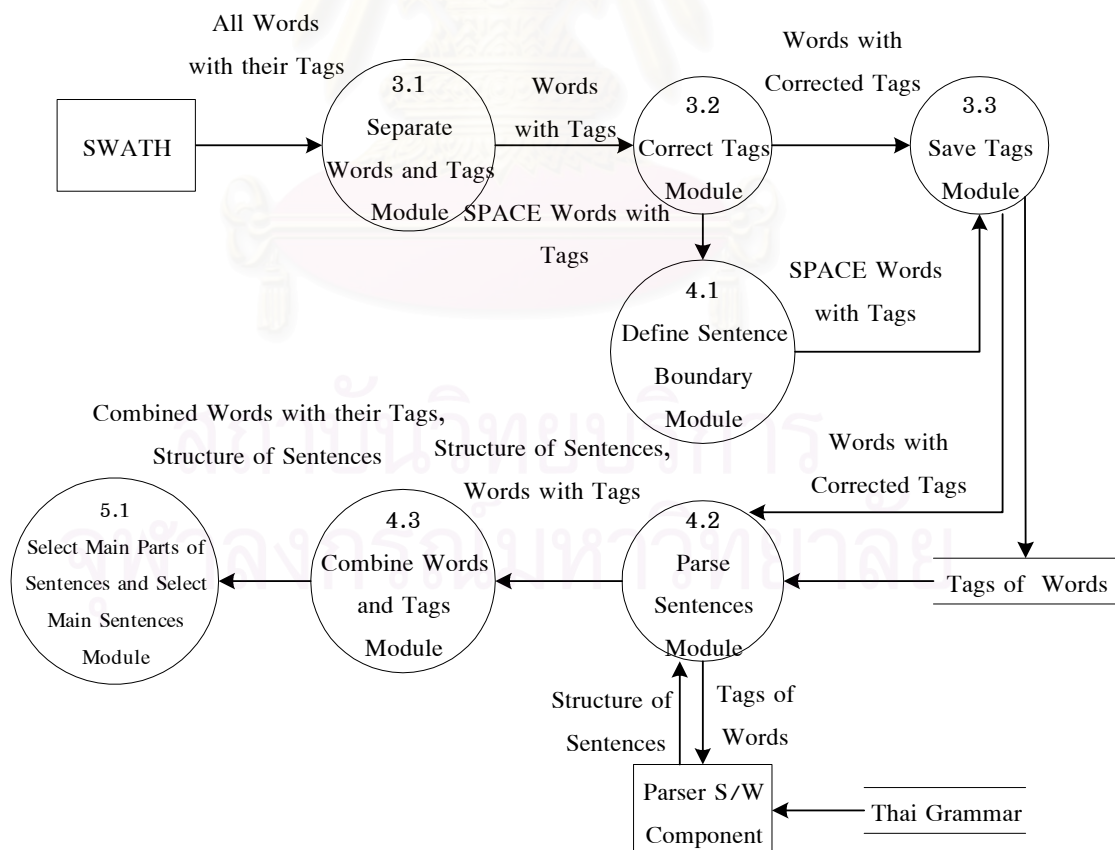
รูปที่ 4.3 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 1

รายละเอียดมอดูลการทำงานภายในของเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ 4.4 -

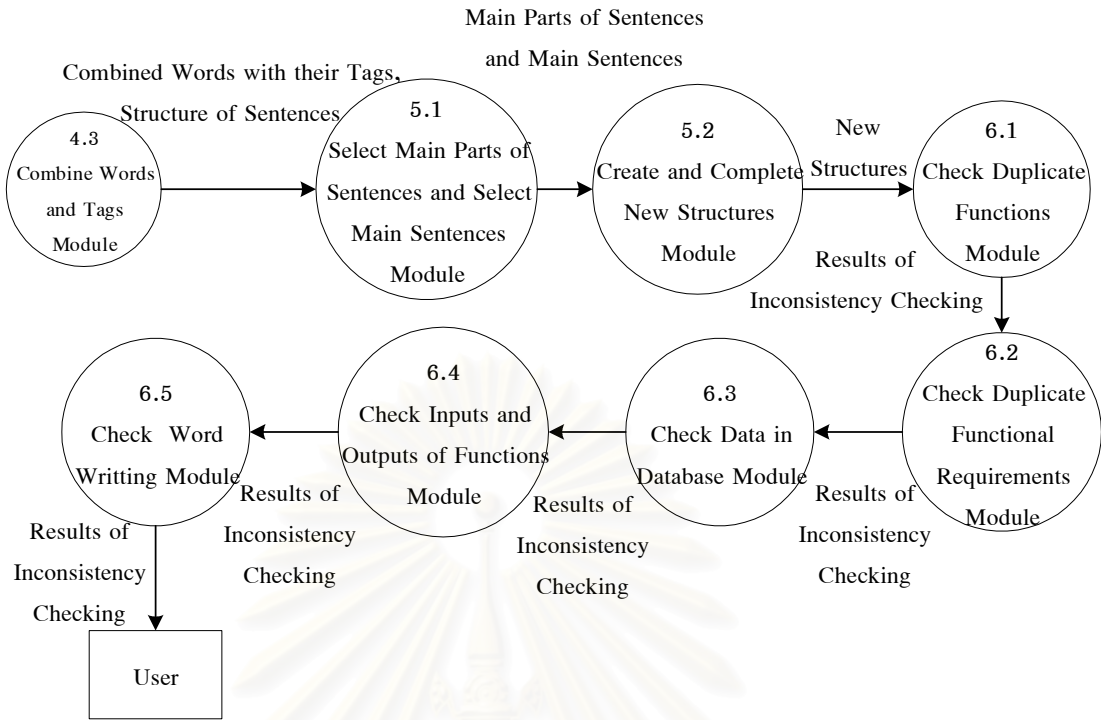
รูปที่ 4.7



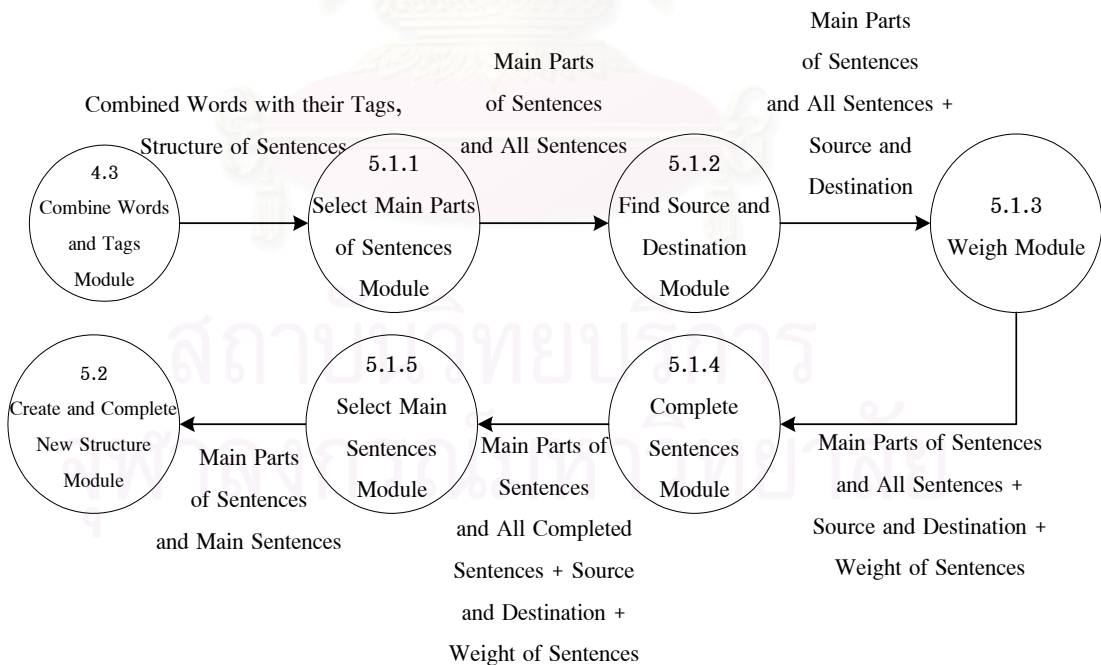
รูปที่ 4.4 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (มอดูลการทำงานของเอดิเตอร์ กับ กระบวนการเตรียมเอกสาร)



รูปที่ 4.5 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (การตัดคำและกำกับหน้าที่คำ กับ การวิเคราะห์ประโยค)



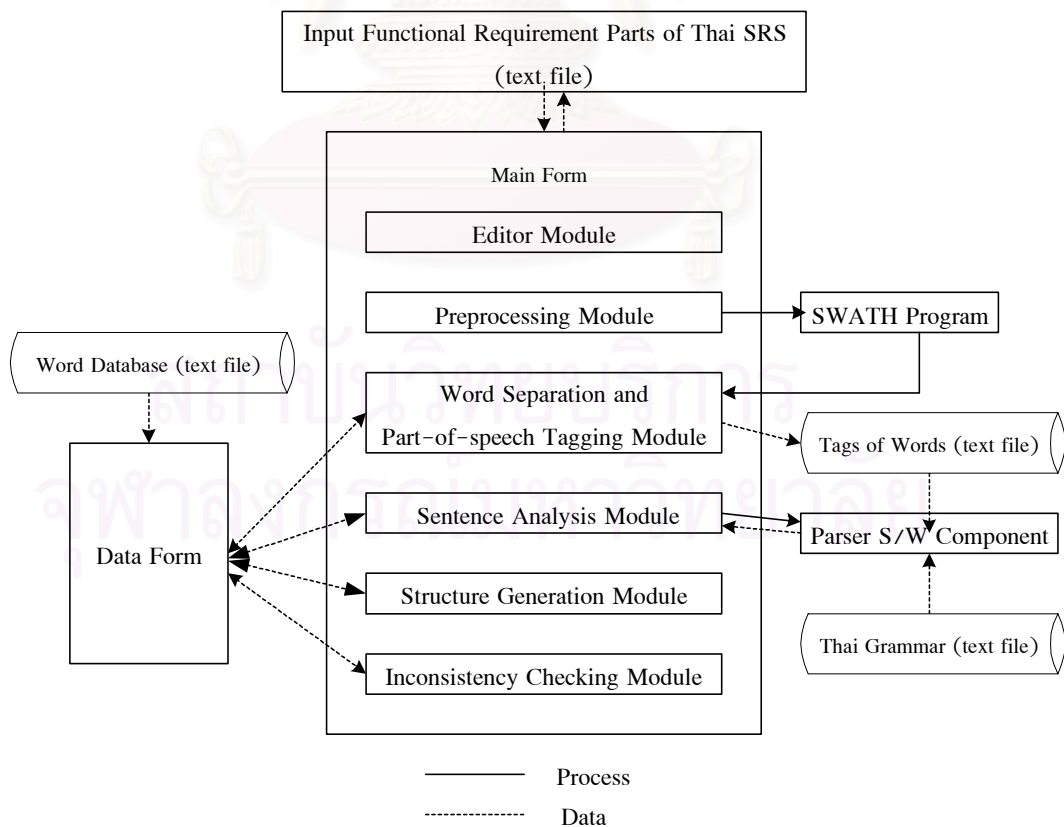
รูปที่ 4.6 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 2 (การสร้างรูปแบบใหม่ กับ การตรวจสอบความไม่ตรงกัน)



รูปที่ 4.7 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ระดับที่ 3 (การเลือกส่วนสำคัญของประโยคและเลือกประโยค)

4.1.1.2 การออกแบบเครื่องมือซอฟต์แวร์

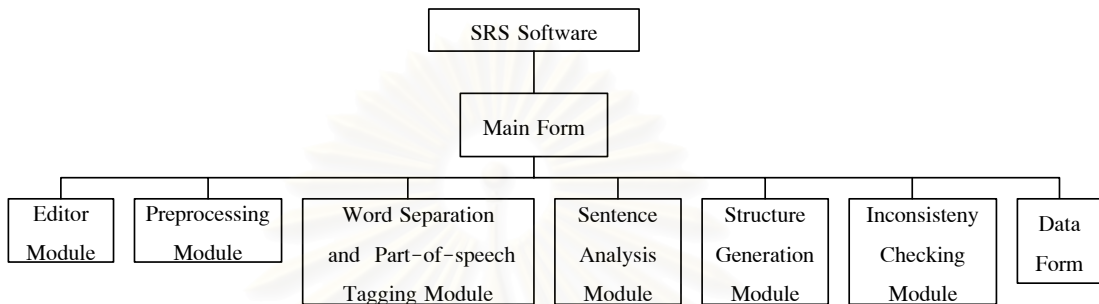
สถาปัตยกรรมภายในของเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ 4.8 โดยแสดงความสัมพันธ์ระหว่างส่วนต่าง ๆ ตามการเรียกใช้งาน และการส่งข้อมูลระหว่างกัน ประกอบด้วย ส่วนต่อประสานกับผู้ใช้ 2 ส่วน ได้แก่ หน้าจอการทำงานหลัก (Main Form) และหน้าจอสำหรับการเก็บข้อมูลระหว่างที่โปรแกรมทำงาน (Data Form) ในหน้าจอการทำงานหลักมีมอดูลการทำงานหลักของเครื่องมือซอฟต์แวร์ 6 มอดูล ได้แก่ มอดูลการทำงานของเอดิเตอร์ (Editor Module) มอดูลขั้นตอนการเตรียมเอกสาร (Preprocessing Module) มอดูลขั้นตอนการตัดคำและการกำกับหน้าที่คำ (Word Separation and Part-of-speech Tagging Module) มอดูลขั้นตอนการวิเคราะห์ประโยค (Sentence Analysis Module) มอดูลขั้นตอนการสร้างรูปแบบใหม่ (Structure Generation Module) และมอดูลขั้นตอนการตรวจสอบความไม่ตรงกัน (Inconsistency Checking Module) โดย 4 มอดูลหลังนั้น ระหว่างการทำงานจะมีการเก็บข้อมูลหรือเรียกใช้ข้อมูลจากหน้าจอเก็บข้อมูล ซึ่งหน้าจอเก็บข้อมูลจะดึงข้อมูลจากฐานข้อมูลคำศัพท์ (Word Database) ขึ้นมาเก็บไว้ด้วย นอกเหนือจากนั้นมอดูลหลักบางมอดูลมีการเรียกใช้งานส่วนอื่น ๆ ได้แก่ โปรแกรม SWATH สำหรับการตัดคำและกำกับหน้าที่คำ ส่วนประกอบซอฟต์แวร์สำหรับการวิเคราะห์ประโยคด้วยไวยากรณ์ และบางมอดูลมีการใช้งานข้อมูลจากฐานข้อมูล ซึ่งเป็นเพิ่มข้อมูลชนิดข้อความ หรือสร้างเพิ่มข้อมูลเพื่อเก็บข้อมูล



รูปที่ 4.8 สถาปัตยกรรมภายในของเครื่องมือซอฟต์แวร์

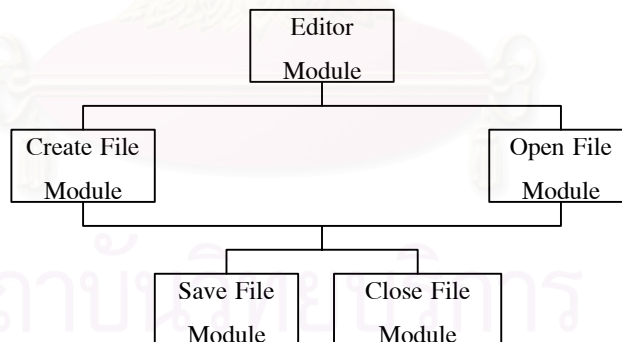
รายละเอียดในส่วนนี้ จะกล่าวถึงเฉพาะมอดูลที่สำคัญในการประมวลผลของ เครื่องมือซอฟต์แวร์ โดยมอดูลการทำงานจะออกแบบตามการวิเคราะห์ในรูปแบบที่ 4.3 – รูปที่ 4.7 แผนภาพการไหลของข้อมูลในการทำงานของเครื่องมือซอฟต์แวร์ และขั้นตอนวิธีการตรวจสอบความไม่ ตี้องกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ออกแบบในบทที่ 3

แผนภาพโครงสร้าง (Structure Chart) ของเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ 4.9

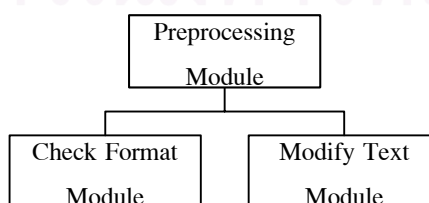


รูปที่ 4.9 แผนภาพโครงสร้างของเครื่องมือซอฟต์แวร์

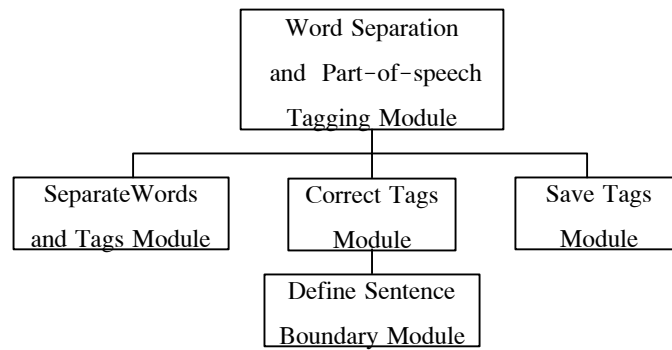
จากรูปที่ 4.9 ในแต่ละมอดูลทั้ง 6 มอดูล จะประกอบด้วยหลายมอดูลย่อย และมีการทำงานที่สัมพันธ์กัน เพื่อให้เห็นชัดเจน จึงเขียนเป็นแผนภาพโครงสร้างแยกแต่ละมอดูล ดังแสดงในรูปที่ 4.10 – รูปที่ 4.15 และจะมีการอธิบายรายละเอียดถัดไปในหัวข้อ 4.2 การพัฒนา เครื่องมือซอฟต์แวร์



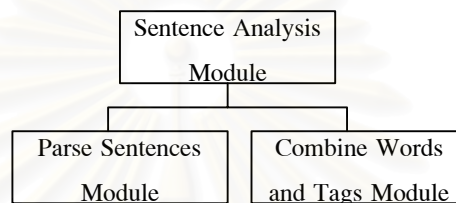
รูปที่ 4.10 แผนภาพโครงสร้างมอดูลการทำงานของเอดิเตอร์



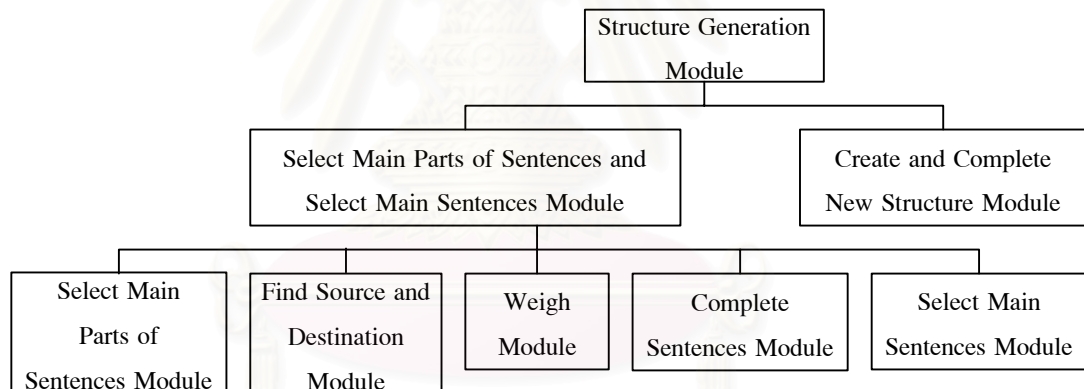
รูปที่ 4.11 แผนภาพโครงสร้างมอดูลขั้นตอนการเตรียมเอกสาร



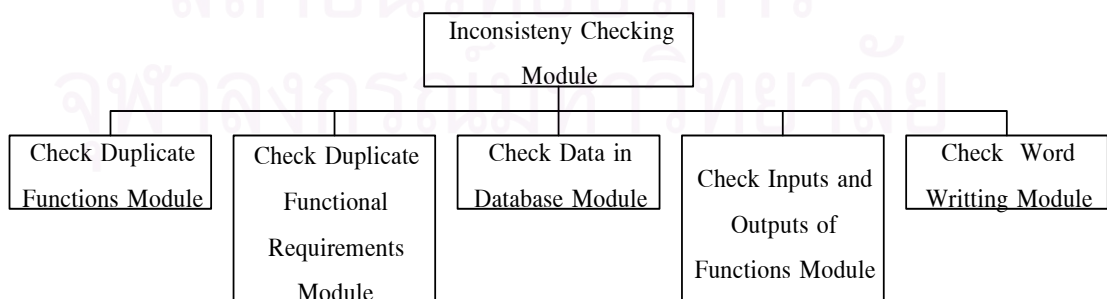
รูปที่ 4.12 แผนภาพโครงสร้างมอดูลขั้นตอนการตัดคำและการกำกับหน้าที่คำ



รูปที่ 4.13 แผนภาพโครงสร้างมอดูลขั้นตอนการวิเคราะห์ประโยค



รูปที่ 4.14 แผนภาพโครงสร้างมอดูลขั้นตอนการสร้างรูปแบบใหม่



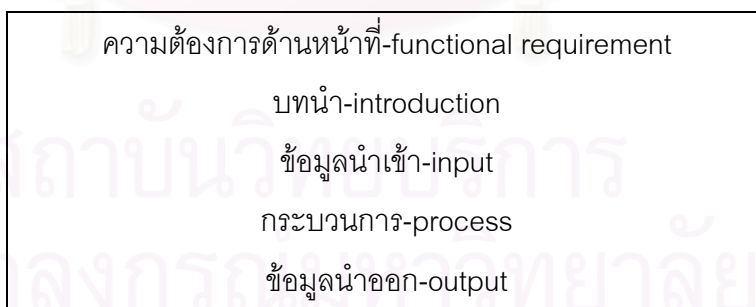
รูปที่ 4.15 แผนภาพโครงสร้างมอดูลการตรวจสอบความไม่ตรงกัน

4.1.2 การออกแบบส่วนฐานข้อมูล

ฐานข้อมูลในงานวิจัยนี้สร้างขึ้น เพื่อช่วยในการประมวลผลของเครื่องมือซอฟต์แวร์ประกอบด้วย ฐานข้อมูลค่านาม (ไทย, อังกฤษ) และคำเฉพาะ คำกริยา คำระบุแหล่งที่มาและแหล่งปลายทางของข้อมูล การจัดเก็บข้อมูลจัดเก็บในรูปแบบของแฟ้มข้อมูลชนิดข้อความ การพัฒนาฐานข้อมูลและฐานความรู้ที่ใช้ในเครื่องมือซอฟต์แวร์ เพื่อช่วยในการทำงาน ไม่มีการพัฒนาแยกออกมาอย่างชัดเจน เป็นเพียงแฟ้มข้อมูลที่เป็นชนิดข้อความ โดยมีมอดูลในเครื่องมือซอฟต์แวร์ทำหน้าที่ดึงข้อมูลในแฟ้มข้อมูลขึ้นมาแสดงผลทางหน้าจอได้ แต่การทำการแก้ไข เพิ่ม ลด เปลี่ยนแปลงข้อมูลในฐานข้อมูล จะใช้การแก้ไขโดยตรงจากแฟ้มข้อความ

4.1.2.1 ฐานข้อมูลค่านามและคำเฉพาะ

ฐานข้อมูลนี้เก็บรวบรวมค่านามซึ่งเป็นคำเฉพาะและคำภาษาอังกฤษที่หมายถึงค่านามเหล่านี้ ในงานวิจัยนี้ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยที่นำมาวิเคราะห์ ไม่มีการใช้คำเฉพาะ และแต่ละเอกสารข้อกำหนดความต้องการซอฟต์แวร์แต่ละเอกสารมักจะมีคำเฉพาะแตกต่างกันไป ดังนั้นฐานข้อมูลค่านามและคำเฉพาะในงานวิจัยนี้จึงยังไม่มีการเก็บข้อมูลคำเฉพาะไว้ ในฐานข้อมูลจึงประกอบด้วยค่านามเพียง 5 คำ แสดงดังรูปที่ 4.16 ซึ่งเป็นคำสำคัญที่ใช้ในการแบ่งส่วนสำหรับอธิบายรายละเอียดของความต้องการด้านหน้าที่เท่านั้น ถ้ามีการนำเครื่องมือซอฟต์แวร์สำหรับการตรวจสอบความไม่ตรงกันนี้ ไปใช้กับเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่มีการใช้คำเฉพาะ สามารถเพิ่มเติมคำเฉพาะเหล่านั้นลงในฐานข้อมูลได้



รูปที่ 4.16 ตัวอย่างฐานข้อมูลค่านามและคำเฉพาะ

4.1.2.2 ฐานข้อมูลคำกริยา

ฐานข้อมูลนี้เก็บรวบรวมคำกริยาที่แสดงถึงฟังก์ชันการทำงานของซอฟต์แวร์ ซึ่งมีลักษณะเป็นฐานความรู้ด้วย เนื่องจากมีรหัสตัวเลขที่แสดงถึงความหมายของคำกริยาเหล่านี้ แต่ยังไม่มีการเก็บคำศัพท์ภาษาอังกฤษที่หมายถึงกริยาเหล่านี้ ฐานข้อมูลนี้ประกอบด้วยข้อมูลคำกริยา 25 คำ ตัวอย่างแสดงดังรูปที่ 4.17

รหัสตัวเลขที่บอกความหมายของคำกริยาแต่ละรหัสจะคั่นด้วยเครื่องหมาย '-' รายละเอียดของรหัสตัวเลข คือ

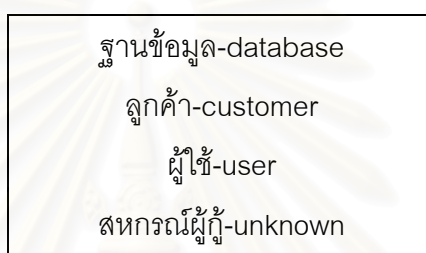
- 1) รหัสตัวเลขรหัสแรก จะแทนลักษณะของคำกริยาว่า คำนามที่พบใกล้กับคำกริยาเหล่านั้น จะเป็นคำนามที่แสดงถึงข้อมูลนำเข้าหรือข้อมูลนำออกของฟังก์ชันการทำงานของ โดย รหัส 1 หมายถึง คำนามนั้นมักจะเป็นข้อมูลนำออกเพียงอย่างเดียวของฟังก์ชันการทำงานของ รหัส 2 หมายถึง คำนามนั้นมักจะเป็นทั้งข้อมูลนำเข้าและข้อมูลนำออกของฟังก์ชันการทำงานของ และรหัส 3 หมายถึง คำนามนั้นมักจะเป็นข้อมูลนำออกไปยังฐานข้อมูลของฟังก์ชันการทำงานของ และเป็นข้อมูลนำเข้าจากฐานข้อมูลของฟังก์ชันการทำงานถัดไป
- 2) รหัสตัวเลขรหัสที่สอง จะแทนลักษณะของคำกริยาว่า คำนามที่แสดงถึงแหล่งที่มาของข้อมูลและแหล่งปลายทางของข้อมูลที่เกี่ยวข้องกับคำกริยาเหล่านั้น ควรจะเป็นแหล่งของข้อมูลแหล่งใด โดย รหัส 1 หมายถึง คำนามนั้นเป็นแหล่งที่มาของข้อมูล รหัส 2 หมายถึง คำนามนั้นเป็นแหล่งปลายทางของข้อมูล

เก็บ-2-2
รับ-2-1
สร้าง-1-2
แสดง-1-2
ค้นหา-3-2
ดึง-3-2
พิมพ์-2-2
ตรวจสอบ-2-1

รูปที่ 4.17 ตัวอย่างฐานข้อมูลคำกริยา

4.1.2.3 ฐานข้อมูลค่านามแสดงถึงแหล่งที่มาหรือแหล่งปลายทางของข้อมูล

ฐานข้อมูลนี้เก็บรวบรวมค่านามซึ่งแสดงถึงแหล่งที่มาหรือแหล่งปลายทางของข้อมูล ทั้งคำภาษาไทยและภาษาอังกฤษที่หมายถึงค่านั้น ซึ่งคำเหล่านี้จะเป็นสิ่งภายนอกที่เข้ามาเกี่ยวข้องกับระบบซอฟต์แวร์ โดยมักจะเป็นผู้ให้ข้อมูลนำเข้ามาในระบบ หรือเป็นผู้ที่ระบบแสดงผลข้อมูลออกไปให้ทราบ เช่น ผู้ใช้ ลูกค้า คำเหล่านี้ในแต่ละระบบซอฟต์แวร์จะมีความแตกต่างกันไป สำหรับงานวิจัยนี้ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ได้มาจะทำงานเกี่ยวข้องกับผู้ใช้และลูกค้าเท่านั้น ดังนั้นในฐานข้อมูลจึงมีเพียงแค่ 4 คำ แสดงดังรูปที่ 4.18



รูปที่ 4.18 ตัวอย่างฐานข้อมูลค่านามแหล่งที่มาหรือแหล่งปลายทางของข้อมูล

4.1.3 การออกแบบส่วนต่อประสานกับผู้ใช้

แบ่งส่วนต่อประสานกับผู้ใช้ได้เป็น 2 ส่วน คือ ส่วนหน้าจอกการทำงานหลัก และส่วนหน้าจอสำหรับการเก็บข้อมูลระหว่างที่โปรแกรมทำงาน รวมถึงแสดงข้อมูลในฐานข้อมูลคำศัพท์และฐานข้อมูลคำศัพท์ที่เป็นฐานความรู้ให้ผู้ใช้สามารถมองเห็นได้

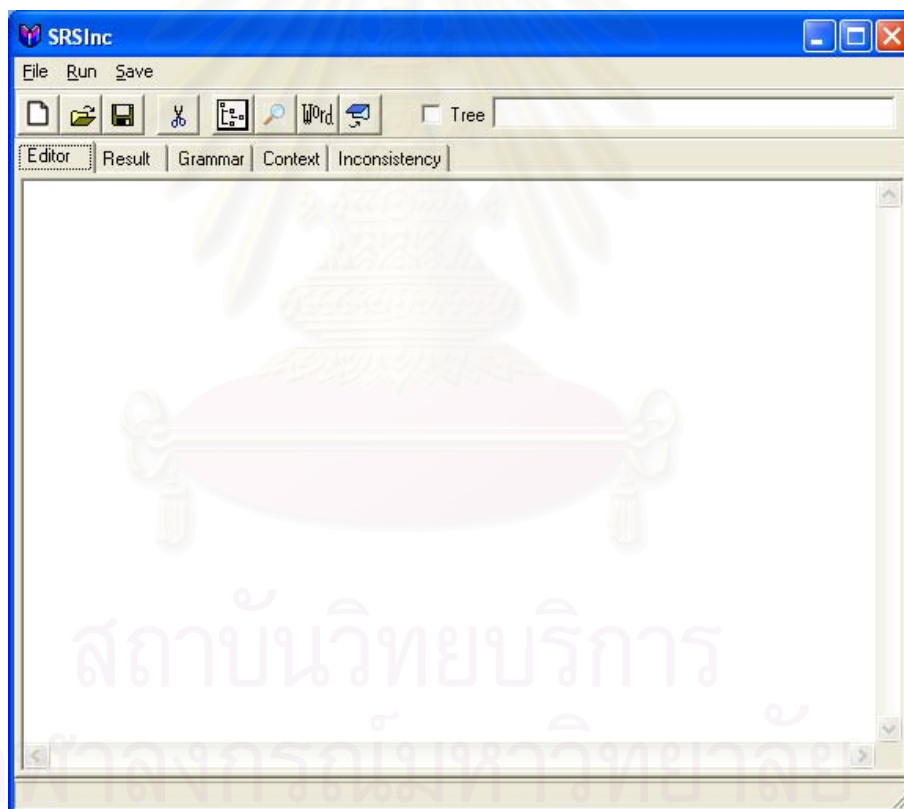
4.1.3.1 ส่วนหน้าจอกการทำงานหลัก

หน้าจอกการทำงานหลัก ประกอบด้วยหน้าจอกการทำงานย่อยอีก 5 ส่วนด้วยกัน โดยแบ่งด้วยแถบการทำงาน แสดงดังรูปที่ 4.19 โดยส่วนบนของหน้าจอจะเป็นแถบเมนู (Menu Bar) และแถบเครื่องมือ (Tool Bar) สำหรับควบคุมและสั่งการให้เครื่องมือซอฟต์แวร์ทำการประมวลผล

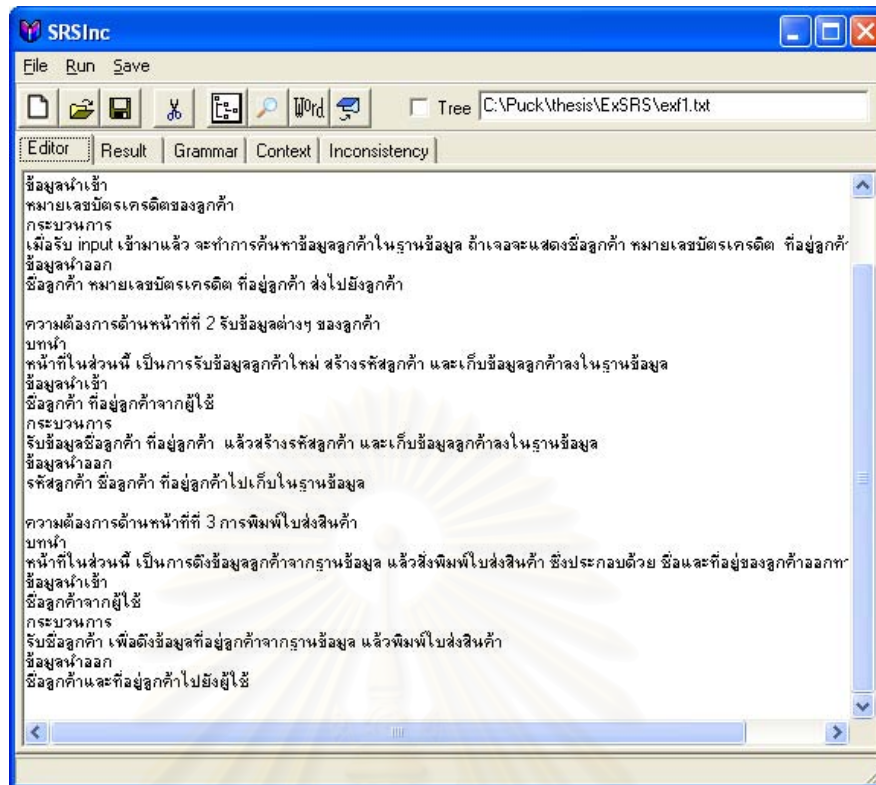
แถบการทำงาน (Tab) ทั้ง 5 แถบ ประกอบด้วย

- 1) แถบเอดิเตอร์ (Editor) เป็นส่วนที่สามารถพิมพ์ แก้ไข เพิ่ม ลบ เปลี่ยนแปลงเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ตัวอย่างแสดงดังรูปที่ 4.20
- 2) แถบรีซัลต์ (Result) เป็นส่วนที่แสดงผลลัพธ์ของการตัดคำและการกำกับหน้าที่คำ โดยแยกคำและหน้าที่ของคำให้เห็นชัดเจน ลักษณะการแสดงผลอยู่ในรูปแบบตาราง ตัวอย่างแสดงดังรูปที่ 4.21

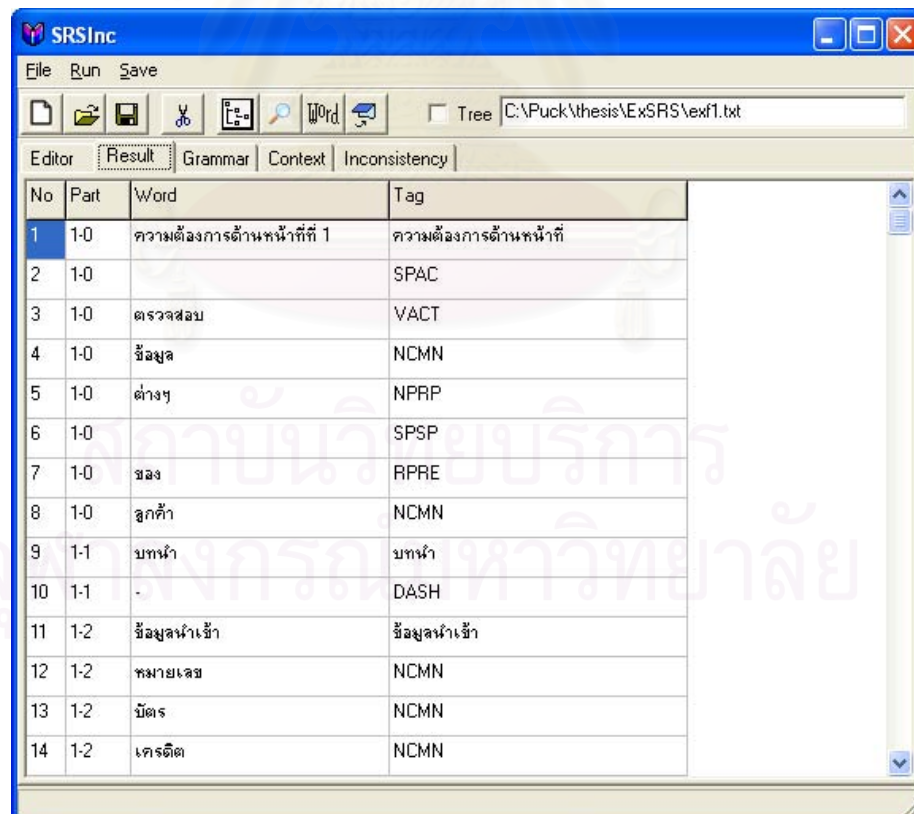
- 3) แถบแกรมมา (Grammar) เป็นส่วนที่แสดงโครงสร้างประโยค ซึ่งเป็นผลลัพธ์ของการวิเคราะห์ประโยคด้วยไวยากรณ์ภาษาไทย ลักษณะการแสดงผลมี 2 ลักษณะ คือ อยู่ในรูปข้อความหรืออยู่ในโครงสร้างรูปต้นไม้ ตัวอย่างแสดงดังรูปที่ 4.22 และ 4.23
- 4) แถบคอนเทค (Context) เป็นส่วนที่แสดงโครงสร้างใหม่ที่สร้างขึ้น เพื่ออธิบายแทนความต้องการด้านหน้าที่ของซอฟต์แวร์ แบ่งเป็น 2 ส่วน คือ โครงสร้างใหม่ที่ได้จากข้อมูลในเอกสาร และโครงสร้างใหม่ที่ปรับปรุงให้มีความสมบูรณ์แล้ว ตัวอย่างแสดงดังรูปที่ 4.24
- 5) แถบอินคอนซิสเทนซี (Inconsistency) เป็นส่วนที่แสดงผลการตรวจสอบความไม่ตรงกัน โดยจะแสดงข้อความบอกถึงความไม่ตรงกันที่พบในเอกสาร ตัวอย่างแสดงดังรูปที่ 4.25



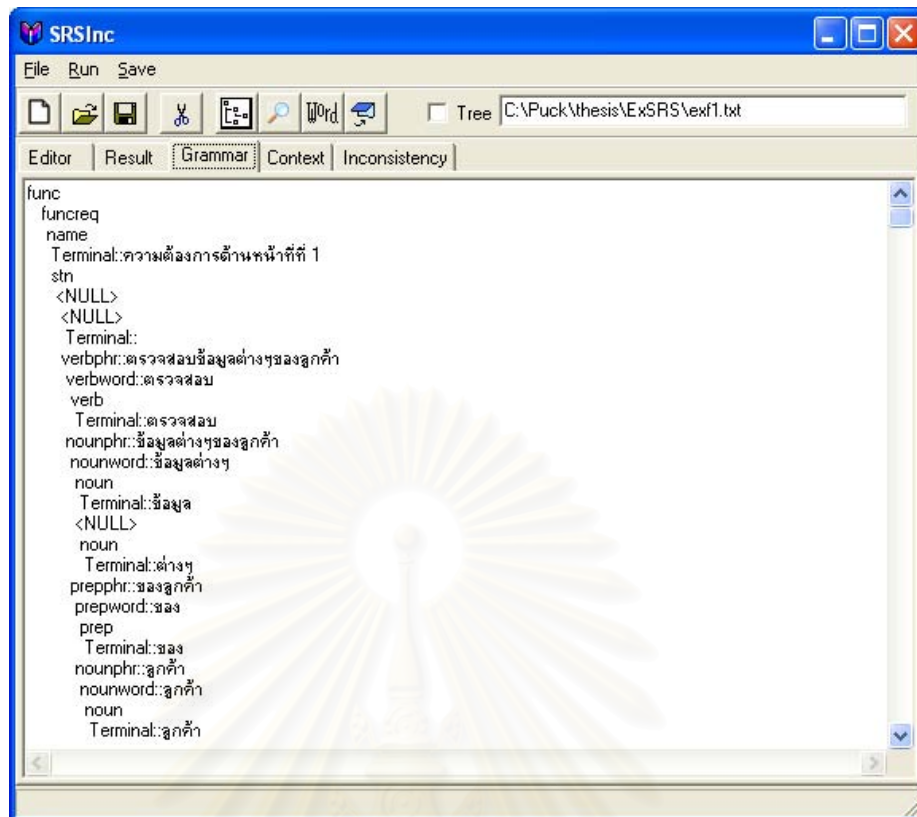
รูปที่ 4.19 หน้าจอการทำงานหลักของเครื่องมือซอฟต์แวร์



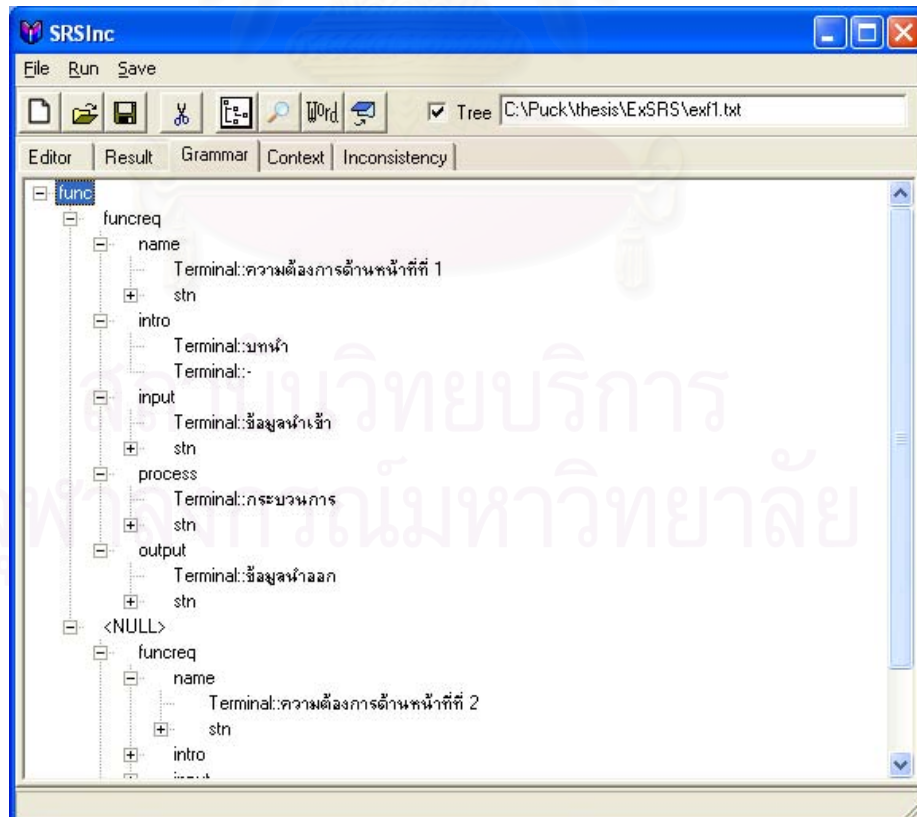
รูปที่ 4.20 แถบเอดิเตอร์ ในหน้าจอการทำงานหลัก



รูปที่ 4.21 แถบบริษัท ในหน้าจอการทำงานหลัก



รูปที่ 4.22 แถบแกรมมา (ลักษณะข้อความ) ในหน้าจอการทำงานหลัก



รูปที่ 4.23 แถบแกรมมา (ลักษณะโครงสร้างรูปต้นไม้) ในหน้าจอการทำงานหลัก

No	Usecas	Sour	Input	Act	Output	Dest.	Usecase	Source	Input	Action	Output	Dest.
1	หน้าที่ที่ 1	ลูกค้า	หมายเลข	รับ	หมายเลข		หน้าที่ที่ 1	ลูกค้า	หมายเลขบัตร	รับหมายเลข	หมายเลขบัตร	ฟังก์ชัน 1
2	หน้าที่ที่ 1		ค้นหา	ข้อมูล	ฐานข้อมูล		หน้าที่ที่ 1	ฟังก์ชัน 1.1	หมายเลขบัตร	ค้นหาข้อมูล	หมายเลขบัตร	ฐานข้อมูล
3	หน้าที่ที่ 1		แสดง	ข้อมูล	ลูกค้า		หน้าที่ที่ 1	ฐานข้อมูล	ข้อมูลลูกค้า	แสดงผลข้อมูล	ข้อมูลลูกค้า	หมายเลข
1	หน้าที่ที่ 2	ผู้ใช้	ข้อมูล	รับ	ข้อมูล		หน้าที่ที่ 2	ผู้ใช้	ข้อมูลข้อมูล	รับข้อมูล	ข้อมูล	ฟังก์ชัน 2
2	หน้าที่ที่ 2		สร้าง	รหัส			หน้าที่ที่ 2	ฟังก์ชัน 2.1	ข้อมูลข้อมูล	สร้างรหัส	ข้อมูล	ฟังก์ชัน 2
3	หน้าที่ที่ 2		ข้อมูล	เก็บ	ข้อมูล	ฐานข้อมูล	หน้าที่ที่ 2	ฟังก์ชัน 2.2	ข้อมูลข้อมูล	เก็บข้อมูล	ข้อมูล	ฐานข้อมูล
1	หน้าที่ที่ 3	ผู้ใช้	ข้อมูล	รับ	ข้อมูล		หน้าที่ที่ 3	ผู้ใช้	ข้อมูล	รับข้อมูล	ข้อมูล	ฟังก์ชัน 3
2	หน้าที่ที่ 3		ดึง	ข้อมูล	ฐานข้อมูล		หน้าที่ที่ 3	ฟังก์ชัน 3.1	ข้อมูล	ดึงข้อมูลที่	ข้อมูล	ฐานข้อมูล
3	หน้าที่ที่ 3		ใบส่ง	พิมพ์	ใบส่ง	ผู้ใช้	หน้าที่ที่ 3	ฐานข้อมูล	ข้อมูล	พิมพ์ใบส่ง	ข้อมูล	ผู้ใช้

รูปที่ 4.24 แถบคอนเทคในหน้าจการทำงานหลัก

ไม่พบความไม่ตรงกันของฐานข้อมูล
ไม่พบความไม่ตรงกันของข้อมูลนำเข้าและข้อมูลส่งออกระหว่างฟังก์ชัน

หน้าที่ที่ 1: ข้อมูลลูกค้า หมายถึง ข้อมูลลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้า
หน้าที่ที่ 2: ข้อมูลลูกค้า หมายถึง รหัสลูกค้า ข้อมูลลูกค้า ที่อยู่ลูกค้า
หน้าที่ที่ 3: ใบส่งสินค้า ประกอบด้วย ข้อมูลลูกค้า ข้อมูลที่อยู่ลูกค้า

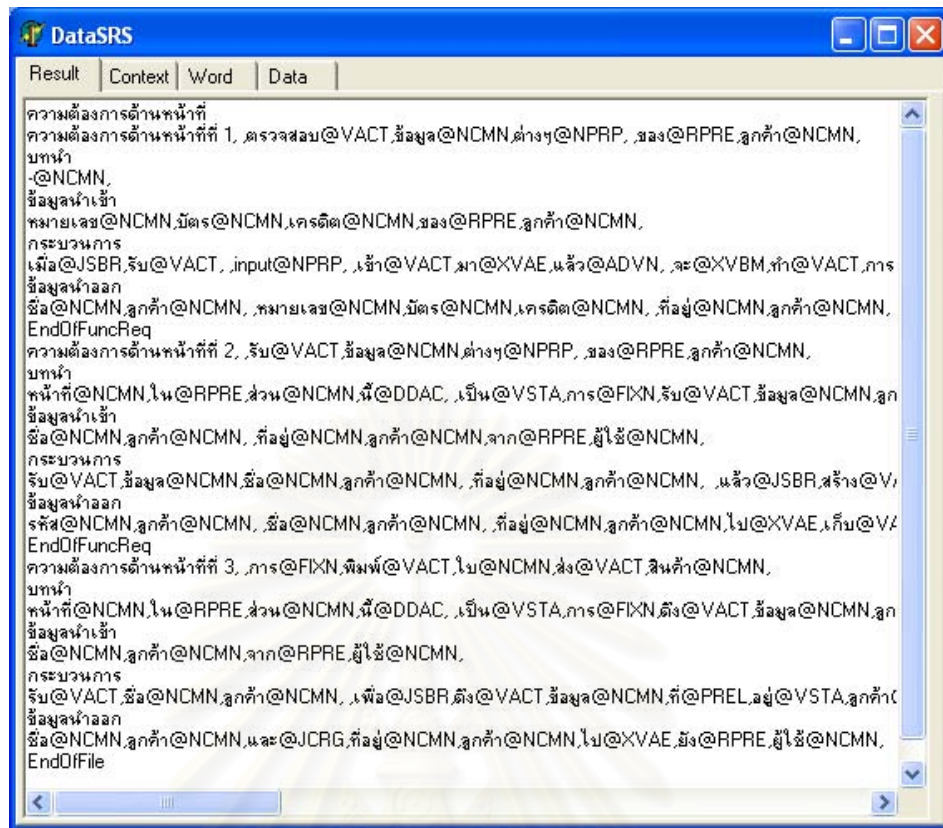
รูปที่ 4.25 แถบอินคอนซิสเทนซีในหน้าจการทำงานหลัก

4.1.3.2 ส่วนหน้าจอสำหรับเก็บข้อมูลระหว่างการทำงาน และส่วนฐานข้อมูล

ส่วนนี้เป็นเหมือนที่เก็บข้อมูลต่าง ๆ ระหว่างที่เครื่องมือซอฟต์แวร์ทำการประมวลผล ซึ่งผู้ใช้งานจะไม่สามารถเข้ามาทำการแก้ไขข้อมูลใด ๆ ได้ แต่สามารถเห็นผลการทำงานระหว่างที่เครื่องมือซอฟต์แวร์ทำการประมวลผลได้อย่างชัดเจน โดยหน้าจอนี้จะแสดงขึ้น เมื่อกดเลือกปุ่ม Word จากหน้าจอการทำงานหลัก

ส่วนของหน้าจอสำหรับเก็บข้อมูลนี้ประกอบด้วยส่วนย่อย 4 ส่วน โดยใช้แถบการทำงานเป็นตัวแบ่ง ประกอบด้วย

- 1) แถบรีซัลต์ (Result) เป็นส่วนที่แสดงผลลัพธ์ของการตัดคำและการกำกับหน้าที่คำ ซึ่งเป็นผลลัพธ์โดยตรงจากโปรแกรมตัดคำภาษาไทยที่ผู้วิจัยนำโปรแกรมที่มีผู้พัฒนาไว้แล้วมาใช้งาน ผลลัพธ์นี้ยังไม่มีปรับปรุงแก้ไขให้มีความถูกต้องมากขึ้น ตัวอย่างแสดงดังรูปที่ 4.26
- 2) แถบคอนเทค (Context) เป็นส่วนที่แสดงผลลัพธ์ของคำที่มีการแก้ไขการตัดคำหรือการกำกับหน้าที่คำแล้ว และแสดงให้เห็นส่วนประกอบที่สำคัญของประโยค และโครงสร้างสำคัญที่ได้จากประโยคทั้งหมด รวมถึงลำดับของคำหรือวลีที่อยู่ในประโยคและหน้าที่คำหรือวลีนั้น โดยประโยคเหล่านี้จะแยกเก็บตามส่วนที่พบประโยคในความต้องการด้านหน้าที่ ได้แก่ ส่วนชื่อ บทนำ ข้อมูลนำเข้า กระบวนการ และข้อมูลนำออก ก่อนที่จะนำไปประมวลผลหาโครงสร้างใหม่ ตัวอย่างแสดงดังรูปที่ 4.27
- 3) แถบเวิร์ด (Word) เป็นส่วนที่แสดงให้เห็นข้อมูลในฐานข้อมูลคำศัพท์และฐานข้อมูลที่เป็นฐานความรู้ในเครื่องมือซอฟต์แวร์ ตัวอย่างแสดงดังรูปที่ 4.28
- 4) แถบดาต้า (Data) เป็นส่วนที่แสดงให้เห็นข้อมูลและคำที่ใช้ในการเขียนเอกสารเมื่อกล่าวถึงข้อมูลนั้น ทั้งข้อมูลที่เกี่ยวข้องกับฐานข้อมูล และข้อมูลที่เป็นข้อมูลนำเข้า และข้อมูลนำออกของฟังก์ชันการทำงาน ในความต้องการด้านหน้าที่ของซอฟต์แวร์ ตัวอย่างแสดงดังรูปที่ 4.29



รูปที่ 4.26 แถบรีซัลในหน้าจอการเก็บข้อมูล

The screenshot shows the DataSRS interface with a detailed view of the data. The tables display columns for No, Fun, Part, Word, Tag, Vact, Noun, Freq, V, S, V, O, T, and V. The data is organized into several sections, each with its own table header.

No	Fun	Part	Word	Tag
3	1	0	ตรวจสอบ	VACT
5	1	0	ข้อมูลต่างๆ	NCMN
7	1	0	สอง	RPPE
8	1	0	ลูกค้า	NCMN
14	1	2	หมายเลขมีตร	NCMN
15	1	2	สอง	RPPE
16	1	2	ลูกค้า	NCMN
19	1	3	รับ	VACT
21	1	3	input	NPRP
24	1	3	เข้ามา	XVAE
28	1	3	จะทำ	XVBM
30	1	3	ค้นหา	VACT
32	1	3	ข้อมูลลูกค้า	NCMN
33	1	3	ใน	RPPE
34	1	3	ฐานข้อมูล	NCMN
38	1	3	จะ	XVBM

No	Vact	Noun	Freq.	V
8	รับ	input	0	4
12	ค้นหา	ข้อมูล	0	4
1	ตรวจสอบ	ข้อมูล	0	2
27	รับ	ข้อมูล	0	2
35	รับ	ข้อมูล	0	3
27	ค้นหา	ข้อมูล	0	2
16		หมายเลข		
103		ข้อมูล		
24	ส่งไป	ลูกค้า	0	1
63		ข้อมูล		

No	S	V	O	T	V
1	หมายเลขรับ	หมายเลข	2	4	
1		ค้นหา	ข้อมูล	3	4
1		แสดง	ข้อมูล	1	4
1	ข้อมูล	ส่งไป	ลูกค้า	2	C
2	ข้อมูล	รับ	ข้อมูล	2	4
2		สร้าง	รหัส	1	E
2		เก็บ	ข้อมูล	2	E
2	รหัส	ไปเก็บ	ฐาน	2	C
2		รับ	ข้อมูล	2	C
3	ข้อมูล	รับ	ข้อมูล	2	4
3		ดึง	ข้อมูล	3	4
3		พิมพ์	ไปส่ง	2	1
3		ดึง	ข้อมูล	3	C

รูปที่ 4.27 แถบคอนเทคในหน้าจอการเก็บข้อมูล

The screenshot shows the DataSRS application window with the 'Data' tab selected. It displays three tables of linguistic data:

No	Verb	RoleN
1	เก็บ	2
2	รับ	2
3	สร้าง	1
4	แสดง	1
5	ค้นหา	3
6	ตั้ง	3
7	พิมพ์	2
8	ตรวจสอบ	2
9	ลบ	1
10	ชำระ	1
11	ทำ	1
12	สั่งซื้อ	2
13	บันทึก	2
14	จัดซื้อ	2
15	รายงาน	2
16	ปรับปรุง	2

No	NounTh	NounEn
1	ความต้องการ	functional req
2	บทนำ	introduction
3	ข้อมูลนำเข้า	input
4	กระบวนการ	process
5	ข้อมูลนำออก	output

No	AgentTh	AgentEn
1	ฐานข้อมูล	database
2	ลูกค้า	customer
3	ผู้ใช้	user
4	สภกรณผู้ดี	unknown

รูปที่ 4.28 แถบเวิร์ดในหน้าจอการเก็บข้อมูล

The screenshot shows the DataSRS application window with the 'Data' tab selected. It displays a detailed data table with the following columns: Input, Output, Location, F, Func, Start, End, Data, Func, Start, End, Data, F.

Input	Output	Locate	F	Func	Start	End	Data	Func	Start	End	Data	F
หมายเลขเครื่อง	ข้อมูลลูกค้า	1.3	1	ฟังก์ชัน 1.2	1	1	หมายเลข	ฟังก์ชัน 1.1	1	1	หมายเลขบัตร	1
รหัสลูกค้า	หมายเลข	1.3	1	ฟังก์ชัน 2.2	2	3	ข้อมูลซื้อ	ฟังก์ชัน 2.1	2	3	ข้อมูลซื้อลูกค้า	1
ข้อมูลลูกค้า	ที่อยู่ลูกค้า	1.3	1	ฟังก์ชัน 2.2			ที่อยู่ลูกค้า	ฟังก์ชัน 2.1			ที่อยู่ลูกค้า	1
ที่อยู่ลูกค้า				ฟังก์ชัน 2.3	4	6	ข้อมูลซื้อ	ฟังก์ชัน 2.2	4	6	รหัสลูกค้า	1
				ฟังก์ชัน 2.3			ที่อยู่ลูกค้า	ฟังก์ชัน 2.2			ข้อมูลลูกค้า	1
				ฟังก์ชัน 2.3			รหัสลูกค้า	ฟังก์ชัน 2.2			ที่อยู่ลูกค้า	1
				ฟังก์ชัน 3.2	7	7	ข้อมูลลูกค้า	ฟังก์ชัน 3.1	7	7	ข้อมูลลูกค้า	1

รูปที่ 4.29 แถบตาข่าย ในหน้าจอการเก็บข้อมูล

4.2 การพัฒนาเครื่องมือซอฟต์แวร์

เครื่องมือซอฟต์แวร์นี้ พัฒนาระบบปฏิบัติการวินโดวส์ (Windows) มีการสร้างส่วนต่อประสานกับผู้ใช้ทำให้ง่ายต่อการใช้งาน ผู้ใช้สามารถทำความเข้าใจ และใช้งานได้อย่างรวดเร็ว เครื่องมือที่ใช้พัฒนาเครื่องมือซอฟต์แวร์ในงานวิจัยนี้ คือ บอร์แลนด์ เดลไฟล์ (Borland Delphi) เวอร์ชัน 5.0 ซึ่งมีการเพิ่มส่วนประกอบซอฟต์แวร์ (Software Component) ที่ช่วยในการวิเคราะห์ด้วยไวยากรณ์ รวมถึงมีส่วนฐานข้อมูลและฐานข้อมูลที่เป็นฐานความรู้จะจัดเก็บอยู่ในรูปแฟ้มข้อมูลชนิดข้อความ ที่สามารถทำการแก้ไข ปรับปรุง ให้ตรงตามความต้องการของผู้ใช้ได้

รายละเอียดในการพัฒนาเครื่องมือซอฟต์แวร์ จะกล่าวถึงมอดูลการทำงานของเครื่องมือซอฟต์แวร์ ซึ่งประกอบด้วย

4.2.1 มอดูลการทำงานของเอดิเตอร์

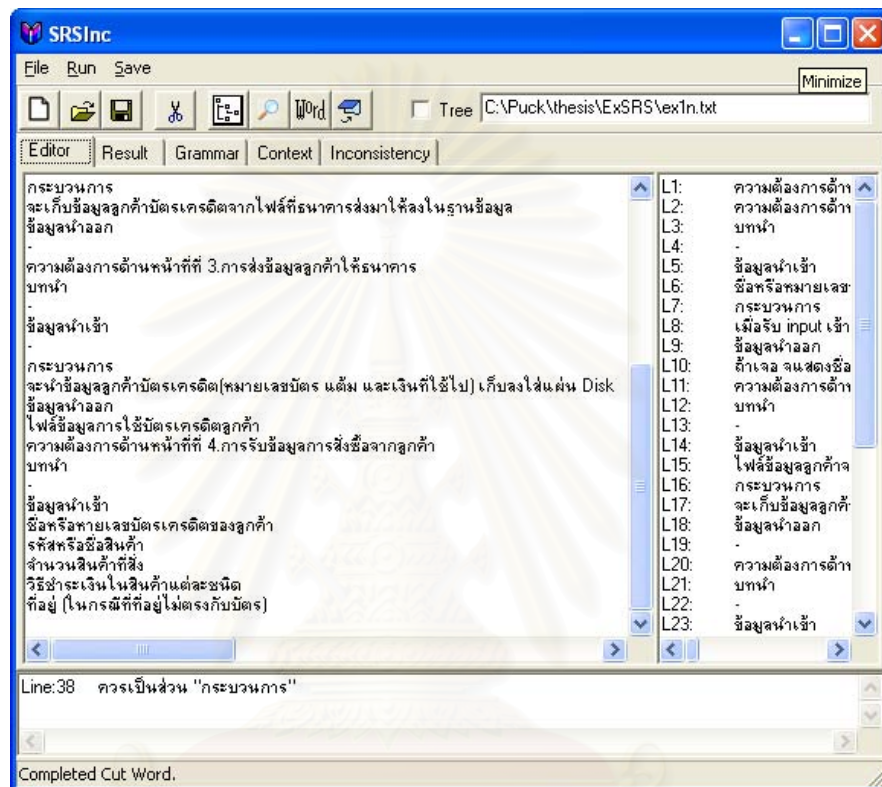
ส่วนนี้เป็นส่วนนำเข้าเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ต้องการตรวจสอบความไม่ตรงกัน และการทำงานต่าง ๆ ที่เกี่ยวข้องกับแฟ้มข้อมูลเอกสารที่เป็นข้อมูลนำเข้า การทำงานของเอดิเตอร์ประกอบด้วยมอดูล 4 มอดูล คือ การเปิดแฟ้มข้อมูลที่เก็บเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่มีอยู่แล้วเพื่อนำมาประมวลผล การสร้างแฟ้มข้อมูลเอกสารใหม่ การบันทึกแฟ้มข้อมูลเอกสาร และการปิดแฟ้มข้อมูลเอกสาร

4.2.2 มอดูลขั้นตอนการเตรียมเอกสาร

ก่อนที่เครื่องมือซอฟต์แวร์จะทำการประมวลผล จะมีการเตรียมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นข้อมูลนำเข้าให้มีรูปแบบตรงตามมาตรฐาน และมีลักษณะเหมาะสมที่จะเป็นข้อมูลนำเข้าของโปรแกรมตัดคำภาษาไทยที่งานวิจัยนี้เลือกมาใช้ ในขั้นตอนนี้ประกอบด้วย มอดูลทั้งหมด 2 มอดูล คือ มอดูลการตรวจสอบรูปแบบของเอกสาร และมอดูลการแก้ไขเปลี่ยนแปลงข้อความในเอกสารโดยอัตโนมัติ รายละเอียดดังต่อไปนี้

- 1) มอดูลการตรวจสอบรูปแบบของเอกสาร จะทำการตรวจสอบเอกสารว่าตรงตามรูปแบบมาตรฐานของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ส่วนความต้องการด้านหน้าที่ที่ผู้วิจัยเลือกใช้หรือไม่ โดยส่วนความต้องการด้านหน้าที่จะประกอบด้วยส่วนสำคัญ 5 ส่วนด้วยกัน คือ ชื่อความต้องการด้านหน้าที่ บทนำ ข้อมูลนำเข้า กระบวนการ และข้อมูลนำออก หากเอกสารที่เป็นข้อมูลนำเข้ามีรูปแบบไม่ตรงตามมาตรฐาน จะมีข้อความแสดงถึงข้อผิดพลาดและบรรทัดที่ข้อผิดพลาดนั้นเกิดขึ้น ตัวอย่างแสดงดังรูปที่ 4.30 และเครื่องมือซอฟต์แวร์จะไม่สามารถทำงานต่อ จนกว่าจะแก้ไขรูปแบบเอกสารที่ไม่ตรงตามมาตรฐานนั้นเรียบร้อยแล้ว รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-1

- 2) มอดูลการแก้ไขเปลี่ยนแปลงข้อความในเอกสารโดยอัตโนมัติ จะมีการแก้ไขและลบอักขระบางตัวที่ผลต่อการทำงานของโปรแกรมตัดคำออกจากเอกสาร และจึงส่งข้อความที่แก้ไขอักขระเหล่านั้นเรียบร้อยแล้ว ในรูปแบบของสายข้อความที่ยาวต่อกัน (Stream Text) เป็นข้อมูลนำเข้า ไปยังโปรแกรมตัดคำภาษาไทย รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก จ. รูปที่ จ-2



รูปที่ 4.30 หน้าจอแสดงข้อความผิดพลาด เมื่อเอกสารไม่ตรงตามมาตรฐาน

4.2.3 มอดูลขั้นตอนการตัดคำและการกำกับหน้าที่คำ

ขั้นตอนนี้เป็นขั้นตอนการรับผลลัพธ์จากโปรแกรมตัดคำ ซึ่งอยู่ในรูปของสายข้อความที่ยาวต่อกันส่งออกมา โดยมีอักขระพิเศษเป็นตัวคั่นระหว่างคำและหน้าที่ของคำ แล้วนำผลลัพธ์นี้มาทำการปรับปรุงแก้ไขให้คำและหน้าที่ของคำมีความถูกต้องมากขึ้น ก่อนส่งคำและหน้าที่คำไปประมวลผลในขั้นตอนการวิเคราะห์ประโยค ขั้นตอนนี้ประกอบด้วยมอดูลทั้งหมด 3 มอดูล คือ มอดูลการรับข้อมูลของผลลัพธ์จากโปรแกรมตัดคำแล้วแยกเก็บส่วนคำและหน้าที่ของคำ มอดูลการแก้ไขหน้าที่คำ และมอดูลการจัดเก็บหน้าที่คำในรูปแบบแฟ้มข้อมูล มีรายละเอียดดังต่อไปนี้

- 1) มอดูลการรับข้อมูลของผลลัพธ์จากโปรแกรมตัดคำ หลังจากศึกษาผลลัพธ์ของโปรแกรมตัดคำ ศึกษาตัวอักขระพิเศษที่โปรแกรมใช้เป็นตัวแบ่งคำและหน้าที่คำ มอดูลนี้จะทำการแยกส่วนคำและหน้าที่คำมาจัดเก็บในเครื่องมือซอฟต์แวร์ โดยมอดูลนี้

จะทำงาน 3 หน้าที่ระหว่างการแยกส่วนคำและจัดเก็บ (รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-3) คือ

- ตรวจสอบคำสำคัญที่แสดงถึงส่วนต่าง ๆ ของความต้องการด้านหน้าที่แล้วรวม คำสำคัญเหล่านั้นเข้าเป็นคำเดิม
- ตรวจสอบคำเฉพาะที่พบในฐานข้อมูลคำศัพท์ แล้วรวมเป็นคำเดิม
- เพิ่มการกำกับหน้าที่คำให้กับ เว้นวรรค ซึ่งโปรแกรมตัดคำจะส่งข้อมูลออกมา เป็นแค่ช่องว่าง โดยสร้างหน้าที่คำเพิ่มเติมจากหน้าที่ของคำ 47 หน้าที่ของคลัง ข้อมูลออร์คิด ตัวอักษรเว้นวรรคจะมีหน้าที่คำเป็น "SPAC" เป็นส่วนที่แสดงให้ ทราบว่าคำนี้คือ เว้นวรรค

2) มอดูลการแก้ไขหน้าที่คำ มีการทำงาน 2 หน้า (รายละเอียดชุดคำสั่งแสดงใน ภาค ผนวก ฉ. รูปที่ ฉ-4) คือ

- แก้ไขหน้าที่คำของเครื่องหมายวรรคตอนให้มีความถูกต้อง
- เพิ่มการกำกับหน้าที่คำให้กับ เครื่องหมายยัติภังค์ โดยสร้างหน้าที่คำเพิ่มเติม จากหน้าที่ของคำ 47 หน้าที่ของคลังข้อมูลออร์คิด คือ แดช

3) มอดูลการจัดเก็บหน้าที่คำในรูปแบบแฟ้มข้อมูล แยกเฉพาะหน้าที่คำออกมาเก็บเป็นแฟ้ม ข้อมูลชนิดข้อความ เพื่อส่งเป็นแฟ้มข้อมูลนำเข้าสำหรับการวิเคราะห์ประโยคด้วย ไวยากรณ์ภาษาไทย

4.2.4 มอดูลขั้นตอนการวิเคราะห์ประโยค

ขั้นตอนนี้ประกอบด้วยมอดูลทั้งหมด 3 มอดูล คือ มอดูลการกำหนดขอบเขตของประโยค มอดูลการวิเคราะห์ประโยคด้วยไวยากรณ์ภาษาไทย และมอดูลการรวมคำและแก้ไขหน้าที่คำด้วย ไวยากรณ์ ซึ่งในการประมวลผลจริงของเครื่องมือซอฟต์แวร์ ทั้งสามมอดูลนี้ไม่ได้ทำงานเรียงกัน ตามลำดับ เหมือนมอดูลในขั้นตอนที่กล่าวมาแล้ว เนื่องจากการวิเคราะห์ประโยคจะมีส่วนที่เกี่ยวข้อง ข้องคาบเกี่ยวกับการตัดคำและการสร้างรูปแบบโครงสร้างใหม่ ดังนั้นบางมอดูลในขั้นตอนนี้ จะ ทำงานแทรกอยู่ในขั้นตอนการทำงานขั้นอื่น มีรายละเอียดดังต่อไปนี้

- 1) มอดูลการกำหนดขอบเขตของประโยค ในการประมวลผลจริงมอดูลนี้จะทำงานแทรก อยู่ในขั้นตอนการตัดคำ โดยจะทำงานก่อนมอดูลที่จัดเก็บหน้าที่คำในรูปแบบแฟ้มข้อมูล การทำงานของมอดูลนี้จะกำจัดเว้นวรรคที่ไม่ใช่จุดสิ้นสุดของประโยคออกไป โดย เปลี่ยนหน้าที่คำของเว้นวรรคที่ไม่เกี่ยวข้องเหล่านี้จาก "SPAC" เป็น "SPSP" เพื่อไม่ให้ เว้นวรรคเหล่านี้ถูกจัดเก็บเข้าในแฟ้มข้อมูลของหน้าที่คำ ใช้การพิจารณาเว้นวรรค ตามที่กล่าวไปแล้วในหัวข้อ 3.2.3.2 รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-5

- 2) มอดูลการวิเคราะห์ส่วนประกอบของประโยคด้วยไวยากรณ์ภาษาไทย จะรับข้อมูลนำเข้าเป็นหน้าที่คำซึ่งอยู่ในแฟ้มข้อมูลและไวยากรณ์ภาษาไทย นำมาวิเคราะห์ประโยคตามไวยากรณ์ มอดูลนี้จะใช้ส่วนประกอบซอฟต์แวร์ที่เพิ่มเข้าไปมาช่วยในการวิเคราะห์ประโยคตามไวยากรณ์ ถ้าประโยคไม่ตรงตามหลักไวยากรณ์จะมีข้อความแสดงข้อผิดพลาด และจะไม่สามารถทำการประมวลผลในขั้นตอนต่อไปได้ หลังจากการวิเคราะห์ตามไวยากรณ์ถ้าประโยคตรงตามไวยากรณ์ มอดูลนี้จะมีการเรียกมอดูลย่อย เพื่อเชื่อมความสัมพันธ์ระหว่างหน้าที่คำที่นำไปวิเคราะห์กับคำเดิม และแสดงผลของคำ วลี หรือข้อความเดิมตามส่วนประกอบของประโยคที่วิเคราะห์ได้ ผลลัพธ์ที่ได้จากการวิเคราะห์ประโยคจะแสดงผลดังรูปที่ 4.22 หรือรูปที่ 4.23
- 3) มอดูลการรวมคำและแก้ไขหน้าที่คำด้วยไวยากรณ์ ในการประมวลผลจริงส่วนหนึ่งของมอดูลนี้ จะทำงานรวมอยู่ในมอดูลการแก้ไขหน้าที่คำในขั้นตอนการตัดคำและการกำกับหน้าที่คำ (รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-4) และอีกส่วนหนึ่งทำงานรวมอยู่กับมอดูลในขั้นตอนการสร้างรูปแบบใหม่ (รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-6)

4.2.5 มอดูลขั้นตอนการสร้างรูปแบบโครงสร้างใหม่

ขั้นตอนนี้พิจารณาข้อมูลทั้งหมดที่ได้จากการประมวลผลในขั้นที่ผ่านมาแล้วนำมาสร้างโครงสร้างใหม่ ประกอบด้วยมอดูลทั้งหมด 2 มอดูล คือ มอดูลการเลือกส่วนสำคัญของประโยค และเลือกประโยคที่จะนำไปสร้างโครงสร้างใหม่ มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมข้อมูลเพื่อให้โครงสร้างใหม่สมบูรณ์ขึ้น มีรายละเอียดดังต่อไปนี้

- 1) มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยคที่จะนำไปสร้างโครงสร้างใหม่ ซึ่งภายในมอดูลนี้จะมีการเรียกใช้มอดูลย่อยอื่น ๆ อีก 5 มอดูล ซึ่งมอดูลย่อยเหล่านี้จะทำงานตามขั้นตอนวิธีที่อธิบายไว้แล้วในหัวข้อ 3.2.4.1 (รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก ฉ. รูปที่ ฉ-6) ได้แก่
 - มอดูลการเลือกส่วนสำคัญของประโยค ระหว่างการทำงานของมอดูลนี้จะมีการใช้มอดูลการรวมคำและแก้ไขหน้าที่คำด้วยไวยากรณ์ ย่อยที่ 3) ในหัวข้อที่ 4.2.4 ทำงานไปพร้อม ๆ กัน เพื่อให้ส่วนสำคัญของประโยคที่ได้อยู่ในรูปคำกริยาที่เป็นกลุ่มคำหรือค่านามที่เป็นกลุ่มคำอย่างถูกต้อง ผลลัพธ์ที่ได้จากการทำงานนี้ส่วนที่สำคัญของประโยค คือ ประธาน กริยา และกรรมของประโยคซึ่งมีความสัมพันธ์กับกริยาที่แสดงฟังก์ชันการทำงานของซอฟต์แวร์จะถูกเลือกมาแสดงผลแยกตามส่วนต่าง ๆ ของเอกสาร ประกอบด้วย ส่วนข้อความต้องการ ด้านหน้าที่ ส่วนบทนำ ส่วนข้อมูลนำเข้า ส่วนกระบวนการ และส่วนข้อมูลนำ

- ออก นอกเหนือจากนั้นยังมีการถ่วงน้ำหนักแต่ละประโยคตามรายละเอียดใน ข้อย่อยที่ 1) ของการกำหนดการถ่วงน้ำหนัก หัวข้อ 3.2.4.1 (ตัวอย่างผลลัพธ์ที่ได้จากการทำงานในมอดูลนี้ แสดงดังรูปที่ 4.27)
- มอดูลการหาแหล่งที่มาหรือแหล่งปลายทางของข้อมูล จะทำงานตามรายละเอียดในการพิจารณาคำนามที่เป็นแหล่งที่มาหรือแหล่งปลายทางของข้อมูล หัวข้อ 3.2.4.1 ผลลัพธ์ที่ได้จากการทำงานจะแสดงผลส่วนนี้ เพิ่มเติมเข้าไปใน ส่วนสำคัญของประโยคที่แสดงผลในมอดูลการเลือกส่วนสำคัญของประโยค (ตัวอย่างผลลัพธ์ที่ได้จากการทำงานในมอดูลนี้ แสดงดังรูปที่ 4.27)
 - มอดูลการถ่วงน้ำหนักของประโยค ทำงานตามรายละเอียดใน ข้อย่อยที่ 2) ของการกำหนดการถ่วงน้ำหนัก หัวข้อ 3.2.4.1 โดยการพิจารณาประโยคที่พบ ในส่วนต่าง ๆ ของเอกสารและการซ้ำกันของคำกริยาและคำนามที่พบ (ตัวอย่างผลลัพธ์ที่ได้จากการทำงานในมอดูลนี้ ดังรูปที่ 4.27)
 - มอดูลการเพิ่มเติมรายละเอียดของประโยค ทำงานตามรายละเอียดใน ข้อย่อย ที่ 3) ของการกำหนดการถ่วงน้ำหนัก หัวข้อ 3.2.4.1 (ตัวอย่างผลลัพธ์ที่ได้จาก การทำงานในมอดูลนี้ แสดงดังรูปที่ 4.27)
 - มอดูลการเลือกประโยคที่สำคัญเพื่อนำไปสร้างเป็นโครงสร้างใหม่ โดย พิจารณาจากการถ่วงน้ำหนักของแต่ละประโยค
- 2) มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมข้อมูลเพื่อให้โครงสร้างใหม่สมบูรณ์ขึ้น มอดูลนี้ทำงานตามรายละเอียดใน หัวข้อ 3.2.4.2 คือ การนำประโยคที่ถูกเลือกแล้วมาสรุ้งเป็นโครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่ โดยจะมีการเรียกมอดูลการตรวจสอบความซ้ำซ้อนกันของฟังก์ชันการทำงานในความต้องการด้านหน้าที่ เดียวกันในขั้นตอนการตรวจสอบความไม่ต้องการกัน โดยตรวจสอบฟังก์ชันการทำงานใน ความต้องการด้านหน้าที่เดียวกันว่าซ้ำซ้อนกันหรือไม่ ถ้าเกิดฟังก์ชันการทำงานที่ซ้ำซ้อน โครงสร้างใหม่ที่ได้จะไม่แสดงส่วนที่เกิดการทำงานซ้ำซ้อน แต่จะแสดงข้อผิดพลาดที่ บอกถึงการซ้ำซ้อนกันของฟังก์ชันการทำงาน นอกเหนือจากนั้นยังทำการเรียกมอดูล การตรวจสอบความซ้ำซ้อนกันของความต้องการด้านหน้าที่ ในขั้นตอนการตรวจสอบ ความไม่ต้องการกัน และจะแสดงข้อความให้ทราบว่า เกิดความซ้ำซ้อนกันของความ ต้องการด้านหน้าที่เมื่อพบความต้องการด้านหน้าที่ที่ต่างกันแต่ทำงานซ้ำซ้อนกัน หลังจากนั้นจะทำการเพิ่มเติมส่วนต่าง ๆ ของโครงสร้างใหม่ให้มีความสมบูรณ์มากขึ้น ทั้ง ลำดับการทำงาน แหล่งที่มาและแหล่งปลายทางของข้อมูล ข้อมูลนำเข้าและข้อมูลนำ ออกที่มีความเกี่ยวข้องกัน โดยมีการเรียกมอดูลตรวจสอบความไม่ต้องการกันของข้อมูล

นำเข้าและข้อมูลนำออกของแต่ละฟังก์ชันการทำงาน เพื่อให้โครงสร้างใหม่ที่ได้ตรงตามกฎความต้องกันกฎข้อที่ 2 เสมอ คือ ข้อมูลนำเข้าของฟังก์ชันการทำงานใด ๆ จะเป็นข้อมูลนำออกเสมอ ผลลัพธ์ที่ได้จะได้โครงสร้างที่ใช้อธิบายความต้องการด้านหน้าที่ที่มีความสมบูรณ์ขึ้น (ตัวอย่างผลลัพธ์ที่ได้จากการทำงานในมอดูลนี้ แสดงดังรูปที่ 4.24) รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก จ. รูปที่ จ-7

4.2.6 มอดูลขั้นตอนการตรวจสอบความไม่ต้องกัน

ขั้นตอนนี้เป็นการตรวจสอบความไม่ต้องกันของความต้องการด้านหน้าที่ จากโครงสร้างใหม่ที่อธิบายความต้องการด้านหน้าที่ (ตัวอย่างผลลัพธ์ที่ได้จากการทำงานในขั้นตอนนี้ แสดงดังรูปที่ 4.25) รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก จ. รูปที่ จ-8 ประกอบด้วยมอดูลทั้งหมด 5 มอดูล คือ มอดูลการตรวจสอบความไม่ต้องกันของข้อมูลในฐานข้อมูล มอดูลการตรวจสอบความซ้ำซ้อนกันของฟังก์ชันการทำงานในความต้องการด้านหน้าที่เดียวกัน มอดูลการตรวจสอบความซ้ำซ้อนกันของความต้องการด้านหน้าที่ มอดูลการตรวจสอบความไม่ต้องกันของข้อมูลนำเข้าและข้อมูลนำออกของฟังก์ชันการทำงาน มอดูลการตรวจสอบคำที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล มีรายละเอียดดังต่อไปนี้

- 1) มอดูลการตรวจสอบความไม่ต้องกันของข้อมูลในฐานข้อมูล ทำงานตามขั้นตอนวิธีที่ออกแบบเพื่อตรวจสอบความไม่ต้องกันในหัวข้อ 3.3.1 ซึ่งเป็นการตรวจสอบตามกฎความต้องกันกฎข้อที่ 1
- 2) มอดูลการตรวจสอบความซ้ำซ้อนกันของฟังก์ชันการทำงานในความต้องการด้านหน้าที่เดียวกัน ในการประมวลผลจริงทำงานอยู่ในมอดูลการเลือกประโยคที่สำคัญของขั้นตอนการสร้างรูปแบบโครงสร้างใหม่ มีการอธิบายไปแล้วข้างต้น ซึ่งเป็นการตรวจสอบตามกฎความต้องกันกฎข้อที่ 5
- 3) มอดูลการตรวจสอบความซ้ำซ้อนกันของความต้องการด้านหน้าที่ ในการประมวลผลจริงทำงานอยู่ในมอดูลการเลือกประโยคที่สำคัญของขั้นตอนการสร้างรูปแบบโครงสร้างใหม่ มีการอธิบายไปแล้วข้างต้น ซึ่งเป็นการตรวจสอบตามกฎความต้องกันกฎข้อที่ 6
- 4) มอดูลการตรวจสอบความไม่ต้องกันของข้อมูลนำเข้าและข้อมูลนำออกของฟังก์ชันการทำงาน โดยทำงานตามขั้นตอนวิธีที่ออกแบบเพื่อตรวจสอบความไม่ต้องกันในหัวข้อ 3.3.2 ซึ่งเป็นการตรวจสอบตามกฎความต้องกันกฎข้อที่ 2 – กฎข้อที่ 4 (ในการประมวลผลจริงการทำงานในการตรวจสอบความไม่ต้องกันตามกฎข้อที่ 2 ทำงานอยู่ในมอดูลการเพิ่มเติมข้อมูลเพื่อให้โครงสร้างใหม่สมบูรณ์ขึ้น)

- 5) มอดูลการตรวจสอบค่าที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล ซึ่งเป็นการตรวจสอบตามกฎความต้องกันกฎข้อที่ 7 โดยทำงานตามขั้นตอนวิธีที่ออกแบบเพื่อตรวจสอบความไม่ต้องกันในหัวข้อ 3.3.3 มีการพิจารณาค่าหรือข้อความที่ใช้อธิบายถึงกลุ่มข้อมูลในฐานข้อมูลที่ปรากฏในแต่ละความต้องการด้านหน้าที่ว่าข้อมูลย่อยที่ประกอบอยู่ในข้อมูลนั้น เป็นข้อมูลย่อยเดียวกันหรือไม่ และมีการแสดงผลข้อมูลย่อยในกลุ่มข้อมูลของแต่ละความต้องการด้านหน้าที่ รายละเอียดชุดคำสั่งแสดงใน ภาคผนวก จ. รูปที่ จ-9

4.2.7 มอดูลอื่น ๆ

ในส่วนนี้เป็นการทำงานที่ไม่เกี่ยวข้องกับขั้นตอนวิธีการตรวจสอบความไม่ต้องกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ แต่เป็นมอดูลที่ต้องใช้งานร่วมกับมอดูลของขั้นตอนวิธีในการตรวจสอบความไม่ต้องกัน ตัวอย่างเช่น มอดูลการดึงข้อมูลจากฐานข้อมูลขึ้นมาแสดงผล มอดูลการตรวจสอบค่า ความหมายค่าจากฐานข้อมูล มอดูลแสดงข้อความที่แสดงถึงข้อผิดพลาด เป็นต้น

4.3 การใช้งานเบื้องต้น

ตัวอย่างของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ก่อนที่จะนำมาตรวจสอบความไม่ต้องกัน โดยใช้เครื่องมือซอฟต์แวร์ ต้องมีกระบวนการเตรียมเอกสารด้วยมือ ให้ตรงกับมาตรฐานของเอกสารที่งานวิจัยนี้เลือกใช้ (มาตรฐานเอกสารมีรายละเอียดในภาคผนวก) และในเครื่องมือซอฟต์แวร์มีการสร้างมอดูลการทำงาน สำหรับตรวจสอบรูปแบบเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่เป็นข้อมูลนำเข้า ว่าตรงกับมาตรฐานของเอกสารหรือไม่ ถ้าไม่ตรงจะไม่สามารถทำการประมวลผลขั้นต่อไปได้

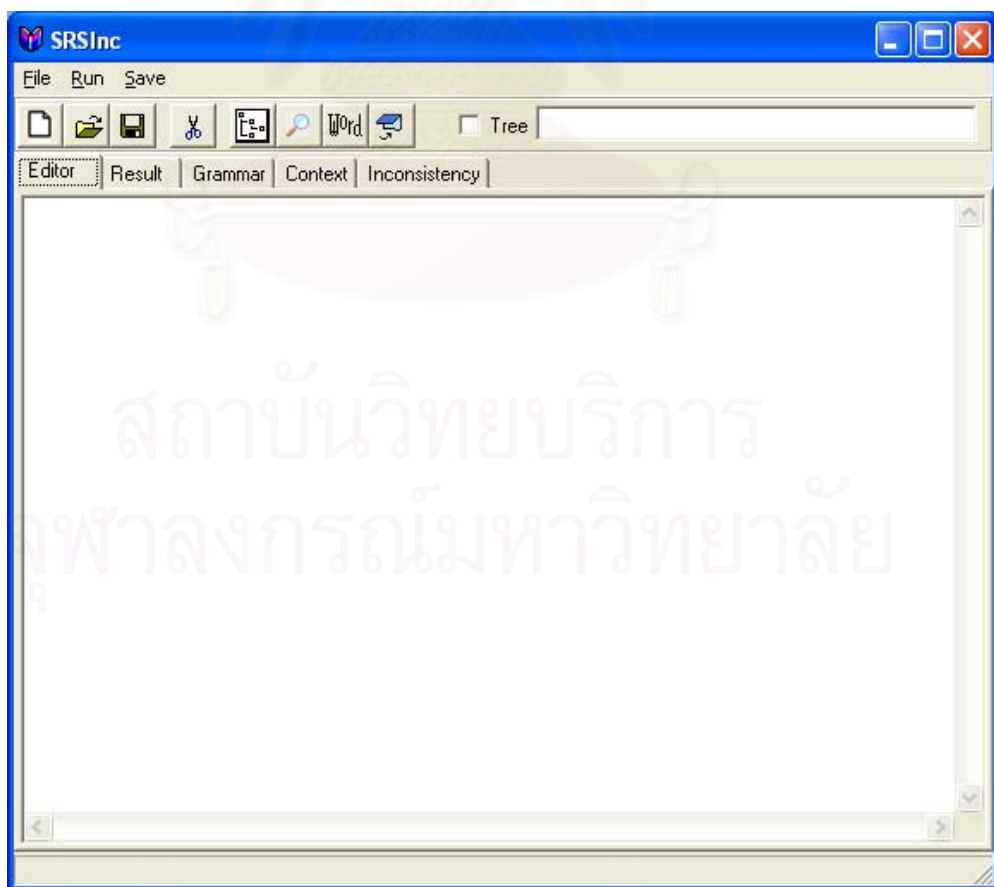
การเริ่มทำงานจะเริ่มที่ รูปที่ 4.19 หน้าจอหลักการทำงาน โดยส่วนแถบเอดิเตอร์รับข้อมูลนำเข้าโปรแกรม ซึ่งคือเอกสารข้อกำหนดความต้องการซอฟต์แวร์ ส่วนความต้องการด้านหน้าที่ต้องการตรวจสอบหาความไม่ต้องกัน เลือกแถบเมนูหรือแถบเครื่องมือที่ต้องการทำงาน เครื่องมือซอฟต์แวร์จะประมวลผล และตรวจสอบความไม่ต้องกันได้โดยอัตโนมัติ (รายละเอียดในภาคผนวก จ.)

บทที่ 5

การทดสอบและประเมินผลเครื่องมือ

งานวิจัยนี้มีการทดสอบเครื่องมือซอฟต์แวร์ โดยการนำเครื่องมือซอฟต์แวร์มาใช้ตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ซึ่งตัวอย่างเอกสารนั้นมีทั้งเอกสารที่ถูกต้องไม่พบความไม่ตรงกัน และเอกสารที่มีการแก้ไขให้เกิดความไม่ตรงกันขึ้น โดยในการทดสอบจะใช้เครื่องมือซอฟต์แวร์ตรวจสอบความไม่ตรงกันจากตัวอย่างเอกสารที่ถูกต้องไม่พบความไม่ตรงกัน และนำตัวอย่างเอกสารนั้นมาทำการแก้ไขให้เกิดความไม่ตรงกันด้านต่าง ๆ ตามที่ระบุไว้ในงานวิจัย สำหรับทดสอบเครื่องมือซอฟต์แวร์ว่าสามารถตรวจสอบพบความไม่ตรงกันเหล่านั้นหรือไม่

เริ่มทำการรัน (Run) เครื่องมือซอฟต์แวร์ โดยเรียกโปรแกรมชื่อว่า “ChkSRS.exe” จะได้จอภาพแรกเริ่มต้นการทำงาน แสดงดังรูปที่ 5.1 หลังจากนั้นเลือกเมนูการทำงานที่ต้องการในแถบเมนูหรือเลือกเครื่องมือที่ต้องการในแถบเครื่องมือ เพื่อให้เครื่องมือซอฟต์แวร์ทำงานตามที่ต้องการ ซึ่งจะอธิบายในส่วนถัดไป



รูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน

5.1 การตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1


การตรวจสอบความไม่ต้งกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 จะทำการตรวจสอบทั้งตัวอย่างเอกสารที่ 1 ที่ยังไม่มีกรแก้ไขให้เกิดความไม่ต้งกัน และตัวอย่างเอกสารที่ 1 ที่ทำการแก้ไขให้เกิดความไม่ต้งกัน ซึ่งเอกสารที่ทำการแก้ไขมาจากเอกสารเดิม จะใช้หมายเลขเอกสารอ้างอิงตามตัวอย่างเอกสารต้นฉบับที่นำมาแก้ไข

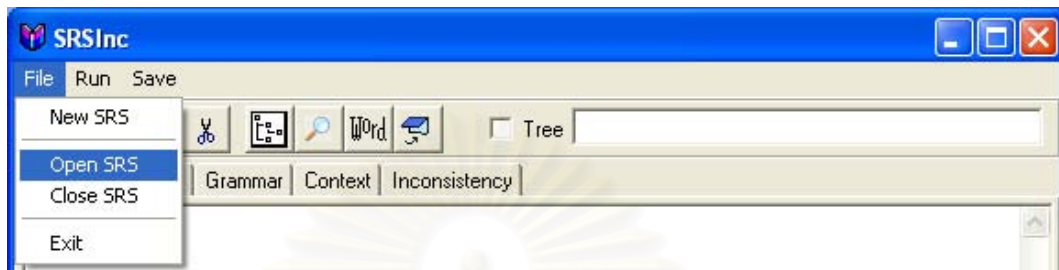
5.1.1 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ที่ไม่มีกรแก้ไขให้เกิดความไม่ต้งกัน

รายละเอียดของตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 แสดงในตารางที่ 5.1

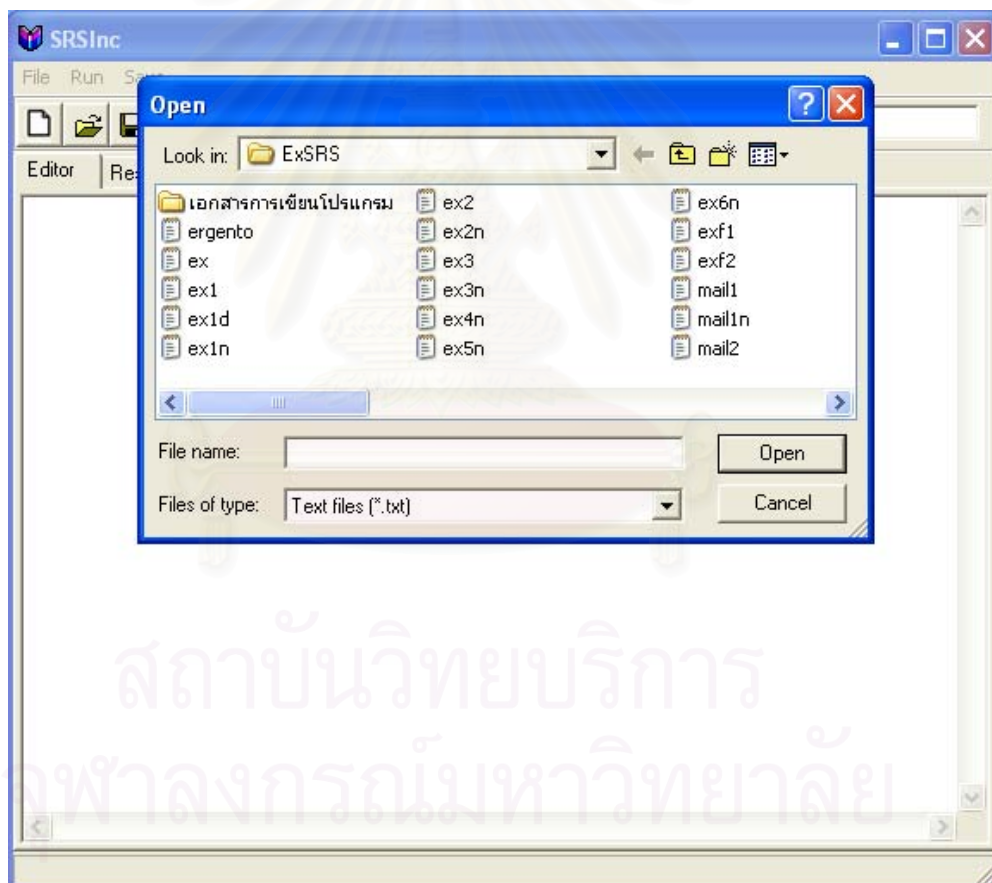
ตารางที่ 5.1 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1

ความต้องการด้านหน้าที่ที่ 1 ตรวจสอบข้อมูลต่างๆ ของลูกค้า	
บทนำ	-
ข้อมูลนำเข้า	หมายเลขบัตรเครดิตของลูกค้า
กระบวนการ	เมื่อรับ input เข้ามาแล้ว จะทำการค้นหาข้อมูลลูกค้าในฐานข้อมูล ถ้าเจอจะแสดงชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้า
ข้อมูลนำออก	ชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้าส่งไปยังลูกค้า
ความต้องการหน้าที่ที่ 2 รับข้อมูลต่าง ๆ ของลูกค้า	
บทนำ	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้า ที่อยู่ลูกค้าจากผู้ใช้
กระบวนการ	รับข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	รหัสลูกค้า ชื่อลูกค้า ที่อยู่ลูกค้าไปเก็บในฐานข้อมูล
ความต้องการหน้าที่ที่ 3 การพิมพ์ใบส่งสินค้า	
บทนำ	หน้าที่ในส่วนนี้ เป็นการดึงข้อมูลลูกค้าจากฐานข้อมูล แล้วสั่งพิมพ์ใบส่งสินค้าซึ่งประกอบด้วย ชื่อและที่อยู่ของลูกค้าออกทางเครื่องพิมพ์
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับชื่อลูกค้า เพื่อดึงข้อมูลที่อยู่ของลูกค้าจากฐานข้อมูล แล้วพิมพ์ใบส่งสินค้า
ข้อมูลนำออก	ชื่อและที่อยู่ของลูกค้าไปยังผู้ใช้

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกแถบเมนู Open SRS (แสดงดังรูปที่ 5.2) จะปรากฏหน้าจอ สำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ต้องการนำมาตรวจสอบความไม่ตรงกัน เพื่อเป็นข้อมูลนำเข้าของเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ 5.3

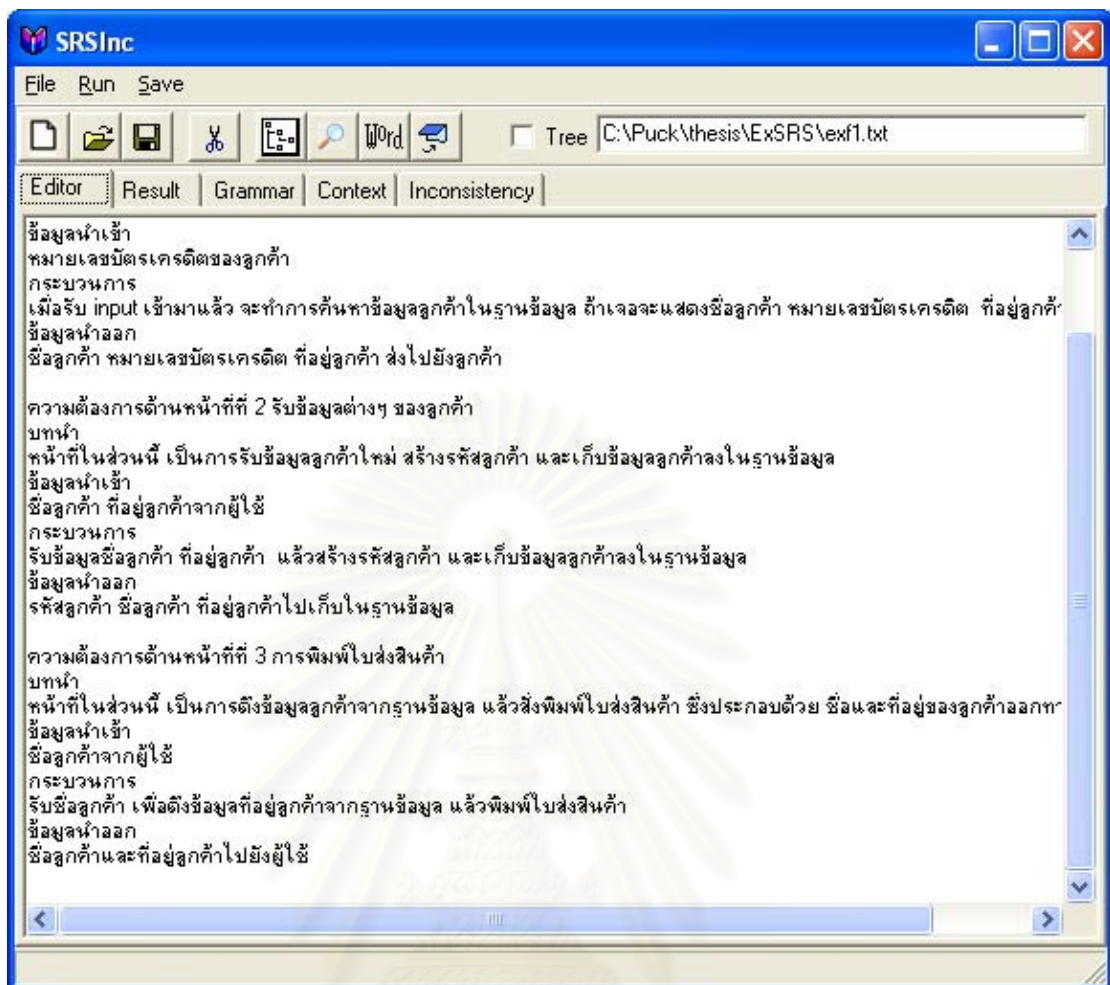


รูปที่ 5.2 แถบเมนู Open SRS เปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่




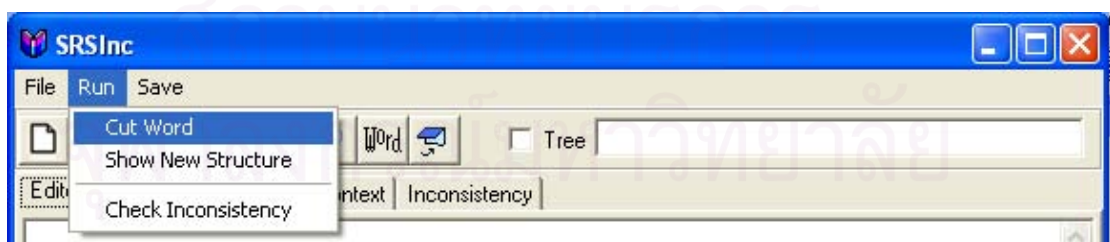
รูปที่ 5.3 หน้าจอสำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ต้องการ

จากรูปที่ 5.3 เลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ผลลัพธ์แสดงดังรูปที่ 5.4

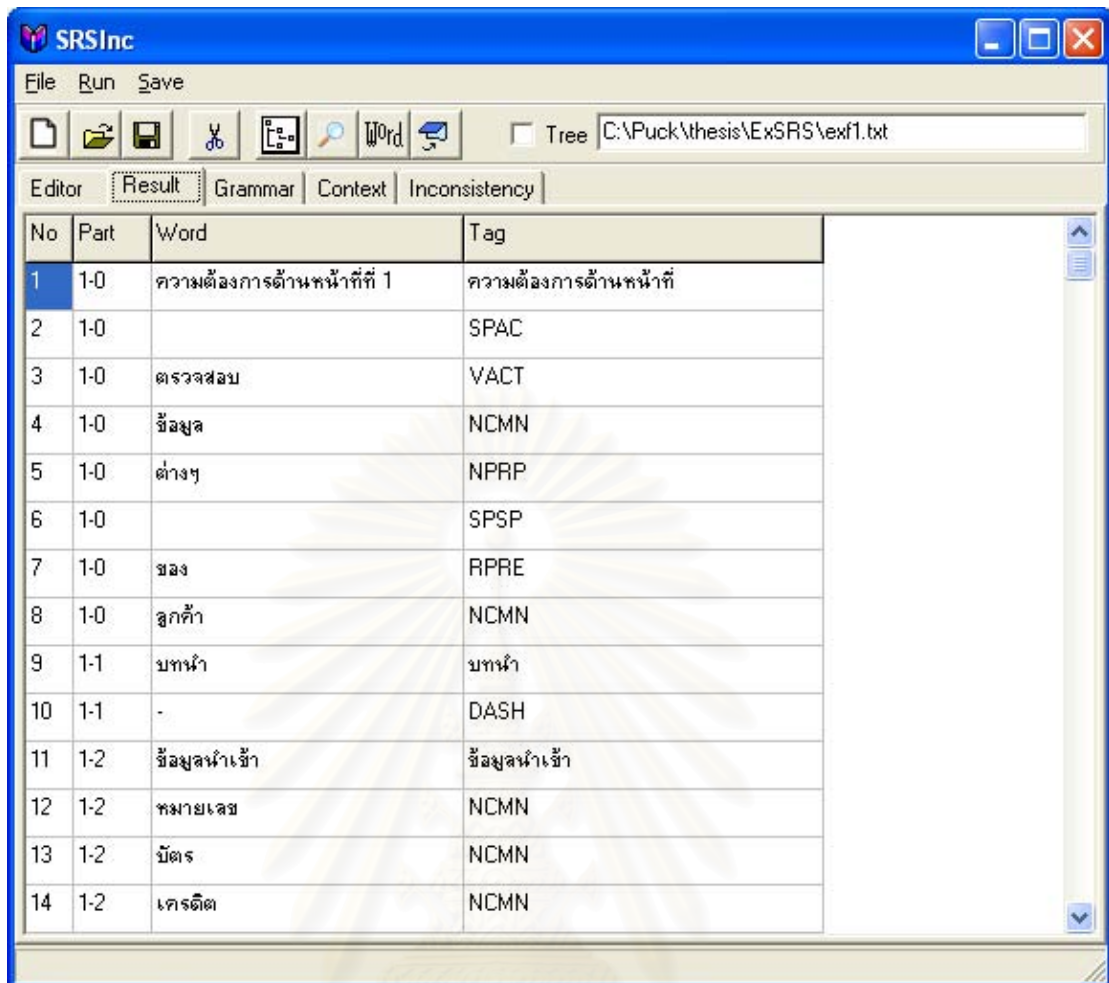


รูปที่ 5.4 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์
ด้านหน้าที่ตัวอย่างที่ 1

เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู (แสดงดังรูปที่ 5.5) เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร ผลลัพธ์แสดงดังรูปที่ 5.6




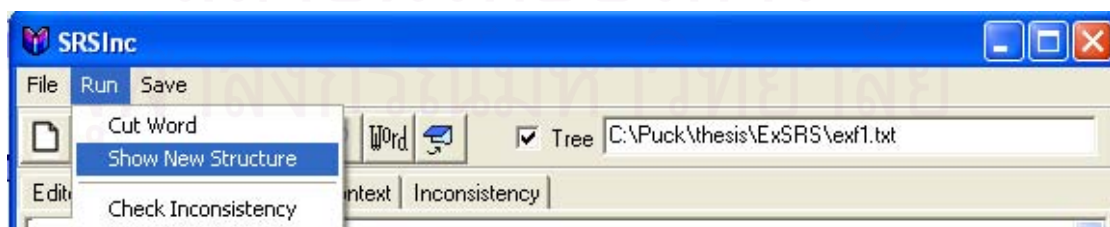
รูปที่ 5.5 แถบเมนู Cut Word



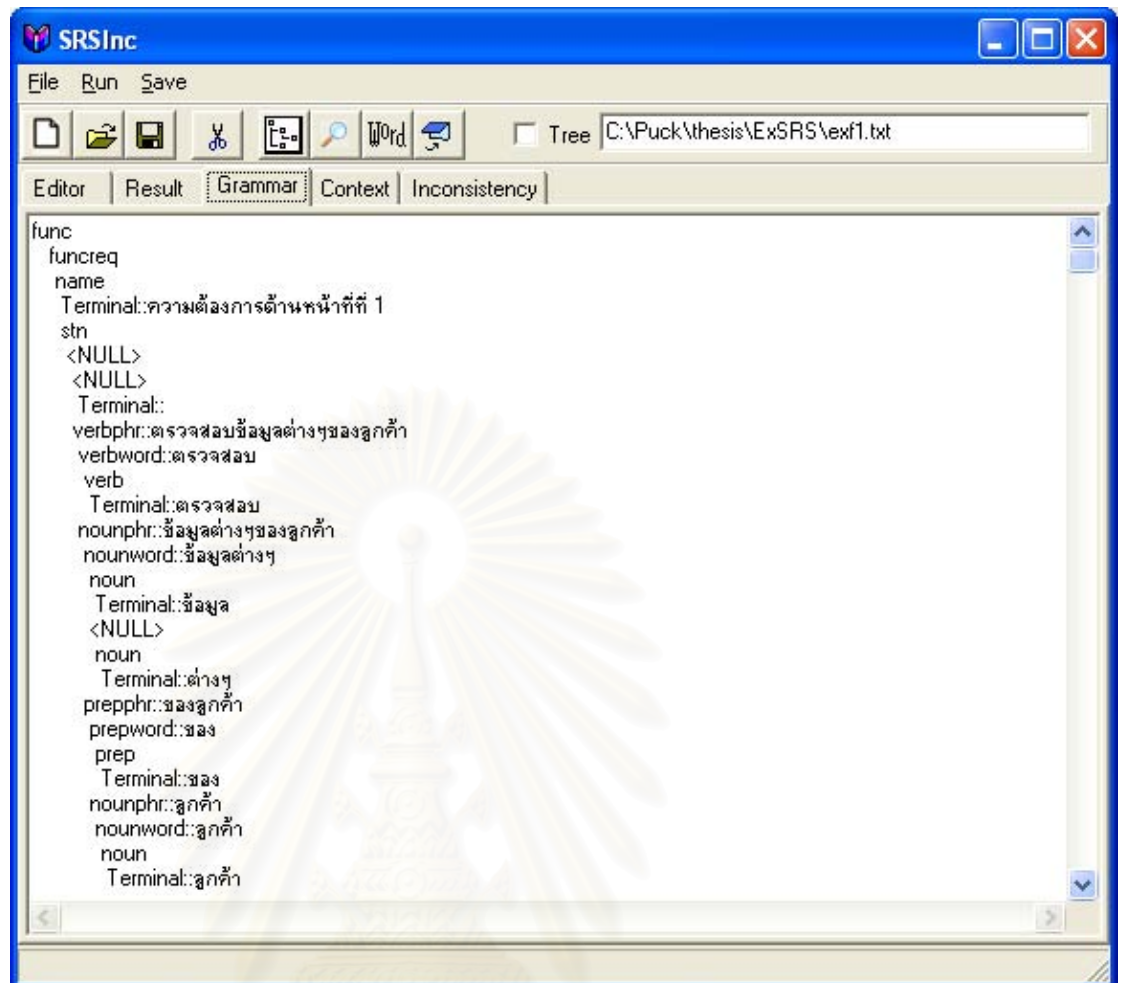
No	Part	Word	Tag
1	1-0	ความต้องการด้านหน้าที่ที่ 1	ความต้องการด้านหน้าที่
2	1-0		SPAC
3	1-0	ตรวจสอบ	VACT
4	1-0	ข้อมูล	NCMN
5	1-0	ต่างๆ	NPRP
6	1-0		SPSP
7	1-0	ลง	RPRE
8	1-0	ลูกค้า	NCMN
9	1-1	บทนำ	บทนำ
10	1-1	-	DASH
11	1-2	ข้อมูลนำเข้า	ข้อมูลนำเข้า
12	1-2	หมายเลข	NCMN
13	1-2	บัตร	NCMN
14	1-2	เครดิต	NCMN

รูปที่ 5.6 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ

เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู (แสดงดังรูปที่ 5.7) เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ โดยการดึงส่วนสำคัญของประโยคที่อธิบายรายละเอียดความต้องการด้านหน้าที่ ผลลัพธ์การวิเคราะห์ประโยคด้วยไวยากรณ์แสดงดังรูปที่ 5.8



รูปที่ 5.7 แถบเมนู Show New Structure

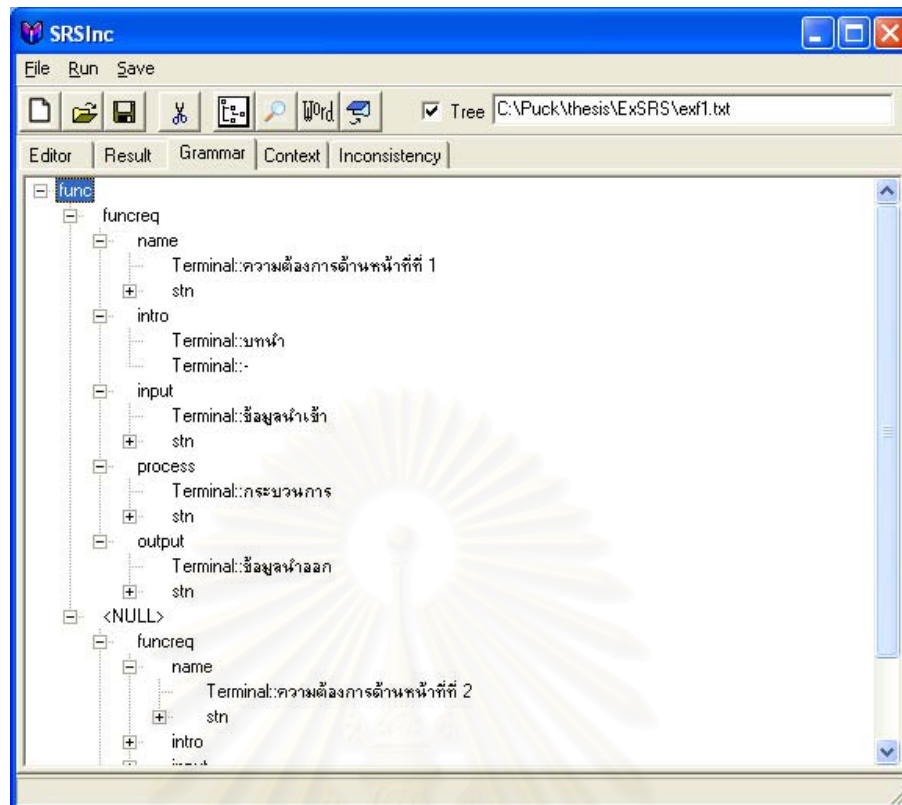


รูปที่ 5.8 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์

หากต้องการให้ผลลัพธ์การวิเคราะห์ประโยคด้วยไวยากรณ์ แสดงลักษณะโครงสร้างรูปต้น
ไม้ ต้องทำการคลิกเลือก Tree ในแถบเครื่องมือ ดังแสดงในรูปที่ 5.9 ก่อนทำการเลือกเครื่องมือ
ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู ผลลัพธ์การวิเคราะห์ประโยคด้วย
ไวยากรณ์ในรูปต้นไม้โครงสร้างวลี แสดงดังรูปที่ 5.10



รูปที่ 5.9 ออบชั่น Tree



รูปที่ 5.10 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ โครงสร้างรูปต้นไม้


No	Usecas	Source	Input	Act	Output	Dest.
1	หน้าที่ที่ 1	ลูกค้า	หมายเลข	รับหมายเลข	หมายเลขบัตร	ฟังก์ชัน 1
2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	หมายเลขบัตร	ค้นหาข้อมูล	หมายเลขบัตร	ฐานข้อมูล
3	หน้าที่ที่ 1	ฐานข้อมูล	ข้อมูลลูกค้า	แสดงผลข้อมูล	ข้อมูลลูกค้า	หมายเลขลูกค้า
1	หน้าที่ที่ 2	ผู้ใช้	ข้อมูลชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ข้อมูลชื่อลูกค้า	ฟังก์ชัน 2
2	หน้าที่ที่ 2	ฟังก์ชัน 2.1	ข้อมูลชื่อลูกค้า	สร้างรหัสลูกค้า	ข้อมูลชื่อลูกค้า	ฟังก์ชัน 2
3	หน้าที่ที่ 2	ฟังก์ชัน 2.2	ข้อมูลชื่อลูกค้า	เก็บข้อมูลลูกค้า	ข้อมูลชื่อลูกค้า	ฐานข้อมูล
1	หน้าที่ที่ 3	ผู้ใช้	ข้อมูลชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ข้อมูลชื่อลูกค้า	ฟังก์ชัน 3
2	หน้าที่ที่ 3	ฟังก์ชัน 3.1	ข้อมูลชื่อลูกค้า	ดึงข้อมูลที่อยู่	ข้อมูลชื่อลูกค้า	ฐานข้อมูล
3	หน้าที่ที่ 3	ฐานข้อมูล	ข้อมูลชื่อลูกค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

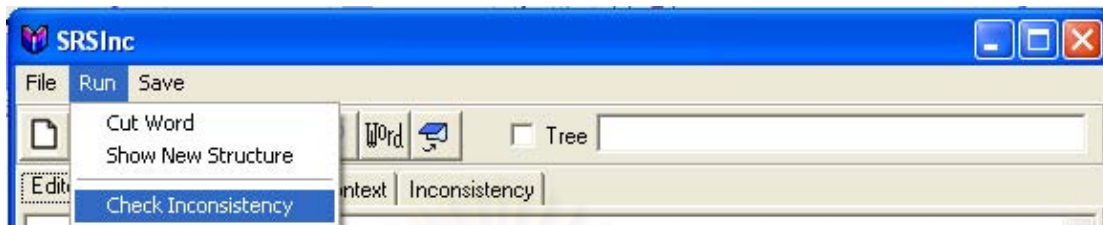
รูปที่ 5.11 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

หลังจากการวิเคราะห์ประโยชน์ด้วยไวยากรณ์ จะทำการดึงส่วนสำคัญของประโยคที่เกี่ยวข้องกับคำหรือวลีสำคัญในการอธิบายความต้องการด้านหน้าที่ เพื่อสร้างโครงสร้างใหม่ที่ใช้อธิบายแทนความต้องการด้านหน้าที่ที่เดิมเป็นข้อความ ผลลัพธ์การทำงานของเครื่องมือซอฟต์แวร์แสดงดังรูปที่ 5.11 (โครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 1 แสดงในตารางที่ 5.2)

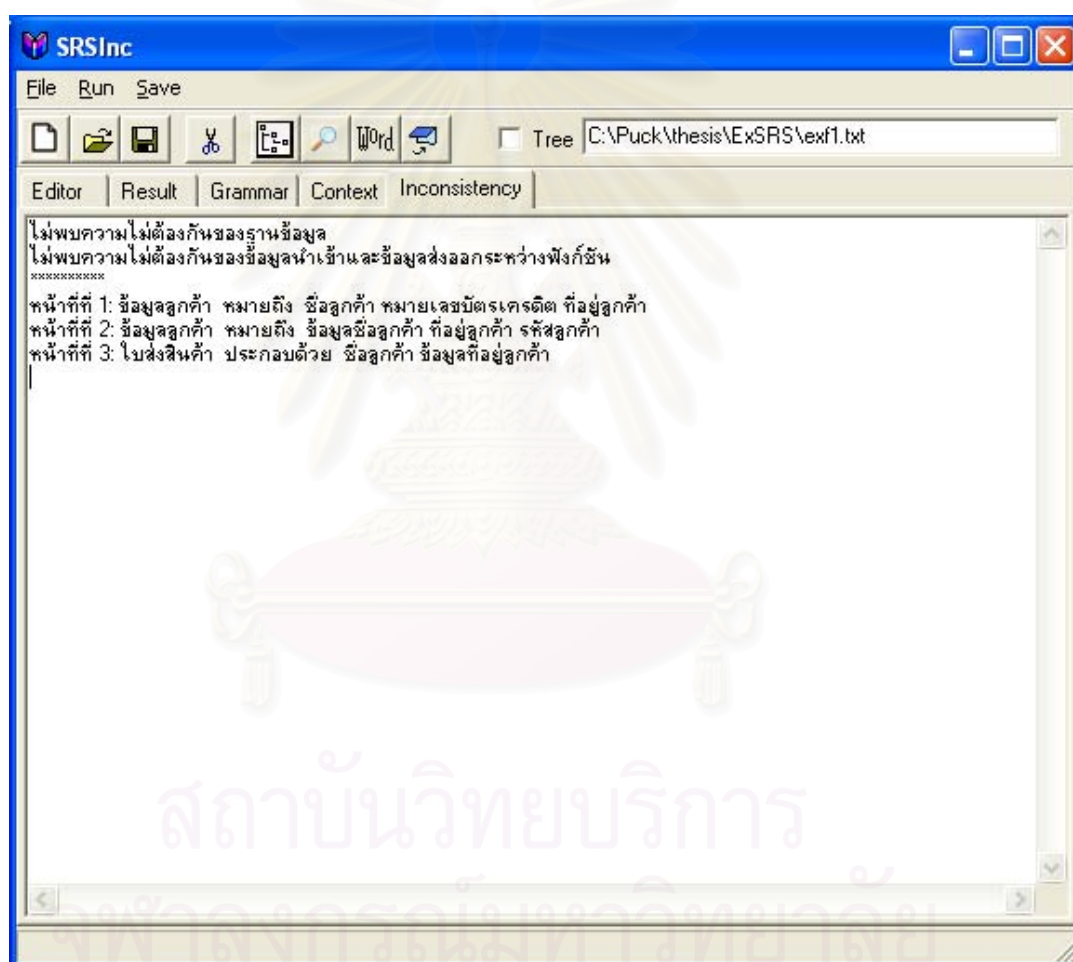
ตารางที่ 5.2 โครงสร้างใหม่ของความต้องการด้านหน้าที่ตัวอย่างที่ 1

No.	Use case	Source	Input	Action	Output	Destination
1.1	หน้าที่ที่ 1	ลูกค้า	หมายเลขบัตร เครดิต	รับหมายเลขบัตร เครดิต	หมายเลขบัตร เครดิต	ฟังก์ชัน 1.2
1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	หมายเลขบัตร เครดิต	ค้นหาข้อมูลลูกค้า	หมายเลขบัตร เครดิต	ฐานข้อมูล
1.3	หน้าที่ที่ 1	ฐานข้อมูล	ชื่อลูกค้า หมายเลข บัตรเครดิต ที่อยู่ลูกค้า	แสดงชื่อลูกค้า หมายเลขบัตร เครดิต ที่อยู่ลูกค้า	ชื่อลูกค้า หมายเลข บัตรเครดิต ที่อยู่ลูกค้า	ลูกค้า
2.1	หน้าที่ที่ 2	ผู้ใช้	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า	รับข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า	ฟังก์ชัน 2.2
2.2	หน้าที่ที่ 2	ฟังก์ชัน 2.1	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า	สร้างรหัสลูกค้า	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า รหัส ลูกค้า	ฟังก์ชัน 2.3
2.3	หน้าที่ที่ 3	ฟังก์ชัน 2.2	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า รหัสลูกค้า	เก็บข้อมูลลูกค้า	ข้อมูลชื่อลูกค้า ที่อยู่ลูกค้า รหัสลูกค้า	ฐานข้อมูล
3.1	หน้าที่ที่ 3	ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 3.2
3.2	หน้าที่ที่ 3	ฟังก์ชัน 3.1	ชื่อลูกค้า	ดึงข้อมูลที่อยู่ลูก ค้า	ชื่อลูกค้า	ฐานข้อมูล
3.3	หน้าที่ที่ 3	ฐานข้อมูล	ชื่อลูกค้า ข้อมูลที่อยู่ลูก ค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

จากรูปที่ 5.11 เลือกรูปเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Check Inconsistency ในแถบเมนู (แสดงดังรูปที่ 5.12) เพื่อให้เครื่องมือซอฟต์แวร์ทำการตรวจสอบความไม่ตรงกันของเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์ตัวอย่างที่ 1 ผลลัพธ์แสดงดังรูปที่ 5.13



รูปที่ 5.12 แถบเมนู Check Inconsistency



รูปที่ 5.13 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน

รายละเอียดผลการตรวจสอบความไม่ตรงกัน ได้แก่

- 1) ไม่พบความไม่ตรงกันของข้อมูลในฐานข้อมูล แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 1
- 2) ไม่พบความไม่ตรงกันของข้อมูลนำเข้าและข้อมูลส่งออกระหว่างฟังก์ชัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 2 – กฎข้อที่ 4

- 3) ไม่มีการแสดงข้อความผิดพลาดว่าฟังก์ชันการทำงานซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 5
- 4) ไม่มีการแสดงข้อความผิดพลาดว่าความต้องการด้านหน้าที่ซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 6
- 5) สำหรับกฎความต้องกันกฎข้อที่ 7 เครื่องมือซอฟต์แวร์จะแสดงรายละเอียดของข้อมูล ที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล ว่าประกอบด้วยข้อมูลย่อยใดบ้าง จากตัวอย่าง เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1 เช่น คำว่า “ข้อมูลลูกค้า”

5.1.2 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1 (เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ที่มีการแก้ไขให้เกิดความไม่ต้องกันขึ้น)


เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1 นี้ สร้างจากการแก้ไข ความต้องการด้านหน้าที่ที่ 2 ของเอกสารตัวอย่างที่ 1 ในส่วนของข้อมูลนำเข้า กระบวนการ และข้อมูลนำออก รายละเอียดของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1 แสดงในตารางที่ 5.3

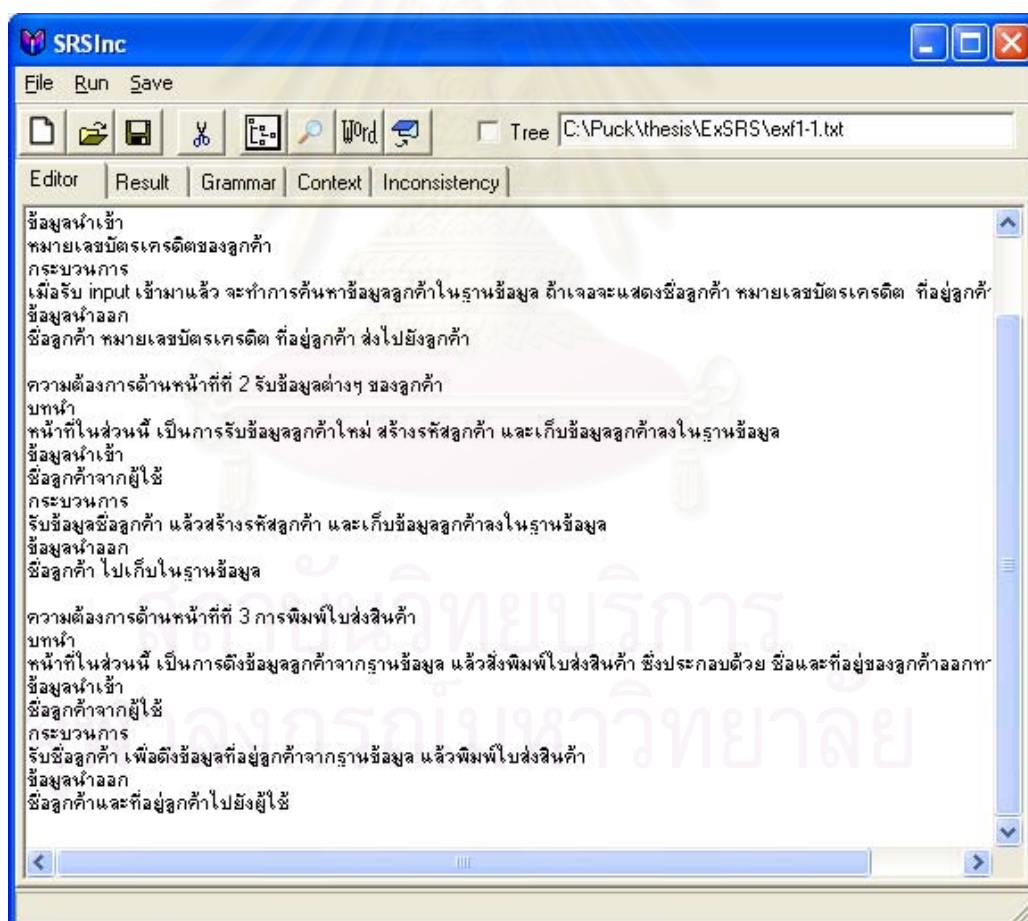
ตารางที่ 5.3 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1

ความต้องการด้านหน้าที่ที่ 1 ตรวจสอบข้อมูลต่างๆ ของลูกค้า	
บทนำ	-
ข้อมูลนำเข้า	หมายเลขบัตรเครดิตของลูกค้า
กระบวนการ	เมื่อรับ input เข้ามาแล้ว จะทำการค้นหาข้อมูลลูกค้าในฐานข้อมูล ถ้าเจอจะแสดงชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้า
ข้อมูลนำออก	ชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้าส่งไปยังลูกค้า
ความต้องการหน้าที่ที่ 2 รับข้อมูลต่าง ๆ ของลูกค้า	
บทนำ	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับข้อมูลชื่อลูกค้า แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	ชื่อลูกค้า ไปเก็บในฐานข้อมูล



ตารางที่ 5.3 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1 (ต่อ)

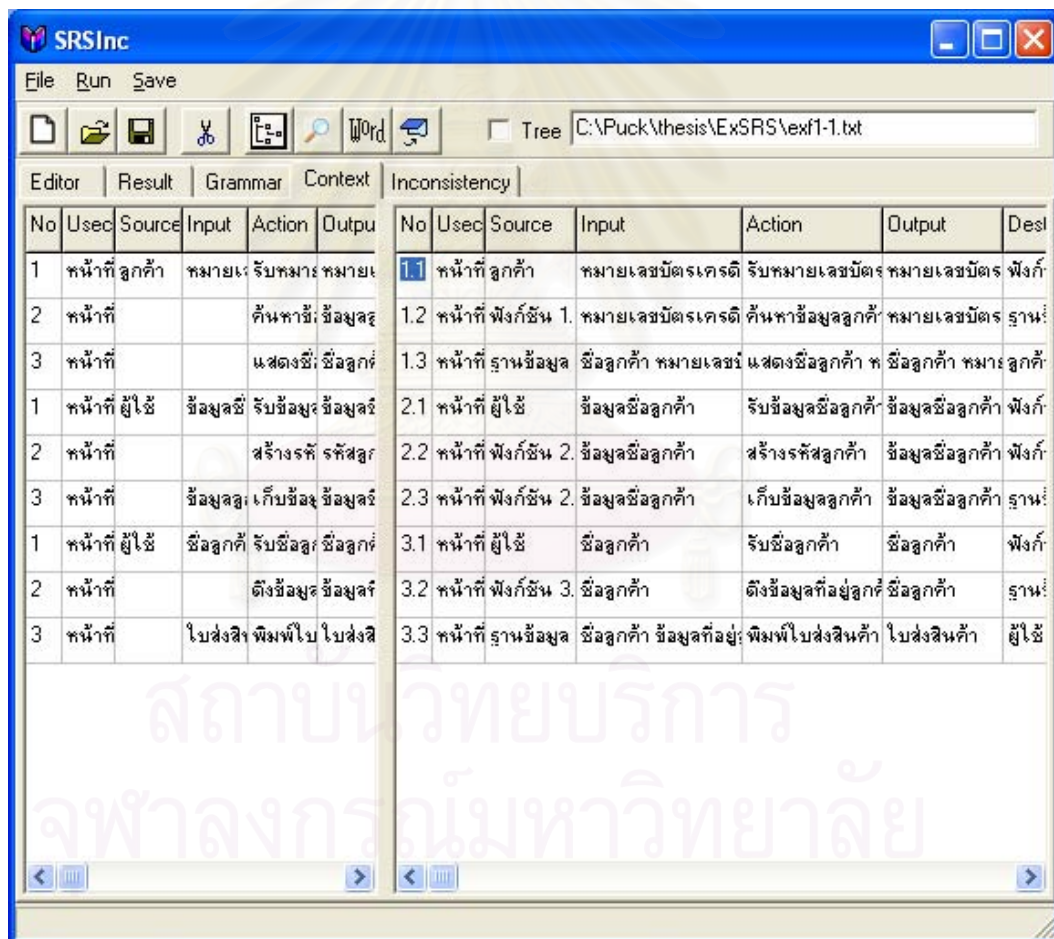
ความต้องการหน้าที่ที่ 3 การพิมพ์ใบส่งสินค้า	
บทบาท	หน้าที่ในส่วนนี้ เป็นการดึงข้อมูลลูกค้าจากฐานข้อมูล แล้วสั่งพิมพ์ใบส่งสินค้า ซึ่งประกอบด้วย ชื่อและที่อยู่ของลูกค้าออกทางเครื่องพิมพ์
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับชื่อลูกค้า เพื่อดึงข้อมูลที่อยู่ของลูกค้าจากฐานข้อมูล แล้วพิมพ์ใบส่งสินค้า
ข้อมูลนำออก	ชื่อและที่อยู่ของลูกค้าไปยังผู้ใช้

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน หลังจากเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกแถบเมนู Open SRS เพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-1 ผลลัพธ์การทำงานแสดงดังรูปที่ 5.14



รูปที่ 5.14 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-1

เนื่องจากการทำงานในลำดับถัดไปของเครื่องมือซอฟต์แวร์ ซึ่งประกอบด้วย การเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร และการเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ หน้าจอผลลัพธ์การทำงานที่ได้จากการตัดคำและกำกับหน้าที่คำ รวมถึงส่วนประกอบของประโยคที่ได้จากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะคล้ายคลึงกับ รายละเอียดที่อธิบายไปแล้วในการตรวจสอบความไม่ตรงกันของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ดังนั้นในส่วนนี้จะละรายละเอียดไว้ โดยจะแสดงเพียงผลลัพธ์ของโครงสร้างใหม่ที่ได้ ดังรูปที่ 5.15 และผลการตรวจสอบความไม่ตรงกัน ดังรูปที่ 5.16 เท่านั้น



Editor						Inconsistency						
No	Used	Source	Input	Action	Output	No	Used	Source	Input	Action	Output	Desl
1	หน้าที่ ลูกค้า	หมายเลข	รับหมายเลข	หมายเลข	หมายเลข	1.1	หน้าที่ ลูกค้า	หมายเลขบัตรเครดิต	รับหมายเลขบัตรเครดิต	หมายเลขบัตร	หมายเลขบัตร	ฟังก์ชัน
2	หน้าที่		ค้นหาชื่อ	ชื่อลูกค้า	ชื่อลูกค้า	1.2	หน้าที่ ฟังก์ชัน 1.	หมายเลขบัตรเครดิต	ค้นหาชื่อลูกค้า	ค้นหาชื่อลูกค้า	หมายเลขบัตร	ฐานข้อมูล
3	หน้าที่		แสดงชื่อ	ชื่อลูกค้า	ชื่อลูกค้า	1.3	หน้าที่ ฐานข้อมูล	ชื่อลูกค้า หมายเลข	แสดงชื่อลูกค้า	ค้นหาชื่อลูกค้า	หมายเลขลูกค้า	หมายเลขลูกค้า
1	หน้าที่ ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ชื่อลูกค้า	2.1	หน้าที่ ผู้ใช้	ชื่อลูกค้าชื่อลูกค้า	รับชื่อลูกค้าชื่อลูกค้า	รับชื่อลูกค้าชื่อลูกค้า	ชื่อลูกค้าชื่อลูกค้า	ฟังก์ชัน
2	หน้าที่		สร้างรหัส	รหัสลูกค้า	รหัสลูกค้า	2.2	หน้าที่ ฟังก์ชัน 2.	ชื่อลูกค้าชื่อลูกค้า	สร้างรหัสลูกค้า	สร้างรหัสลูกค้า	ชื่อลูกค้าชื่อลูกค้า	ฟังก์ชัน
3	หน้าที่	ชื่อลูกค้า	เก็บชื่อลูกค้า	ชื่อลูกค้า	ชื่อลูกค้า	2.3	หน้าที่ ฟังก์ชัน 2.	ชื่อลูกค้าชื่อลูกค้า	เก็บชื่อลูกค้า	เก็บชื่อลูกค้า	ชื่อลูกค้าชื่อลูกค้า	ฐานข้อมูล
1	หน้าที่ ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ชื่อลูกค้า	3.1	หน้าที่ ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน
2	หน้าที่		ดึงชื่อลูกค้า	ชื่อลูกค้า	ชื่อลูกค้า	3.2	หน้าที่ ฟังก์ชัน 3.	ชื่อลูกค้า	ดึงชื่อลูกค้าที่ชื่อลูกค้า	ดึงชื่อลูกค้า	ชื่อลูกค้า	ฐานข้อมูล
3	หน้าที่	ใบส่งสินค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ใบส่งสินค้า	3.3	หน้าที่ ฐานข้อมูล	ชื่อลูกค้า ชื่อลูกค้าที่อยู่	พิมพ์ใบส่งสินค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้


รูปที่ 5.15 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

รายละเอียดโครงสร้างใหม่ของการความต้องการด้านหน้าที่ ตัวอย่างที่ 1-1 แสดงในตารางที่

5.4

ตารางที่ 5.4 โครงสร้างใหม่ของการความต้องการด้านหน้าที่ตัวอย่างที่ 1-1

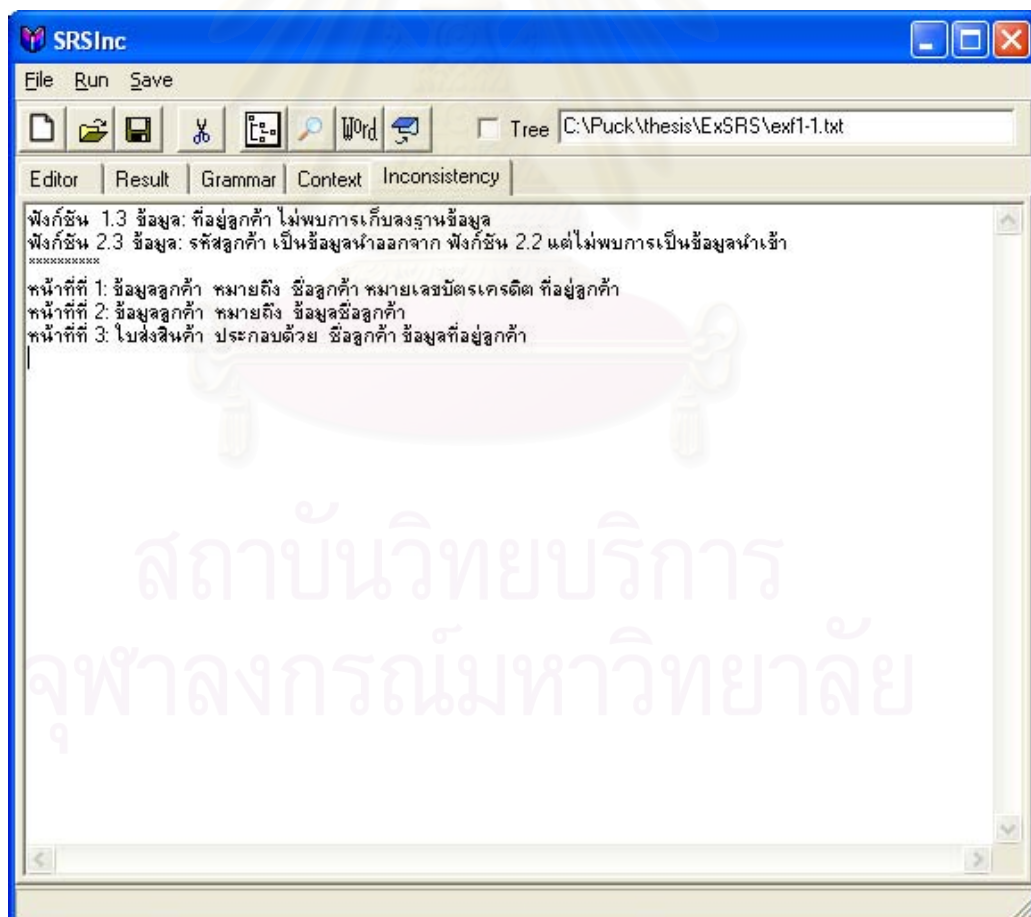
No.	Use case	Source	Input	Action	Output	Destination
1.1	หน้าที่ที่ 1	ลูกค้า	หมายเลขบัตร เครดิต	รับหมายเลขบัตร เครดิต	หมายเลขบัตร เครดิต	ฟังก์ชัน 1.2
1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	หมายเลขบัตร เครดิต	ค้นหาข้อมูลลูกค้า	หมายเลขบัตร เครดิต	ฐานข้อมูล
1.3	หน้าที่ที่ 1	ฐานข้อมูล	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	แสดงชื่อลูกค้า หมายเลขบัตร เครดิต ที่อยู่ลูกค้า	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	ลูกค้า
2.1	หน้าที่ที่ 2	ผู้ใช้	ข้อมูลชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ข้อมูลชื่อลูกค้า	ฟังก์ชัน 2.2
2.2	หน้าที่ที่ 2	ฟังก์ชัน 2.1	ข้อมูลชื่อลูกค้า	สร้างรหัสลูกค้า	ข้อมูลชื่อลูกค้า รหัสลูกค้า	ฟังก์ชัน 2.3
2.3	หน้าที่ที่ 3	ฟังก์ชัน 2.2	ข้อมูลชื่อลูกค้า	เก็บข้อมูลลูกค้า	ข้อมูลชื่อลูกค้า	ฐานข้อมูล
3.1	หน้าที่ที่ 3	ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 3.2
3.2	หน้าที่ที่ 3	ฟังก์ชัน 3.1	ชื่อลูกค้า	ดึงข้อมูลที่อยู่ลูกค้า	ชื่อลูกค้า	ฐานข้อมูล
3.3	หน้าที่ที่ 3	ฐานข้อมูล	ชื่อลูกค้า ข้อมูลที่อยู่ลูกค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

จากรูปที่ 5.15 เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Check Inconsistency ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตรวจสอบความไม่ตรงกันของเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์ตัวอย่างที่ 1-1 ผลลัพธ์แสดงดังรูปที่ 5.16

รายละเอียดผลการตรวจสอบความไม่ตรงกัน ได้แก่

- 1) พบความไม่ตรงกันของข้อมูลในฐานข้อมูล แสดงว่าไม่ตรงตามกฎความต้องกันกฎข้อที่ 1 เนื่องจากไม่มีฟังก์ชันการทำงานใด เก็บข้อมูลที่อยู่ลูกค้าลงฐานข้อมูล แต่มีการเรียกข้อมูลที่อยู่จากฐานข้อมูลมาแสดงผล

- 2) พบความไม่ตรงกันของข้อมูลนำออกจากฟังก์ชันการทำงานก่อนหน้า ไม่เป็นข้อมูลนำเข้าไปในฟังก์ชันการทำงานถัดไป แสดงว่าไม่ตรงตามกฎความต้องกันกฎข้อที่ 3 แต่ตรงตามกฎความต้องกันข้อที่ 2 และกฎข้อที่ 4 เนื่องจากรหัสลูกค้ำเป็นข้อมูลนำออกจากฟังก์ชัน 2.2 แต่ไม่พบว่าเป็นข้อมูลนำเข้าของฟังก์ชัน 2.3
- 3) ไม่มีการแสดงข้อความผิดพลาดว่าฟังก์ชันการทำงานซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 5
- 4) ไม่มีการแสดงข้อความผิดพลาดว่าความต้องการด้านหน้าที่ซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 6
- 5) สำหรับกฎความต้องกันกฎข้อที่ 7 เครื่องมือซอฟต์แวร์จะแสดงรายละเอียดของข้อมูลที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล ว่าประกอบด้วยข้อมูลย่อยใดบ้าง จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-1 เช่น คำว่า “ข้อมูลลูกค้ำ”



รูปที่ 5.16 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน

5.1.3 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2 (เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ที่มีการแก้ไขให้เกิดความไม่ต้งกันขึ้น)


เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2 นี้ สร้างจากการแก้ไขความต้องการด้านหน้าที่ที่ 2 ของเอกสารตัวอย่างที่ 1 ในส่วนของข้อมูลนำเข้า กระบวนการ ข้อมูลนำออก และเพิ่มความต้งการด้านหน้าที่ที่ 4 รายละเอียดของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2 แสดงในตารางที่ 5.5

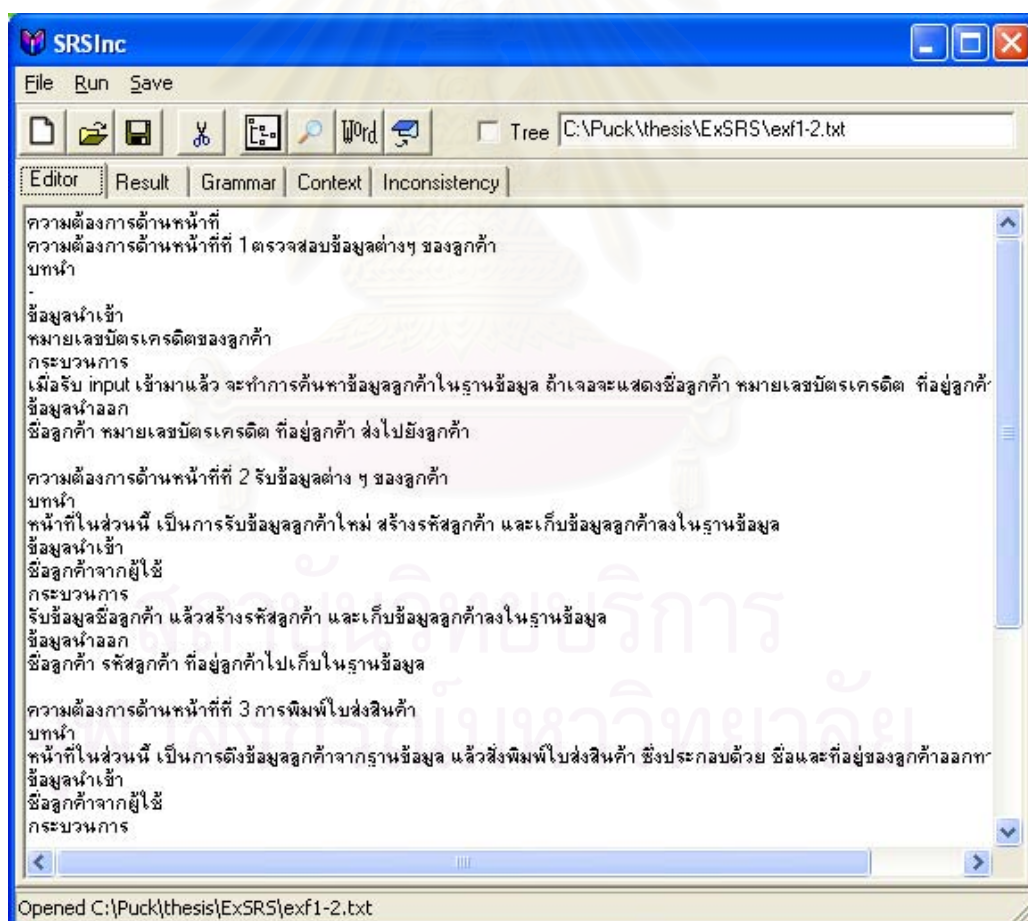
ตารางที่ 5.5 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2

ความต้องการด้านหน้าที่ที่ 1 ตรวจสอบข้อมูลต่างๆ ของลูกค้า	
บทนำ	-
ข้อมูลนำเข้า	หมายเลขบัตรเครดิตของลูกค้า
กระบวนการ	เมื่อรับ input เข้ามาแล้ว จะทำการค้นหาข้อมูลลูกค้าในฐานข้อมูล ถ้าเจอจะแสดงชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้า
ข้อมูลนำออก	ชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้าส่งไปยังลูกค้า
ความต้องการหน้าที่ที่ 2 รับข้อมูลต่าง ๆ ของลูกค้า	
บทนำ	หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับข้อมูลชื่อลูกค้า แล้วสร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล
ข้อมูลนำออก	ชื่อลูกค้า รหัสลูกค้า ที่อยู่ลูกค้าไปเก็บในฐานข้อมูล
ความต้องการหน้าที่ที่ 3 การพิมพ์ใบส่งสินค้า	
บทนำ	หน้าที่ในส่วนนี้ เป็นการดึงข้อมูลลูกค้าจากฐานข้อมูล แล้วส่งพิมพ์ใบส่งสินค้า ซึ่งประกอบด้วย ชื่อและที่อยู่ของลูกค้าออกทางเครื่องพิมพ์
ข้อมูลนำเข้า	ชื่อลูกค้าจากผู้ใช้
กระบวนการ	รับชื่อลูกค้า เพื่อดึงข้อมูลที่อยู่ของลูกค้าจากฐานข้อมูล แล้วพิมพ์ใบส่งสินค้า
ข้อมูลนำออก	ชื่อและที่อยู่ของลูกค้าไปยังผู้ใช้



ตารางที่ 5.5 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2 (ต่อ)

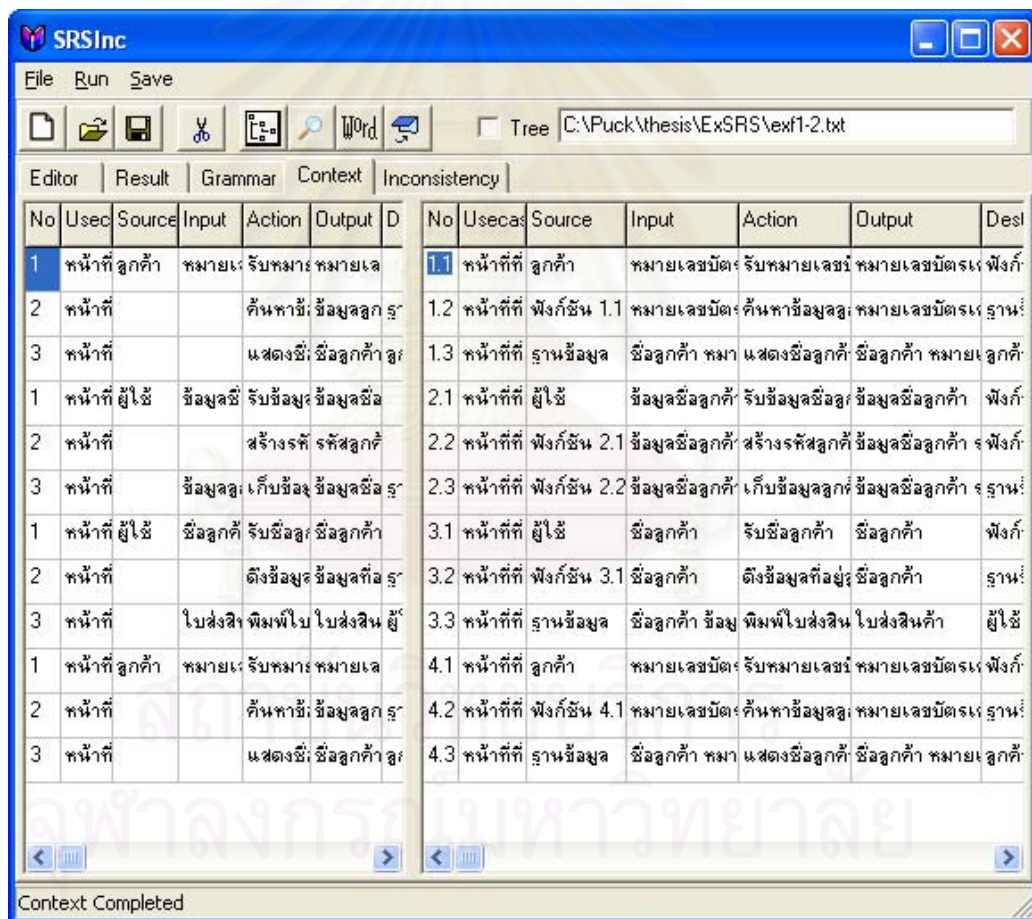
ความต้องการด้านหน้าที่ที่ 4 ตรวจสอบข้อมูลลูกค้า	
บทนำ	-
ข้อมูลนำเข้า	หมายเลขบัตรเครดิตของลูกค้า
กระบวนการ	รับหมายเลขบัตรเครดิต เพื่อค้นหาข้อมูลลูกค้าในฐานข้อมูล แล้วแสดงชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้า
ข้อมูลนำออก	ชื่อลูกค้า หมายเลขบัตรเครดิต ที่อยู่ลูกค้าส่งไปยังลูกค้า

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน หลังจากเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกเมนู Open SRS เพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1-2 ผลลัพธ์การทำงานแสดงดังรูปที่ 5.17



รูปที่ 5.17 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-2

เนื่องจากการทำงานในลำดับถัดไปของเครื่องมือซอฟต์แวร์ ซึ่งประกอบด้วย การเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร และการเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ หน้าจอผลลัพธ์การทำงานที่ได้จากการตัดคำและกำกับหน้าที่คำ รวมถึงส่วนประกอบของประโยคที่ได้จากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะคล้ายคลึงกับ รายละเอียดที่อธิบายไปแล้วในการตรวจสอบความไม่ตรงกันของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 1 ดังนั้นในส่วนนี้จะละรายละเอียดไว้ โดยจะแสดงเพียงผลลัพธ์ของโครงสร้างใหม่ที่ได้ ดังรูปที่ 5.18 และผลการตรวจสอบความไม่ตรงกัน ดังรูปที่ 5.19 เท่านั้น




No	Used	Source	Input	Action	Output	D	No	Used	Source	Input	Action	Output	Des
1	หน้าที่ ลูกค้า	หมายเลข	รับหมาย	หมายเลข	หมายเลข		1.1	หน้าที่ที่	ลูกค้า	หมายเลขบัตร	รับหมายเลข	หมายเลขบัตร	ฟังก์
2	หน้าที่		ค้นหา	ข้อมูล	สร		1.2	หน้าที่ที่	ฟังก์ชัน 1.1	หมายเลขบัตร	ค้นหาข้อมูล	หมายเลขบัตร	ฐาน
3	หน้าที่		แสดง	ข้อมูล	สร		1.3	หน้าที่ที่	ฐานข้อมูล	ข้อมูล	ค้นหา	แสดงข้อมูล	ข้อมูล
1	หน้าที่ ผู้ใช้	ข้อมูล	รับข้อมูล	ข้อมูล	สร		2.1	หน้าที่ที่	ผู้ใช้	ข้อมูล	รับข้อมูล	ข้อมูล	ฟังก์
2	หน้าที่		สร้าง	รหัส	สร		2.2	หน้าที่ที่	ฟังก์ชัน 2.1	ข้อมูล	สร้างรหัส	ข้อมูล	ฟังก์
3	หน้าที่	ข้อมูล	เก็บ	ข้อมูล	สร		2.3	หน้าที่ที่	ฟังก์ชัน 2.2	ข้อมูล	เก็บข้อมูล	ข้อมูล	ฐาน
1	หน้าที่ ผู้ใช้	ข้อมูล	รับข้อมูล	ข้อมูล	สร		3.1	หน้าที่ที่	ผู้ใช้	ข้อมูล	รับข้อมูล	ข้อมูล	ฟังก์
2	หน้าที่		ดึง	ข้อมูล	สร		3.2	หน้าที่ที่	ฟังก์ชัน 3.1	ข้อมูล	ดึงข้อมูล	ข้อมูล	ฐาน
3	หน้าที่	ใบส่ง	พิมพ์	ใบส่ง	สร		3.3	หน้าที่ที่	ฐานข้อมูล	ข้อมูล	พิมพ์	ใบส่ง	ผู้ใช้
1	หน้าที่ ลูกค้า	หมายเลข	รับหมาย	หมายเลข	หมายเลข		4.1	หน้าที่ที่	ลูกค้า	หมายเลขบัตร	รับหมายเลข	หมายเลขบัตร	ฟังก์
2	หน้าที่		ค้นหา	ข้อมูล	สร		4.2	หน้าที่ที่	ฟังก์ชัน 4.1	หมายเลขบัตร	ค้นหาข้อมูล	หมายเลขบัตร	ฐาน
3	หน้าที่		แสดง	ข้อมูล	สร		4.3	หน้าที่ที่	ฐานข้อมูล	ข้อมูล	ค้นหา	แสดงข้อมูล	ข้อมูล

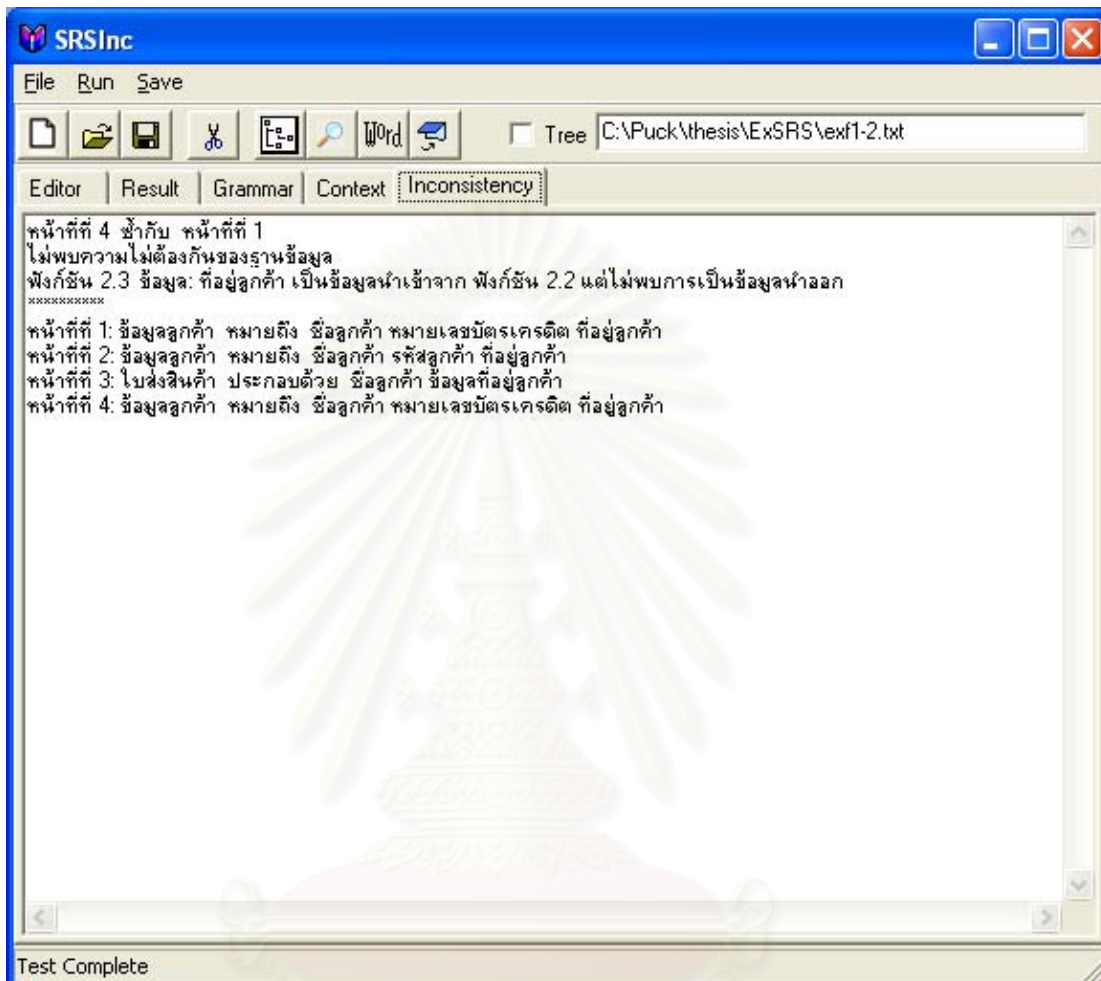
รูปที่ 5.18 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

รายละเอียดโครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 1-2 แสดงในตารางที่

ตารางที่ 5.6 โครงสร้างใหม่ของความต้องการด้านหน้าที่ตัวอย่างที่ 1-2

No.	Use case	Source	Input	Action	Output	Destination
1.1	หน้าที่ที่ 1	ลูกค้า	หมายเลขบัตร เครดิต	รับหมายเลขบัตร เครดิต	หมายเลขบัตร เครดิต	ฟังก์ชัน 1.2
1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	หมายเลขบัตร เครดิต	ค้นหาข้อมูลลูก ค้า	หมายเลขบัตร เครดิต	ฐานข้อมูล
1.3	หน้าที่ที่ 1	ฐานข้อมูล	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	แสดงชื่อลูกค้า หมายเลขบัตร เครดิต ที่อยู่ลูกค้า	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	ลูกค้า
2.1	หน้าที่ที่ 2	ผู้ใช้	ข้อมูลชื่อลูกค้า	รับข้อมูลชื่อลูกค้า	ข้อมูลชื่อลูกค้า	ฟังก์ชัน 2.2
2.2	หน้าที่ที่ 2	ฟังก์ชัน 2.1	ข้อมูลชื่อลูกค้า	สร้างรหัสลูกค้า	ข้อมูลชื่อลูกค้า รหัสลูกค้า	ฟังก์ชัน 2.3
2.3	หน้าที่ที่ 3	ฟังก์ชัน 2.2	ข้อมูลชื่อลูกค้า รหัสลูกค้า ที่อยู่ลูกค้า	เก็บข้อมูลลูกค้า	รหัสลูกค้า ที่อยู่ลูกค้า	ฐานข้อมูล
3.1	หน้าที่ที่ 3	ผู้ใช้	ชื่อลูกค้า	รับชื่อลูกค้า	ชื่อลูกค้า	ฟังก์ชัน 3.2
3.2	หน้าที่ที่ 3	ฟังก์ชัน 3.1	ชื่อลูกค้า	ดึงข้อมูลที่อยู่ลูก ค้า	ชื่อลูกค้า	ฐานข้อมูล
3.3	หน้าที่ที่ 3	ฐานข้อมูล	ชื่อลูกค้า ข้อมูลที่อยู่ลูก ค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้
4.1	หน้าที่ที่ 4	ลูกค้า	หมายเลขบัตร เครดิต	รับหมายเลขบัตร เครดิต	หมายเลขบัตร เครดิต	ฟังก์ชัน 4.2
4.2	หน้าที่ที่ 4	ฟังก์ชัน 4.1	หมายเลขบัตร เครดิต	ค้นหาข้อมูลลูก ค้า	หมายเลขบัตร เครดิต	ฐานข้อมูล
4.3	หน้าที่ที่ 4	ฐานข้อมูล	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	แสดงชื่อลูกค้า หมายเลขบัตร เครดิต ที่อยู่ลูกค้า	ชื่อลูกค้า หมายเลข เลขบัตรเครดิต ที่อยู่ลูกค้า	ลูกค้า

จากรูปที่ 5.18 เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Check Inconsistency ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตรวจสอบความไม่ตรงกันของเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์ตัวอย่างที่ 1-2 ผลลัพธ์แสดงดังรูปที่ 5.19



รูปที่ 5.19 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน

รายละเอียดผลการตรวจสอบความไม่ตรงกัน ได้แก่

- 1) ไม่พบความไม่ตรงกันของข้อมูลในฐานข้อมูล แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 1
- 2) พบความไม่ตรงกันของข้อมูลนำเข้าในฟังก์ชันการทำงานถัดไป ไม่เป็นข้อมูลนำออกจากฟังก์ชันการทำงานก่อนหน้า แสดงว่าไม่ตรงตามกฎความต้องกันกฎข้อที่ 4 แต่ตรงตามกฎความต้องกันข้อที่ 2 – กฎข้อที่ 3 เนื่องจากข้อมูลที่อยู่ลูกค้าเป็นข้อมูลนำเข้าของฟังก์ชัน 2.3 แต่ไม่พบการเป็นข้อมูลนำออกจากฟังก์ชัน 2.2
- 3) ไม่มีการแสดงข้อความผิดพลาดว่าฟังก์ชันการทำงานซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 5

- 4) มีการแสดงข้อความผิดพลาดว่าเกิดความต้องการด้านหน้าที่ซ้ำซ้อนกัน แสดงว่าไม่ตรงตามกฎความต้องการกันกฎข้อที่ 6 เนื่องจากความต้องการด้านหน้าที่ที่ 4 มีการทำงานซ้ำซ้อนกับความต้องการด้านหน้าที่ที่ 1
- 5) สำหรับกฎความต้องการกันกฎข้อที่ 7 เครื่องมือซอฟต์แวร์จะแสดงรายละเอียดของข้อมูล ที่หมายถึงกลุ่มข้อมูลในฐานะข้อมูล ว่าประกอบด้วยข้อมูลย่อยใดบ้าง จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 1-2 เช่น คำว่า “ข้อมูลลูกค้า”

5.2 การตรวจสอบความไม่ต้องการกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2


การตรวจสอบความไม่ต้องการกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 จะทำการตรวจสอบทั้งตัวอย่างเอกสารที่ 2 ที่ยังไม่มีการแก้ไขให้เกิดความไม่ต้องการกัน และตัวอย่างเอกสารที่ 2 ที่ทำการแก้ไขให้เกิดความไม่ต้องการกัน

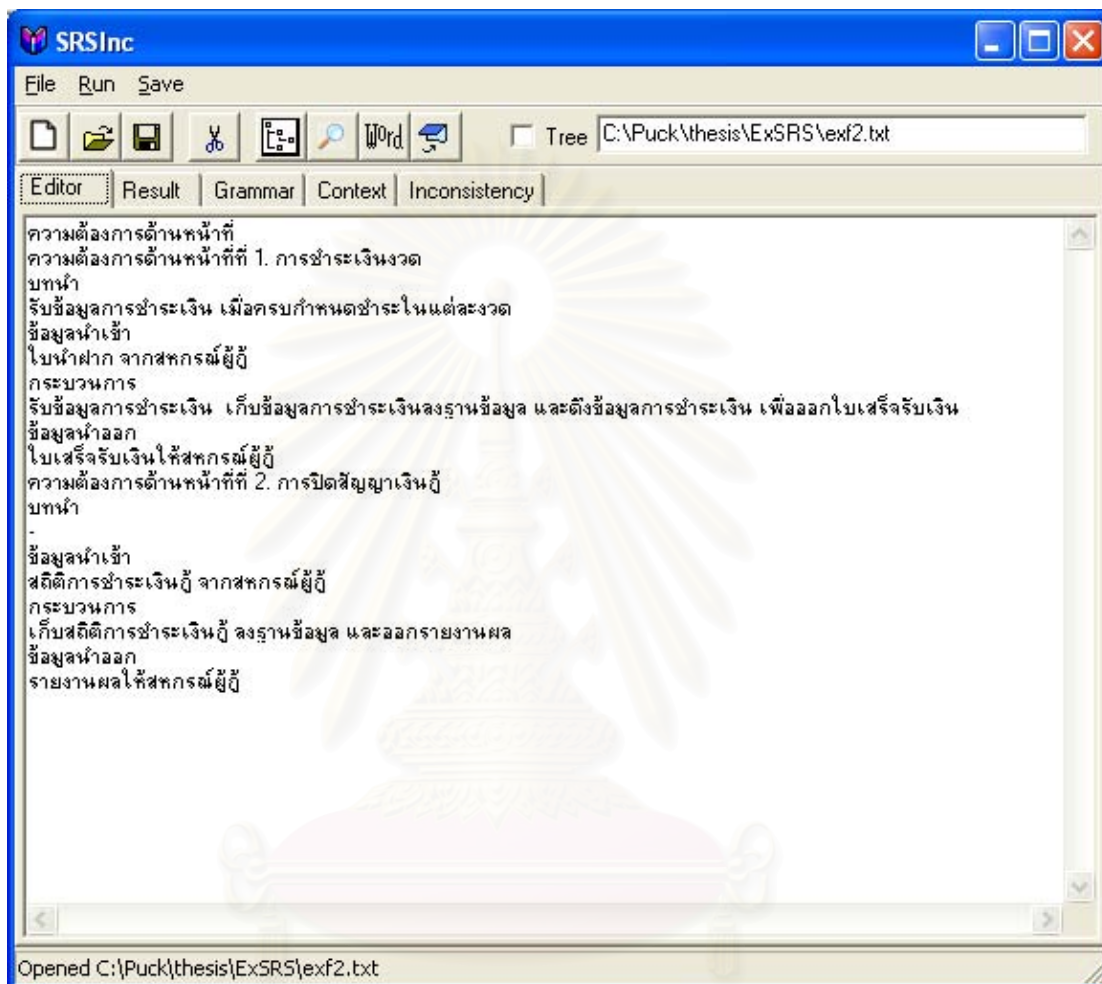
5.2.1 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 ที่ไม่มีการแก้ไขให้เกิดความไม่ต้องการกัน

รายละเอียดของตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 แสดงในตารางที่ 5.7


ตารางที่ 5.7 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2

ความต้องการด้านหน้าที่ที่ 1 การชำระเงินงวด	
บทนำ	รับข้อมูลการชำระเงิน เมื่อครบกำหนดชำระในแต่ละงวด
ข้อมูลนำเข้า	ใบนำฝาก จากสหกรณ์ผู้กู้
กระบวนการ	รับข้อมูลการชำระเงิน เก็บข้อมูลการชำระเงินลงฐานข้อมูล และดึงข้อมูลการชำระเงิน เพื่อออกใบเสร็จรับเงิน
ข้อมูลนำออก	ใบเสร็จรับเงินให้สหกรณ์ผู้กู้
ความต้องการด้านหน้าที่ที่ 2. การปิดสัญญาเงินกู้	
บทนำ	-
ข้อมูลนำเข้า	สถิติการชำระเงินกู้ จากสหกรณ์ผู้กู้
กระบวนการ	เก็บสถิติการชำระเงินกู้ ลงฐานข้อมูล และออกรายงานผล
ข้อมูลนำออก	รายงานผลให้สหกรณ์ผู้กู้

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกแถบเมนู Open SRS เพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 ผลลัพธ์การทำงานแสดงดังรูปที่ 5.20




รูปที่ 5.20 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์
ด้านหน้าที่ตัวอย่างที่ 2

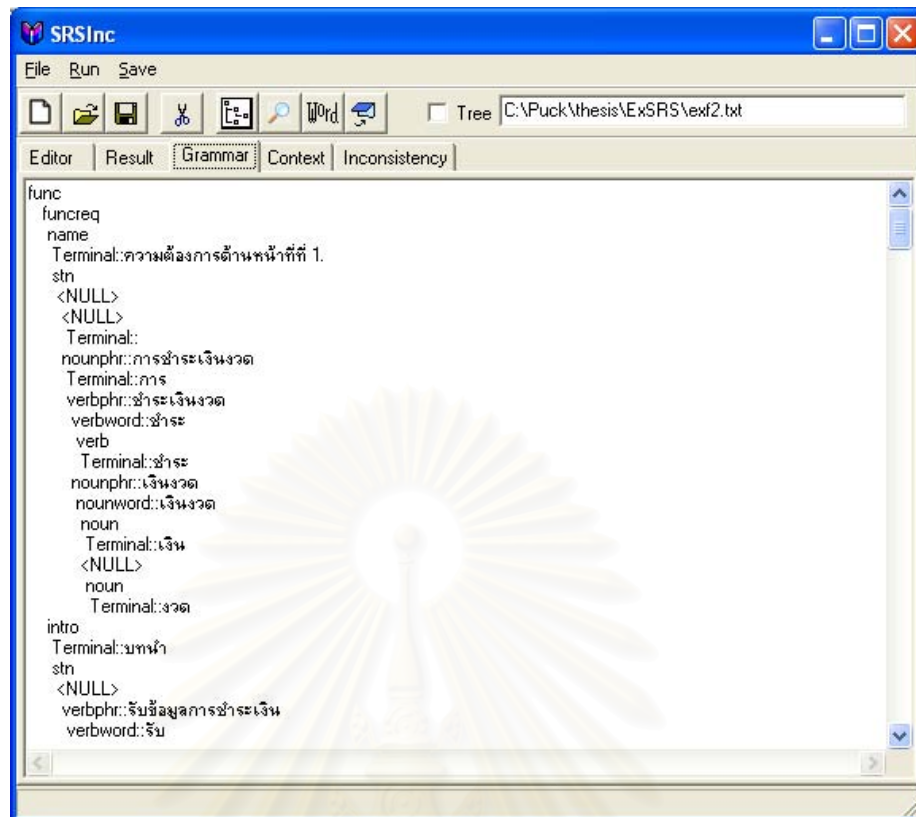
เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร ผลลัพธ์แสดงดังรูปที่ 5.21

No	Part	Word	Tag
1	1-0	ความต้องการด้านหน้าที่ 1.	ความต้องการด้านหน้าที่
2	1-0		SPAC
3	1-0	การ	FIXN
4	1-0	ชำระ	VACT
5	1-0	เงิน	NCMN
6	1-0	งวด	NCMN
7	1-1	บทนำ	บทนำ
8	1-1	รับ	VACT
9	1-1	ซื้อ	NCMN
10	1-1	การ	FIXN
11	1-1	ชำระ	VACT
12	1-1	เงิน	NCMN
13	1-1		SPAC
14	1-1	เมื่อ	JSBR

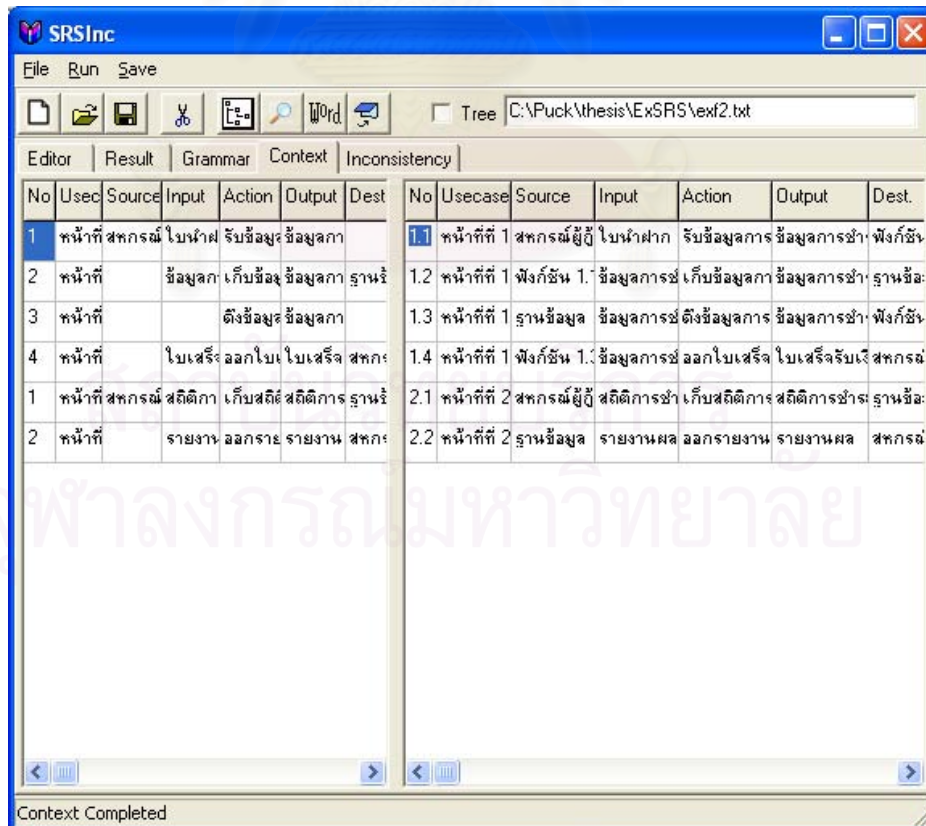
รูปที่ 5.21 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ

เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ โดยการดึงส่วนสำคัญของประโยคที่อธิบายรายละเอียดความต้องการด้านหน้าที่ ผลลัพธ์การวิเคราะห์ประโยคด้วยไวยากรณ์แสดงดังรูปที่ 5.22

หลังจากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะทำการดึงส่วนสำคัญของประโยคที่เกี่ยวข้องกับคำหรือวลีสำคัญในการอธิบายความต้องการด้านหน้าที่ เพื่อสร้างโครงสร้างใหม่ที่ใช้อธิบายแทนความต้องการด้านหน้าที่ที่เดิมเป็นข้อความ ผลลัพธ์การทำงานของเครื่องมือซอฟต์แวร์แสดงดังรูปที่ 5.23



รูปที่ 5.22 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์




รูปที่ 5.23 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

รายละเอียดโครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 2 แสดงในตารางที่ 5.8

ตารางที่ 5.8 โครงสร้างใหม่ของความต้องการด้านหน้าที่ตัวอย่างที่ 2

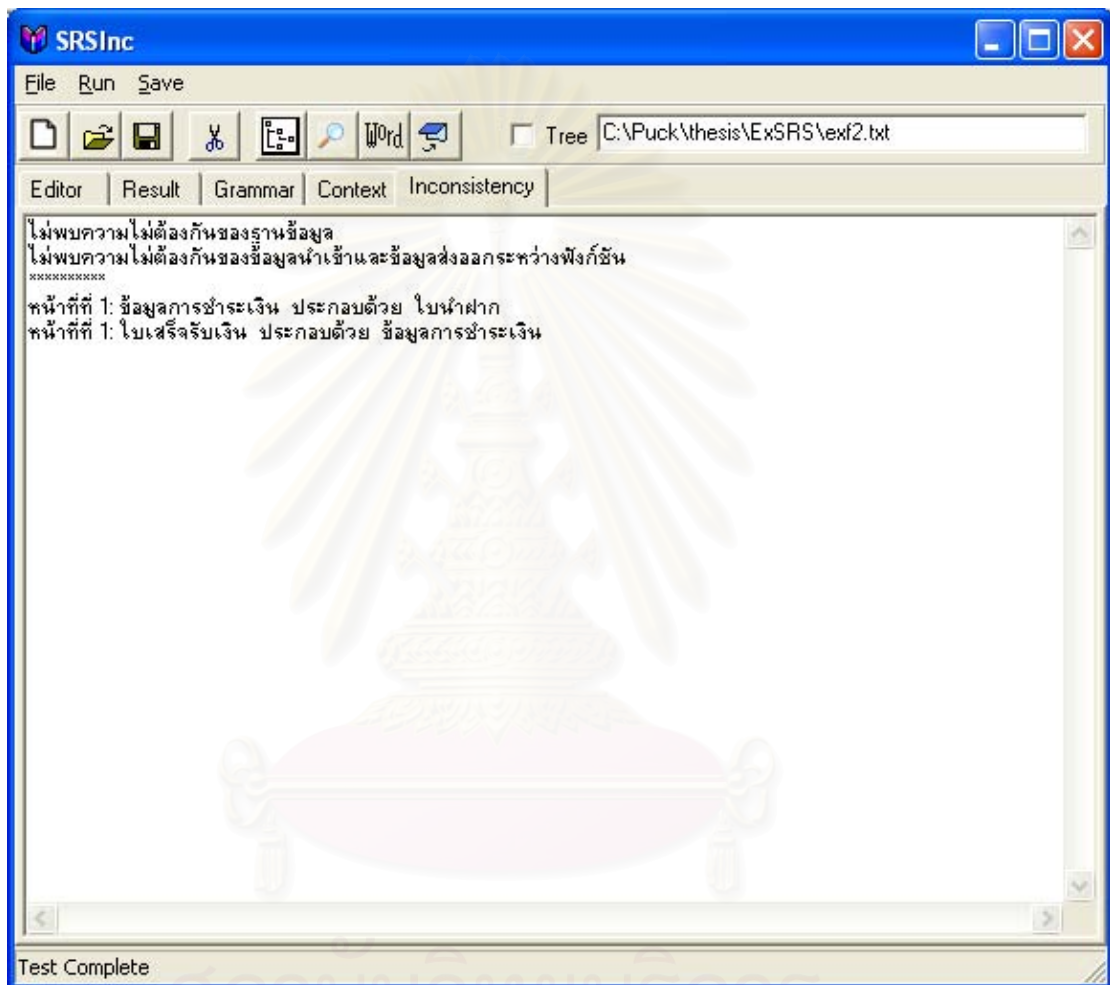
No.	Use case	Source	Input	Action	Output	Destination
1.1	หน้าที่ที่ 1	สหกรณ์ผู้กู้	ใบนำฝาก	รับข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฟังก์ชัน 1.2
1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	ข้อมูล การชำระเงิน	เก็บข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฐานข้อมูล
1.3	หน้าที่ที่ 1	ฐานข้อมูล	ข้อมูล การชำระเงิน	ดึงข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฟังก์ชัน 1.4
1.4	หน้าที่ที่ 1	ฟังก์ชัน 1.3	ข้อมูล การชำระเงิน	ออกไปเสร็จ รับเงิน	ใบเสร็จรับเงิน	สหกรณ์ผู้กู้
2.1	หน้าที่ที่ 2	สหกรณ์ผู้กู้	สถิติการชำระ เงินกู้	เก็บสถิติการชำระ เงินกู้	สถิติการชำระ เงินกู้	ฐานข้อมูล
2.2	หน้าที่ที่ 2	ฐานข้อมูล	รายงานผล	ออกรายงานผล	รายงานผล	สหกรณ์ผู้กู้

จากรูปที่ 5.23 เลือกรูปเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Check Inconsistency ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตรวจสอบความไม่ตรงกันของเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์ตัวอย่างที่ 2 ผลลัพธ์แสดงดังรูปที่ 5.24

รายละเอียดผลการตรวจสอบความไม่ตรงกัน ได้แก่

- 1) ไม่พบความไม่ตรงกันของข้อมูลในฐานข้อมูล แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 1
- 2) ไม่พบความไม่ตรงกันของข้อมูลนำเข้าและข้อมูลส่งออกระหว่างฟังก์ชัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 2 – กฎข้อที่ 4
- 3) ไม่มีการแสดงข้อความผิดพลาดว่าฟังก์ชันการทำงานซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 5
- 4) ไม่มีการแสดงข้อความผิดพลาดว่าความต้องการด้านหน้าที่ซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 6

- 5) สำหรับกฎความต้องกันกฎข้อที่ 7 เครื่องมือซอฟต์แวร์จะแสดงรายละเอียดของข้อมูลที่หมายถึงกลุ่มข้อมูลในฐานข้อมูล ว่าประกอบด้วยข้อมูลย่อยใดบ้าง จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 2 เช่น คำว่า “ข้อมูลการชำระเงิน” ในฟังก์ชัน 1.1 หมายถึง โอนฝาก และในฟังก์ชัน 1.4 โอนเสร็จรับเงินที่ออก ก็คือข้อมูลการชำระเงิน เช่นกัน




รูปที่ 5.24 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ต้องกัน

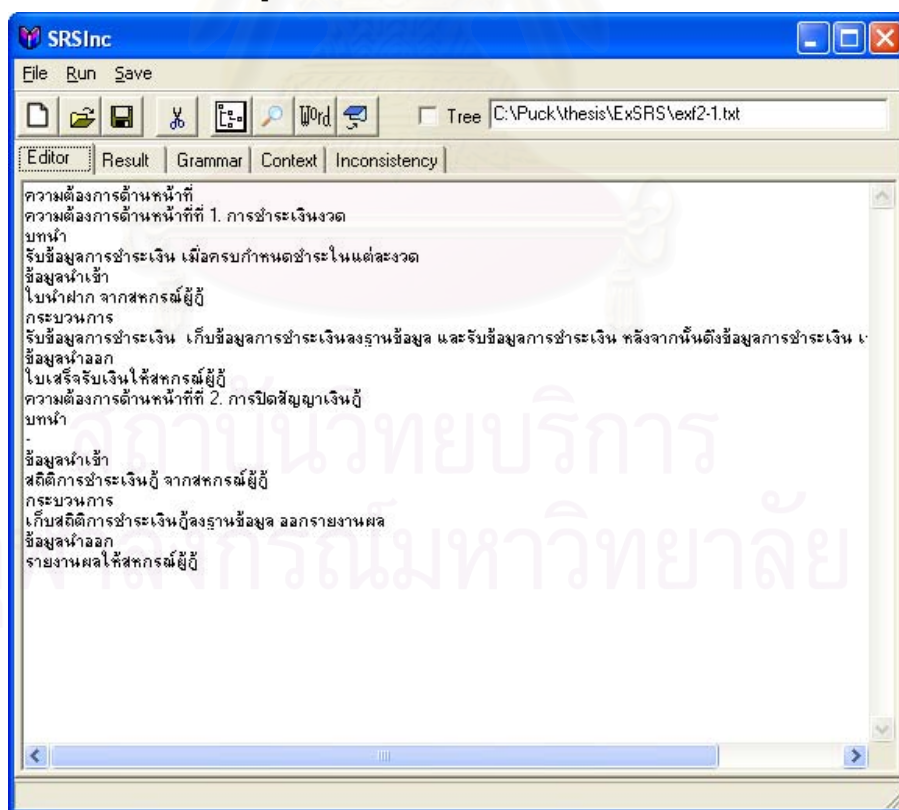
5.2.2 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2-1 (เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 ที่มีการแก้ไขให้เกิดความไม่ต้องกันขึ้น)

เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2-1 นี้ สร้างจากการแก้ไขความต้องการด้านหน้าที่ที่ 1 ของเอกสารตัวอย่างที่ 2 ในส่วนของกระบวนการ รายละเอียดของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2-1 แสดงในตารางที่ 5.9



ตารางที่ 5.9 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2-1

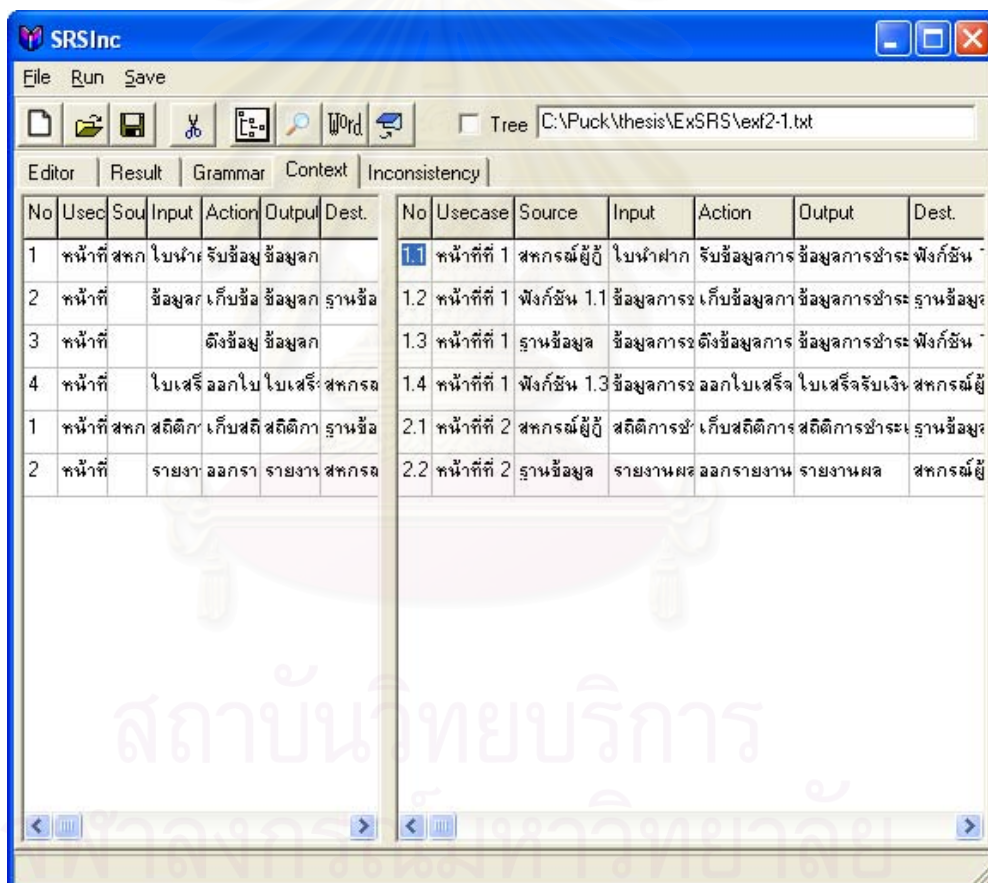
ความต้องการด้านหน้าที่ที่ 1 การชำระเงินงวด	
บทนำ	รับข้อมูลการชำระเงิน เมื่อครบกำหนดชำระในแต่ละงวด
ข้อมูลนำเข้า	ใบนำฝาก จากสหกรณ์ผู้กู้
กระบวนการ	รับข้อมูลการชำระเงิน เก็บข้อมูลการชำระเงินลงฐานข้อมูล และรับข้อมูลการชำระเงิน หลังจากนั้นดึงข้อมูลการชำระเงิน เพื่อออกใบเสร็จรับเงิน
ข้อมูลนำออก	ใบเสร็จรับเงินให้สหกรณ์ผู้กู้
ความต้องการด้านหน้าที่ที่ 2. การปิดสัญญาเงินกู้	
บทนำ	-
ข้อมูลนำเข้า	สถิติการชำระเงินกู้ จากสหกรณ์ผู้กู้
กระบวนการ	เก็บสถิติการชำระเงินกู้ ลงฐานข้อมูล และออกรายงานผล
ข้อมูลนำออก	รายงานผลให้สหกรณ์ผู้กู้

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน หลังจากเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกเมนู Open SRS เพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2-1 ผลลัพธ์การทำงานแสดงดังรูปที่ 5.25



รูปที่ 5.25 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 2-1

เนื่องจากการทำงานในลำดับถัดไปของเครื่องมือซอฟต์แวร์ ซึ่งประกอบด้วย การเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร และการเลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ หน้าจอผลลัพธ์การทำงานที่ได้จากการตัดคำและกำกับหน้าที่คำ รวมถึงส่วนประกอบของประโยคที่ได้จากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะคล้ายคลึงกับ รายละเอียดที่อธิบายไปแล้วในการตรวจสอบความไม่ต้องการของเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 2 ดังนั้นในส่วนนี้จะละรายละเอียดไว้ โดยจะแสดงเพียงผลลัพธ์ของโครงสร้างใหม่ที่ได้ ดังรูปที่ 5.26 และผลการตรวจสอบความไม่ต้องการ ดังรูปที่ 5.27 เท่านั้น




No	Used	Sou	Input	Action	Output	Dest.	No	Usecase	Source	Input	Action	Output	Dest.
1	หน้าที่ สกท	ใบนำ	รับข้อมูล	ข้อมูล			1.1	หน้าที่ที่ 1	สภกรณ์ผู้	ใบนำฝาก	รับข้อมูลการ	ข้อมูลการชำระ	ฟังก์ชัน
2	หน้าที่	ข้อมูล	เก็บข้อมูล	ข้อมูล	ฐานข้อมูล		1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	ข้อมูลการ	เก็บข้อมูล	ข้อมูลการชำระ	ฐานข้อมูล
3	หน้าที่		ดึงข้อมูล	ข้อมูล			1.3	หน้าที่ที่ 1	ฐานข้อมูล	ข้อมูลการ	ดึงข้อมูล	ข้อมูลการชำระ	ฟังก์ชัน
4	หน้าที่	ใบเสร็จ	ออกใบ	ใบเสร็จ	สภกรณ์		1.4	หน้าที่ที่ 1	ฟังก์ชัน 1.3	ข้อมูลการ	ออกใบเสร็จ	ใบเสร็จรับเงิน	สภกรณ์ผู้
1	หน้าที่ สกท	สถิติ	เก็บสถิติ	สถิติ	ฐานข้อมูล		2.1	หน้าที่ที่ 2	สภกรณ์ผู้	สถิติการ	เก็บสถิติ	สถิติการชำระ	ฐานข้อมูล
2	หน้าที่	รายงาน	ออก	รายงาน	สภกรณ์		2.2	หน้าที่ที่ 2	ฐานข้อมูล	รายงาน	ออก	รายงาน	รายงานผล

รูปที่ 5.26 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

รายละเอียดโครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 2-1 แสดงในตารางที่ 5.10 จะเห็นได้ว่า โครงสร้างใหม่ที่ได้เหมือนกับโครงสร้างใหม่ของความต้องการด้านหน้าที่ ตัวอย่างที่ 2 เนื่องจากเครื่องมือซอฟต์แวร์มีการตรวจสอบความซ้ำซ้อนของฟังก์ชันการทำงานในแต่ละความต้องการด้านหน้าที่ โดยจะไม่แสดงฟังก์ชันการทำงานที่ซ้ำซ้อนในโครงสร้างใหม่ แต่จะมีการรายงานข้อผิดพลาด ซึ่งอธิบายในส่วนถัดไป

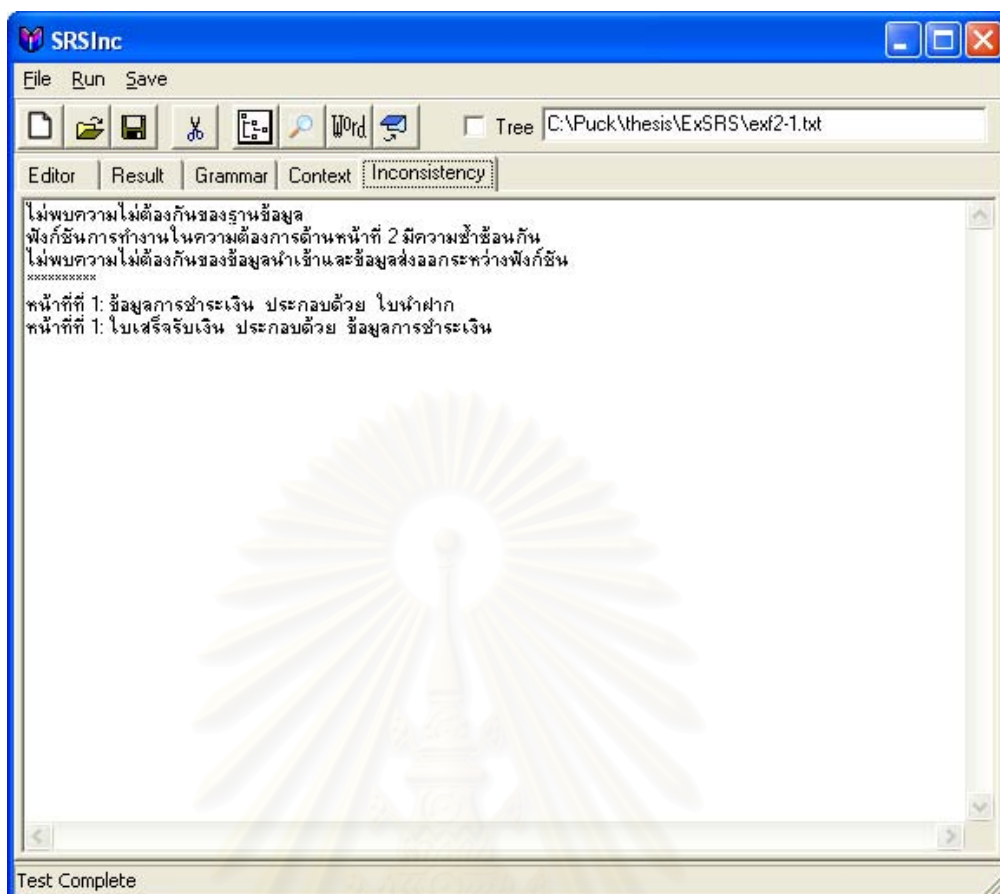
ตารางที่ 5.10 โครงสร้างใหม่ของการต้องการด้านหน้าที่ตัวอย่างที่ 2-1

No.	Use case	Source	Input	Action	Output	Destination
1.1	หน้าที่ที่ 1	สหกรณ์ผู้กู้	ใบนำฝาก	รับข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฟังก์ชัน 1.2
1.2	หน้าที่ที่ 1	ฟังก์ชัน 1.1	ข้อมูล การชำระเงิน	เก็บข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฐานข้อมูล
1.3	หน้าที่ที่ 1	ฐานข้อมูล	ข้อมูล การชำระเงิน	ดึงข้อมูลการ ชำระเงิน	ข้อมูล การชำระเงิน	ฟังก์ชัน 1.4
1.4	หน้าที่ที่ 1	ฟังก์ชัน 1.3	ข้อมูล การชำระเงิน	ออกไปเสร็จ รับเงิน	ใบเสร็จรับเงิน	สหกรณ์ผู้กู้
2.1	หน้าที่ที่ 2	สหกรณ์ผู้กู้	สถิติการชำระ เงินกู้	เก็บสถิติการชำระ เงินกู้	สถิติการชำระ เงินกู้	ฐานข้อมูล
2.2	หน้าที่ที่ 2	ฐานข้อมูล	รายงานผล	ออกรายงานผล	รายงานผล	สหกรณ์ผู้กู้

จากรูปที่ 5.26 เลือกรุ่นมือ  ในแถบเครื่องมือ หรือเมนู Check Inconsistency ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตรวจสอบความไม่ตรงกันของเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์ตัวอย่างที่ 2-1 ผลลัพธ์แสดงดังรูปที่ 5.27

รายละเอียดผลการตรวจสอบความไม่ตรงกัน ได้แก่

- 1) ไม่พบความไม่ตรงกันของข้อมูลในฐานข้อมูล แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 1
- 2) ไม่พบความไม่ตรงกันของข้อมูลนำเข้าและข้อมูลส่งออกระหว่างฟังก์ชัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 2 – กฎข้อที่ 4
- 3) มีการแสดงข้อความผิดพลาดว่าฟังก์ชันการทำงานซ้ำซ้อนกัน แสดงว่าไม่ตรงตามกฎความต้องกันกฎข้อที่ 5 เนื่องจากในความต้องการด้านหน้าที่ที่ 1
- 4) ไม่มีการแสดงข้อความผิดพลาดว่าความต้องการด้านหน้าที่ซ้ำซ้อนกัน แสดงว่าตรงตามกฎความต้องกันกฎข้อที่ 6
- 5) สำหรับกฎความต้องกันกฎข้อที่ 7 เครื่องมือซอฟต์แวร์จะแสดงรายละเอียดของข้อมูลที่เกี่ยวข้องถึงกลุ่มข้อมูลในฐานข้อมูล ว่าประกอบด้วยข้อมูลย่อยใดบ้าง จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ ตัวอย่างที่ 2-1 เช่น คำว่า “ข้อมูลการชำระเงิน” ในฟังก์ชัน 1.1 หมายถึง ใบนำฝาก และในฟังก์ชัน 1.4 ใบเสร็จรับเงินที่ออก ก็คือข้อมูลการชำระเงิน เช่นกัน



รูปที่ 5.27 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน

5.3 การตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 3


ตัวอย่างเอกสารนี้เป็นตัวอย่างเอกสารที่ไม่เหมาะสมที่จะนำมาใช้กับเครื่องมือซอฟต์แวร์ที่งานวิจัยนี้พัฒนาขึ้นมา เนื่องจากรูปแบบการเขียนมีความซับซ้อน แต่ละประโยคไม่ได้แสดงถึงส่วนสำคัญของความต้องการด้านหน้าที่อย่างชัดเจน ตัวอย่างเช่น มีการใช้คำและประโยคฟุ่มเฟือย มีประโยคที่ไม่ใช่การอธิบายฟังก์ชันการทำงานของซอฟต์แวร์อธิบายอยู่ในส่วนของกระบวนการจำนวนมาก มีการใช้คำศัพท์ภาษาอังกฤษในการเขียนเอกสาร โครงสร้างประโยคที่ใช้มีความซับซ้อน ดังนั้นโครงสร้างใหม่ที่ใช้อธิบายความต้องการด้านหน้าที่ที่ได้จากเอกสารมีความไม่ถูกต้องเนื่องจากเครื่องมือซอฟต์แวร์ ไม่สามารถแยกแยะประโยคที่เกี่ยวข้องกับฟังก์ชันการทำงาน และไม่เกี่ยวข้องได้อย่างชัดเจน รวมถึงไม่สามารถดึงคำสำคัญในประโยคที่เกี่ยวข้องกับ ฟังก์ชันการทำงาน แหล่งที่มาของข้อมูล ข้อมูลนำเข้า ข้อมูลนำออก และแหล่งปลายทางของข้อมูลได้อย่างถูกต้อง


ตารางที่ 5.11 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 3

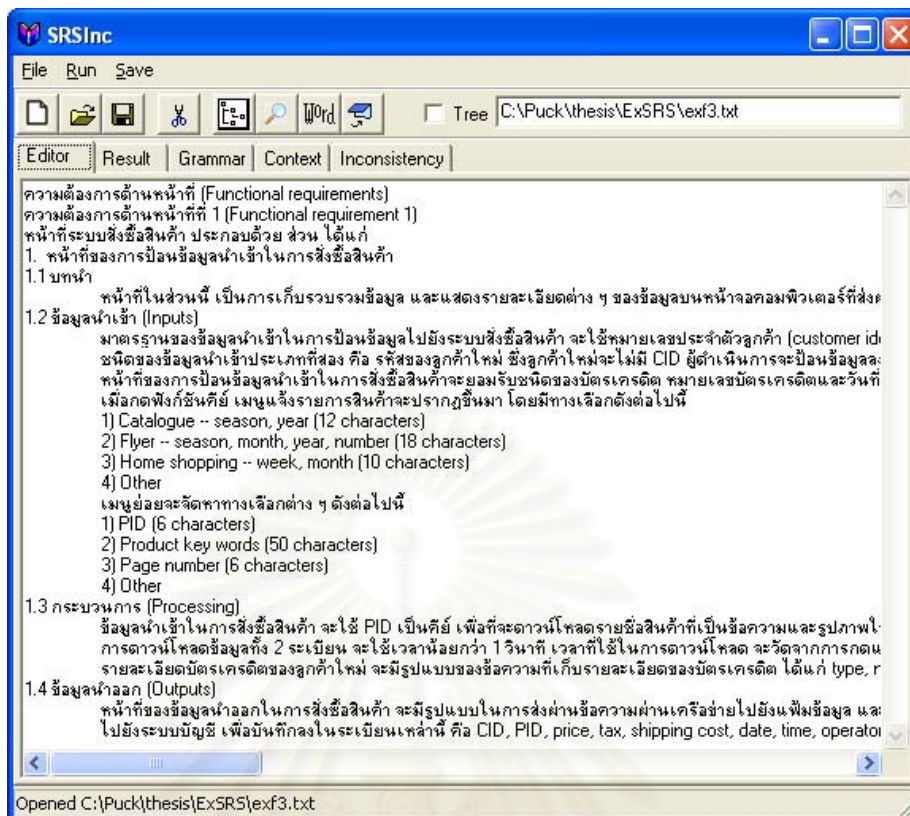
หน้าที่ที่ 1	หน้าที่ระบบสั่งซื้อสินค้า
บทนำ	หน้าที่ในส่วนนี้ เป็นการเก็บรวบรวมข้อมูล และแสดงรายละเอียดต่าง ๆ ของข้อมูลบนหน้าจอคอมพิวเตอร์ที่ส่งผลกระทบต่อการใช้ของลูกค้าและการขายโดยผู้ดำเนินการ หน้าที่ในการเข้าถึงฐานข้อมูลเป็นแบบอ่านอย่างเดียว และการแสดงรายละเอียดต่าง ๆ ไปยังผู้ดำเนินการ โดยรายละเอียดเหล่านี้จะได้รับการถ่ายทอดไปยังลูกค้าทางโทรศัพท์
ข้อมูลนำเข้า	<p>มาตรฐานของข้อมูลนำเข้าในการป้อนข้อมูลไปยังระบบสั่งซื้อสินค้า จะใช้หมายเลขประจำตัวลูกค้า (customer identification number: CID) และหมายเลขประจำตัวสินค้า (product identification number: PID) โดย CID จะเป็นอักขระตัวอักษรผสมกับตัวเลข (alphanumeric) 6 หลัก และ PID จะเป็นอักขระตัวอักษรผสมตัวเลข 12 หลัก ชนิดของข้อมูลนำเข้าประเภทที่สอง คือ รหัสของลูกค้าใหม่ ซึ่งลูกค้าใหม่จะไม่มี CID ผู้ดำเนินการจะป้อนข้อมูลลงในรายการเปลี่ยนแปลงของลูกค้าใหม่ เมนูจะแสดงหน้าจอของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้า เพื่อให้กรอกชื่อลูกค้า ที่อยู่ และเบอร์โทรศัพท์ รายละเอียดเหล่านี้ จะยังคงเก็บอยู่ในหน่วยความจำ จนกระทั่งทำการเปลี่ยนแปลงเสร็จสมบูรณ์</p> <p>หน้าที่ของการป้อนข้อมูลนำเข้าในการสั่งซื้อสินค้าจะยอมรับชนิดของบัตรเครดิต หมายเลขบัตรเครดิตและวันที่หมดอายุของบัตรเครดิต ผู้ดำเนินการจะป้อนคำสั่งต่าง ๆ เพื่อตรวจสอบความถูกต้องของบัตรเครดิต เมนูจะแสดงบัญชีรายชื่อชนิดของบัตรเครดิตที่ได้รับการยอมรับผู้ดำเนินการสามารถกดแป้นคีย์บอร์ดเพื่อเลือกชนิดของบัตรเครดิตได้ สำหรับหมายเลขบัตรเครดิต และวันที่หมดอายุของบัตรเครดิต จะเก็บอยู่ในระเบียบของลูกค้ารายนั้น</p> <p>เมื่อกดฟังก์ชันคีย์ เมนูแจ้งรายการสินค้าจะปรากฏขึ้นมา โดยมีทางเลือกดังต่อไปนี้</p> <ol style="list-style-type: none"> 1) Catalogue -- season, year (12 characters) 2) Flyer -- season, month, year, number (18 characters) 3) Home shopping -- week, month (10 characters) 4) Other <p>เมนูย่อยจะจัดหาทางเลือกต่าง ๆ ดังต่อไปนี้</p> <ol style="list-style-type: none"> 1) PID (6 characters) 2) Product key words (50 characters) 3) Page number (6 characters) 4) Other

ตารางที่ 5.11 เอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 3 (ต่อ)

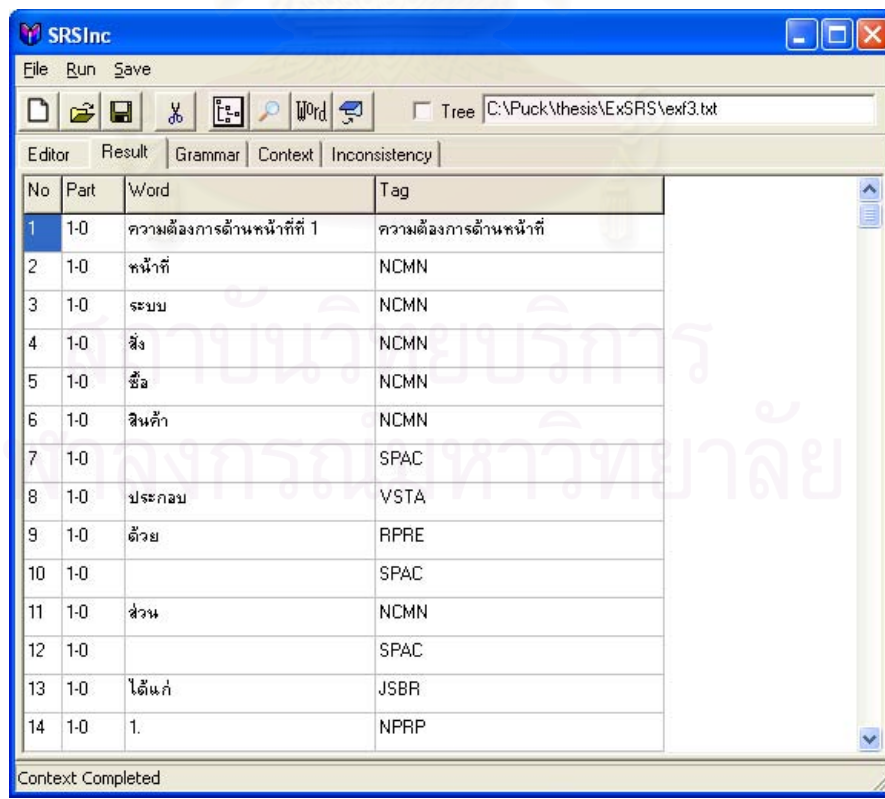
กระบวนการ	<p>ข้อมูลนำเข้าในการสั่งซื้อสินค้า จะใช้ PID เป็นคีย์ เพื่อที่จะดาวน์โหลดรายชื่อสินค้าที่เป็นข้อความและรูปภาพให้ปรากฏบนหน้าจอคอมพิวเตอร์ พร้อมทั้งจะดาวน์โหลดข้อมูลลูกค้าที่สมบูรณ์โดยใช้ CID เป็นคีย์</p> <p>การดาวน์โหลดข้อมูลทั้ง 2 ระเบียบ จะใช้เวลาน้อยกว่า 1 วินาที เวลาที่ใช้ในการดาวน์โหลด จะวัดจากการกดแป้นคีย์บอร์ดเพื่อป้อน PID หรือ CID ให้ไปปรากฏบนหน้าจอคอมพิวเตอร์</p> <p>รายละเอียดบัตรเครดิตของลูกค้าใหม่ จะมีรูปแบบของข้อความที่เก็บรายละเอียดของบัตรเครดิต ได้แก่ type, number, expiration, date, operator ID, station ID, date และ time ข้อความต่าง ๆ จะได้รับการตรวจสอบความถูกต้องของบัตรเครดิตภายในเวลา 1 วินาที พร้อมทั้งทำการโต้ตอบหรือแสดงผลภายใน 2 วินาที ผู้ดำเนินการสามารถอ่านรายละเอียดเหล่านี้ได้อย่างชัดเจน และไม่มีวามคลุมเครือเกิดขึ้น</p>
ข้อมูลนำออก	<p>หน้าที่ของข้อมูลนำออกในการสั่งซื้อสินค้า จะมีรูปแบบในการส่งผ่านข้อความผ่านเครือข่ายไปยังแฟ้มข้อมูล และระบบย่อย ไปยังระบบบัญชี เพื่อบันทึกลงในระเบียบเหล่านี้ คือ CID, PID, price, tax, shipping cost, date, time, operator ID</p>

จากรูปที่ 5.1 หน้าจอเริ่มต้นการทำงาน เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเลือกแถบเมนู Open SRS เพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ตัวอย่างที่ 3 ผลลัพธ์การทำงานแสดงดังรูปที่ 5.28


เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Cut Word ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการตัดคำและกำกับหน้าที่คำในเอกสาร ผลลัพธ์แสดงดังรูปที่ 5.29

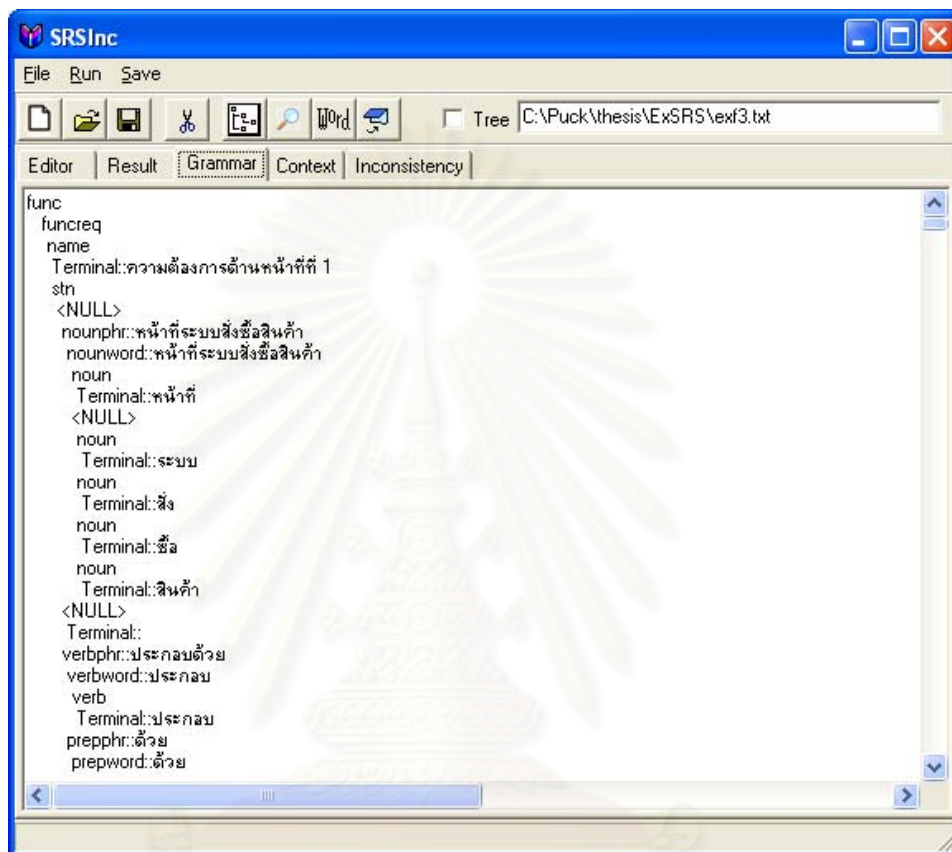


รูปที่ 5.28 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์
ด้านหน้าที่ตัวอย่างที่ 3



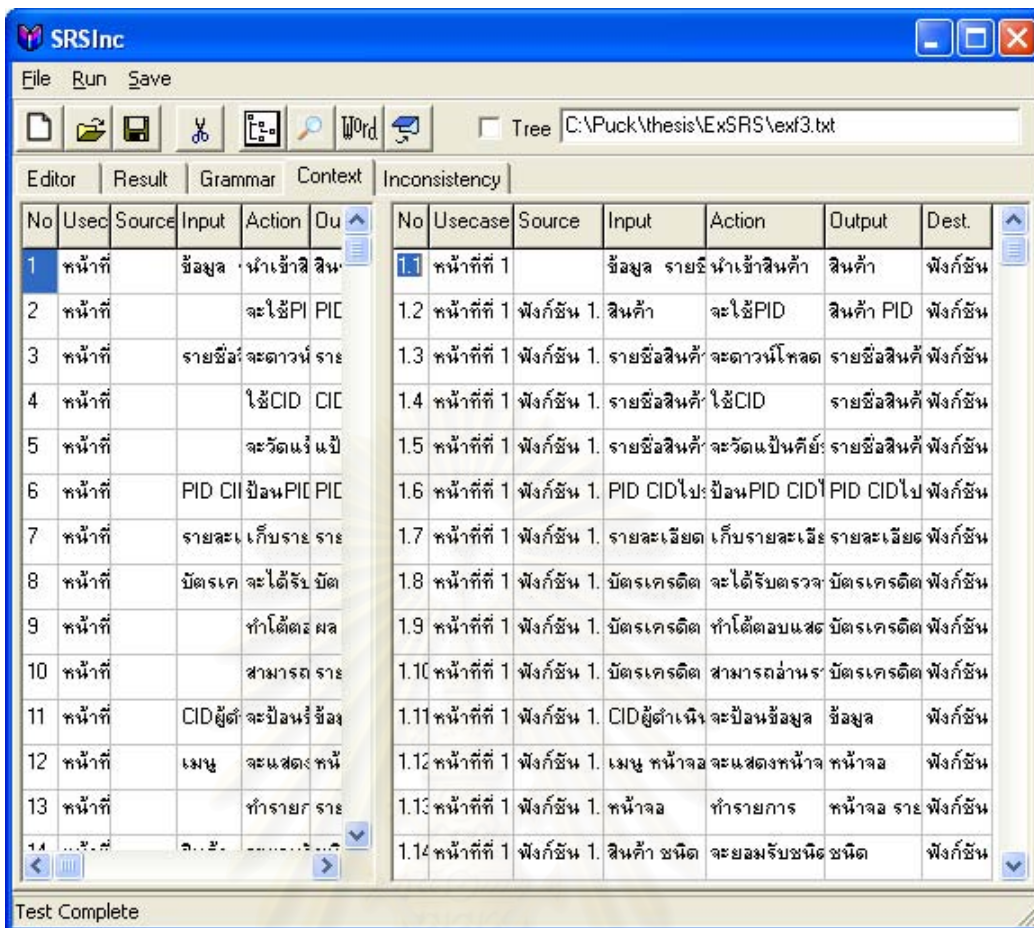
รูปที่ 5.29 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ

เลือกเครื่องมือ  ในแถบเครื่องมือ หรือเมนู Show New Structure ในแถบเมนู เพื่อให้เครื่องมือซอฟต์แวร์ทำการวิเคราะห์ประโยคด้วยไวยากรณ์ และสร้างโครงสร้างใหม่อธิบายความต้องการด้านหน้าที่ โดยการดึงส่วนสำคัญของประโยคที่อธิบายรายละเอียดความต้องการด้านหน้าที่ ผลลัพธ์การวิเคราะห์ประโยคด้วยไวยากรณ์แสดงดังรูปที่ 5.30



รูปที่ 5.30 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์

หลังจากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะทำการดึงส่วนสำคัญของประโยคที่เกี่ยวข้องกับคำหรือวลีสำคัญในการอธิบายความต้องการด้านหน้าที่ เพื่อสร้างโครงสร้างใหม่ที่ใช้อธิบายแทนความต้องการด้านหน้าที่ที่เดิมเป็นข้อความ ผลลัพธ์การทำงานของเครื่องมือซอฟต์แวร์แสดงดังรูปที่ 5.31 จะเห็นได้ว่า โครงสร้างใหม่ที่ได้เป็นโครงสร้างที่ไม่ถูกต้อง ทั้งข้อมูลนำเข้า กระบวนการทำงาน และข้อมูลนำออกไม่ตรงตามการทำงานจริงของความต้องการด้านหน้าที่ เนื่องมาจากการดึงส่วนสำคัญของความต้องการด้านหน้าที่มาสร้างโครงสร้างใหม่นั้น ไม่สามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งมีผลมาจากรูปแบบการเขียนเอกสารที่กล่าวไปแล้วข้างต้น



รูปที่ 5.31 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

เนื่องจากโครงสร้างใหม่ที่ใช้อธิบายความต้องการด้านหน้าที่เป็นโครงสร้างที่ไม่ถูกต้อง ดังนั้นการประมวลผลของเครื่องมือซอฟต์แวร์ในขั้นถัดไป ซึ่งก็คือการตรวจสอบความไม่ตรงกันนั้น ไม่สามารถทำการตรวจสอบได้ ผลลัพธ์ที่ได้จากการตรวจสอบจะไม่ถูกต้อง เพราะว่าข้อมูลที่นำไปตรวจสอบ ตัวอย่างเช่น ข้อมูลนำเข้าของฟังก์ชันการทำงาน ข้อมูลนำออกของฟังก์ชันการทำงาน กระบวนการของฟังก์ชันการทำงาน ไม่ใช่ข้อมูลที่ถูกต้อง

5.4 ผลการประเมิน

จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ทดสอบกับเครื่องมือซอฟต์แวร์ เห็นได้ว่า มีเพียงเอกสารที่มีรูปแบบการเขียนที่ง่าย โครงสร้างประโยคที่ตรงตามหลักของไวยากรณ์ เท่านั้น ที่สามารถตรวจสอบความไม่ตรงกันโดยเครื่องมือซอฟต์แวร์นี้ แสดงว่า เครื่องมือซอฟต์แวร์นี้ เหมาะกับเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่มีการเขียนโครงสร้างประโยคที่ตรงตามหลักของไวยากรณ์ ทำความเข้าใจได้ง่าย มีข้อมูลที่เกี่ยวข้องถึงส่วนสำคัญของความต้องการด้านหน้าที่ไว้อย่างครบถ้วน เครื่องมือซอฟต์แวร์นี้ไม่เหมาะกับเอกสารที่มีรูปแบบการเขียนที่ซับซ้อน มีการใช้ประโยคฟุ่มเฟือย มีการเขียนบรรยายความรายละเอียดของความต้องการด้านหน้าที่อย่างละเอียด โดยที่ประโยคเหล่านั้นไม่ใช่ส่วนสำคัญของความต้องการด้านหน้าที่

สำหรับเอกสารที่สามารถทำการตรวจสอบความไม่ตรงกันในเอกสารได้ด้วยเครื่องมือซอฟต์แวร์นั้น ผลลัพธ์การทำงานที่ได้ แสดงให้เห็นว่าเครื่องมือซอฟต์แวร์สามารถทำงานได้ทั้งในส่วนการตัดคำ การกำกับหน้าที่คำ การวิเคราะห์ประโยคด้วยไวยากรณ์ และการดึงข้อมูลที่สำคัญของความต้องการด้านหน้าที่ได้อย่างถูกต้อง โครงสร้างใหม่ที่สร้างขึ้นเพื่ออธิบายความต้องการด้านหน้าที่ที่ได้นั้น มีความถูกต้อง ครบถ้วน แต่สำหรับเอกสารที่มีความกำกวม โครงสร้างประโยคมีความซับซ้อน เครื่องมือซอฟต์แวร์นี้จะไม่สามารถทำงานได้ เนื่องจากโครงสร้างใหม่ที่สร้างขึ้นไม่ถูกต้อง ทำให้ไม่สามารถนำโครงสร้างนั้นไปตรวจสอบหาความไม่ตรงกันในเอกสารได้

จากการแก้ไขตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ให้เกิดความไม่ตรงกันขึ้นนั้น แสดงให้เห็นว่าเครื่องมือสามารถตรวจสอบพบความไม่ตรงกันในเอกสารตามที่ระบุไว้ในงานวิจัยได้อย่างถูกต้องทุกประเภทความไม่ตรงกัน

สรุปผลการวิจัย ปัญหา และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้ทำการออกแบบขั้นตอนวิธี ในการตรวจสอบความไม่ต้องการในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย เฉพาะเอกสารในส่วนความต้องการด้านหน้าที่ การทำงานของซอฟต์แวร์ ทำให้ผู้พัฒนาซอฟต์แวร์สามารถตรวจสอบความไม่ต้องการในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยได้ง่าย และปรับปรุงเอกสารให้มีความถูกต้อง เหมาะสมต่อการนำไปวิเคราะห์ต่อไป ในวิทยานิพนธ์นี้ใช้ตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย ที่มาจากแหล่งข้อมูลต่างกัน 5 แหล่ง นำมาประยุกต์ให้ตรงกับมาตรฐานเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยที่เลือกใช้ [2] และนำมาดัดแปลงให้เกิดความไม่ต้องการขึ้นในเอกสาร เพื่อเป็นตัวอย่างข้อมูลสำหรับการตรวจสอบความไม่ต้องการ

การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ต้องการ แบ่งเป็น 2 ส่วนใหญ่ ได้แก่ ส่วนการประมวลผลภาษาธรรมชาติที่ใช้ในการเขียนเอกสาร และส่วนการตรวจสอบความไม่ต้องการในเอกสาร ในส่วนการประมวลผลภาษาธรรมชาติ ประกอบด้วย การตัดคำ การกำกับหน้าที่คำ และการวิเคราะห์ประโยค เพื่อสร้างเป็นรูปแบบโครงสร้างใหม่ที่ใช้ในการตรวจสอบความไม่ต้องการ รูปแบบโครงสร้างใหม่ที่ได้จะแสดงให้เห็นส่วนสำคัญของความต้องการด้านหน้าที่ของซอฟต์แวร์ คือ ยูสเคส (ผู้รับผิดชอบฟังก์ชัน) แหล่งที่มาของข้อมูล ข้อมูลนำเข้า ฟังก์ชันการทำงาน ข้อมูล นำออก และแหล่งปลายทางของข้อมูล ในส่วนการตรวจสอบความไม่ต้องการในเอกสาร เป็นการหาขั้นตอนวิธีในการตรวจสอบความไม่ต้องการ จากรูปแบบโครงสร้างที่ได้จากการประมวลผลใน ส่วนแรก โดยมีการพัฒนาเครื่องมือซอฟต์แวร์เพื่อให้สามารถทำการตรวจสอบความไม่ต้องการในเอกสารได้โดยอัตโนมัติ

ขั้นตอนวิธีตรวจสอบความไม่ต้องการที่ได้จากการวิจัย สามารถตรวจสอบความไม่ต้องการ 3 ประเภทด้วยกัน คือ ความไม่ต้องการในส่วนของข้อมูลในฐานข้อมูล ความไม่ต้องการในความต้องการด้านหน้าที่เดียวกัน ซึ่งประกอบด้วย ฟังก์ชันการทำงานที่ซ้ำซ้อนในความต้องการด้านหน้าที่เดียวกัน ข้อมูลนำเข้าและข้อมูลนำออกของแต่ละฟังก์ชันการทำงานไม่ต้องการ ข้อมูลนำเข้าและข้อมูลนำออกของฟังก์ชันการทำงานที่ทำงานต่อเนื่องกันไม่ต้องการ และความไม่ต้องการระหว่างความต้องการด้านหน้าที่ที่ต่างหน้าที่กัน ซึ่งประกอบด้วย ความต้องการด้านหน้าที่ที่ซ้ำซ้อนกัน คำที่หมายถึงกลุ่มข้อมูลเดียวกันในฐานข้อมูล แต่ประกอบด้วยข้อมูลย่อยไม่ตรงกัน ในความต้องการด้านหน้าที่ที่ต่างกัน

ขั้นตอนวิธีการตรวจสอบความไม่ต้อกันที่ออกแบบและพัฒนาเครื่องมือซอฟต์แวร์ขึ้นมา นั้น มีการทดสอบประสิทธิภาพโดยนำเครื่องมือซอฟต์แวร์ที่พัฒนาขึ้น มาตรวจสอบความไม่ต้อกัน จากตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย สรุปผลได้ว่าขั้นตอนวิธีการตรวจสอบความไม่ต้อกันที่ออกแบบสามารถตรวจสอบพบความไม่ต้อกันทั้ง 3 ประเภทที่กล่าวมาแล้วข้างต้นได้ถูกต้อง โดยประสิทธิภาพของขั้นตอนวิธีการตรวจสอบความไม่ต้อกันจะขึ้นกับเงื่อนไขที่สำคัญ คือ ลักษณะการเขียนเอกสาร โครงสร้างประโยค คำศัพท์ที่ใช้ เนื่องจากสิ่งเหล่านี้มีผลกับการทำงานในส่วนการประมวลผลภาษาธรรมชาติ ซึ่งจะวิเคราะห์หาคำสำคัญ เพื่อนำมาสร้างเป็นรูปแบบโครงสร้างใหม่ หากรูปแบบโครงสร้างใหม่ที่ได้จากขั้นตอนการประมวลผลภาษาธรรมชาติ ไม่สามารถแสดงส่วนสำคัญของความต้องการด้านหน้าที่การทำงานของซอฟต์แวร์อย่างถูกต้องครบถ้วน จะมีผลกับการตรวจสอบความไม่ต้อกันในเอกสาร

6.2 ปัญหาและข้อจำกัดของงานวิจัย

1. การหาตัวอย่างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย หาได้ยาก เนื่องจากผู้พัฒนาซอฟต์แวร์ในประเทศไทย มักจะไม่ทำเอกสารที่เป็นมาตรฐานของข้อกำหนดความต้องการซอฟต์แวร์ออกมา และบางบริษัทที่เป็นผู้พัฒนาซอฟต์แวร์ ไม่สามารถเปิดเผยเอกสารได้ เพราะถือเป็นลิขสิทธิ์ของบริษัท
2. เอกสารที่นำมาตรวจสอบความไม่ต้อกันโดยใช้เครื่องมือซอฟต์แวร์จากงานวิจัยนี้ ต้องเป็นเอกสารที่ตรงตามมาตรฐานที่เลือกไว้เท่านั้น ดังนั้นจึงต้องมีการเตรียมเอกสารให้เหมาะสมกับเครื่องมือซอฟต์แวร์นี้ด้วยมือ และรูปแบบการเขียนเอกสาร ลักษณะโครงสร้างประโยคที่ใช้ต้องไม่กำกวม สามารถแสดงถึงส่วนสำคัญของความต้องการด้านหน้าที่ได้อย่างชัดเจน ถ้ารูปแบบการเขียนเอกสารมีความกำกวม ไม่ชัดเจน หรือใช้คำพึมพำเพี้ยนไป จะมีผลต่อประสิทธิภาพของการประมวลผลภาษาธรรมชาติ และกระทบไปยังประสิทธิภาพของการตรวจสอบความไม่ต้อกัน
3. โครงสร้างใหม่ที่สร้างขึ้นจากขั้นตอนการประมวลผลภาษาธรรมชาติ เพื่อใช้อธิบายความต้องการด้านหน้าที่นั้น ลักษณะฟังก์ชันการทำงานของซอฟต์แวร์ในโครงสร้างใหม่จะมีลักษณะการทำงานที่ต่อเนื่องกันแบบหนึ่งต่อหนึ่ง (One To One) เท่านั้น คือแต่ละฟังก์ชันการทำงาน จะมีการนำเข้าข้อมูลจากแหล่งที่มาของข้อมูลเพียงแหล่งเดียว และมีการส่งข้อมูลนำออกไปยังแหล่งปลายทางของข้อมูลเพียงแหล่งเดียว โดยถ้าความต้องการด้านหน้าที่นั้นมีหลายฟังก์ชันการทำงาน ฟังก์ชันการทำงานเหล่านั้น จะต้องเป็นการทำงานที่ต่อเนื่องกัน

4. การสร้างฐานข้อมูลในส่วนของฐานข้อมูลคำศัพท์ ทั้งที่เป็นฐานข้อมูลธรรมดาและฐานข้อมูลที่เป็นฐานความรู้ที่ยังไม่สมบูรณ์ มีเพียงคำศัพท์บางกลุ่มเท่านั้น เนื่องจากระยะเวลาและตัวอย่างข้อมูลไม่เพียงพอ ในการสร้างฐานความรู้ที่ตรงต่อความต้องการ สำหรับช่วยในการทำงานได้อย่างมีประสิทธิภาพ ดังนั้นจึงยังมีปัญหาในขั้นตอนการประมวลผลภาษาธรรมชาติที่เกี่ยวกับคำเฉพาะ คำภาษาต่างประเทศ คำนามที่เป็นกลุ่มคำ และคำกริยาที่เป็นกลุ่มคำ
5. การออกแบบขั้นตอนวิธีในการตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยในส่วนของความต้องการด้านหน้าที่ของงานวิจัยนี้ แบ่งการออกแบบขั้นตอนวิธีเป็น 2 ส่วน ดังนั้นทำให้ประสิทธิภาพของการตรวจสอบความไม่ตรงกันในส่วนหลังขึ้นกับประสิทธิภาพของขั้นตอนการประมวลผลภาษาธรรมชาติใน ส่วนแรก ที่สร้างโครงสร้างใหม่ในการอธิบายความต้องการด้านหน้าที่ โดยถ้าโครงสร้างใหม่ที่ได้ไม่ถูกต้อง ครบถ้วน ผลลัพธ์ของการตรวจสอบความไม่ตรงกันก็จะไม่ถูกต้องด้วย
6. การตรวจสอบความไม่ตรงกันในงานวิจัยนี้ จะตรวจสอบเฉพาะในส่วนความต้องการด้านหน้าที่ของซอฟต์แวร์ และตรวจสอบได้เฉพาะความไม่ตรงกันที่ระบุไว้ในงานวิจัยเท่านั้น

6.3 ข้อเสนอแนะ

1. เนื่องจากประสิทธิภาพของขั้นตอนวิธีในการตรวจสอบความไม่ตรงกัน ขึ้นกับผลลัพธ์ที่ได้จากการประมวลผลภาษาธรรมชาติเป็นหลัก ดังนั้นควรมีการพัฒนาประสิทธิภาพการทำงานในการประมวลผลภาษาธรรมชาติให้ดีขึ้น ทำให้สามารถสร้างโครงสร้างใหม่ในการอธิบายความต้องการด้านหน้าที่ที่มีความถูกต้องสมบูรณ์ขึ้น การพัฒนาประสิทธิภาพของขั้นตอนการประมวลผลภาษาธรรมชาติสามารถทำได้โดย
 - สร้างข้อมูลคำศัพท์ที่ใช้ในการเขียนเอกสารข้อกำหนดความต้องการที่มีข้อมูลทั้งด้านไวยากรณ์และด้านความหมายของคำศัพท์ เพื่อเป็นฐานความรู้ที่มีความสมบูรณ์ ถูกต้องมากขึ้น
 - พัฒนาขั้นตอนวิธีการตัดคำและการกำกับหน้าที่คำให้มีประสิทธิภาพมากขึ้น สามารถจัดการกับค่านาม หรือคำกริยาที่เป็นกลุ่มคำได้อย่างเหมาะสม

2. เนื่องจากไม่มีฐานความรู้ด้านความหมายของคำศัพท์ที่สมบูรณ์ ทำให้การวิเคราะห์ ส่วนของประโยคและหาคำสำคัญนั้น ใช้การวิเคราะห์ด้านไวยากรณ์เป็นหลัก ยังขาด ความสำคัญในการวิเคราะห์ด้านความหมายของคำศัพท์ ดังนั้นการพัฒนาในอนาคต ควรให้ความสำคัญกับความหมายของคำศัพท์มากขึ้น จะสามารถแสดงส่วนสำคัญของ ความต้องการด้านหน้าที่การทำงานของซอฟต์แวร์ได้ถูกต้องมากขึ้น
3. เนื่องจากขั้นตอนวิธีในการตรวจสอบความไม่ตรงกัน จะตรวจสอบได้เฉพาะความไม่ ตรงกันที่ระบุไว้ในงานวิจัยเท่านั้น ดังนั้นการพัฒนาต่อไปในอนาคตควรตรวจสอบ ความไม่ตรงกันในด้านอื่น ๆ เพิ่มขึ้น เช่น เงื่อนไข (Constrain) ที่มีผลต่อการ ทำงานของซอฟต์แวร์
4. เนื่องจากงานวิจัยนี้ศึกษาเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทยจาก 5 แหล่งข้อมูลเท่านั้น ทำให้ไม่ครอบคลุมเอกสารข้อกำหนดความต้องการซอฟต์แวร์ทุก ลักษณะการทำงาน (Business Domain) ดังนั้นการพัฒนาต่อไปในอนาคตควร พิจารณาเอกสารข้อกำหนดความต้องการซอฟต์แวร์ในลักษณะการทำงานอื่นเพิ่มเติม

รายการอ้างอิง

1. Sugimoto, H. and Ohnishi, A. A Detecting and Interpreting Method of the Inconsistency of Software Requirements Specification. IEEE (1999) : 208-215.
2. Assawamekin, N. Development a Guideline for Preparing Software Requirement Specification Document. Degree of Master Computer Science in Department of Computer Engineering Faculty of Engineering Chulalongkorn University, 1999.
3. Osborne, M. and MacNish, C. K. Processing Natural Language Software Requirement Specifications. IEEE, Proceeding of ICRE'96 (1996) : 229-236.
4. Kim, H., Ko, Y. , et al. Informal Requirements Analysis Supporting System for Human Engineer. IEEE (1999) : 1013-1018.
5. Kotonya, G. and Sommerville, I. Requirements Engineering (Processes and Techniques). : John Willey & Sons, 1997.
6. Kim, D. and Chong, K. A Method of checking Errors and Consistency in the Process of Object-Oriented Analysis. IEEE (1996) : 208-216.
7. Charoenpornawat, P. Feature-based Thai Word Segmentation. Degree of Master Computer Engineering in Department of Computer Engineering Faculty of Engineering Chulalongkorn University, 1998.
8. Hoppenbrouwers, J., et al. The Grammalizer: A CASE Tool Based on Textual Analysis. the CAISE97 Conference (November 1996) : 1-13.
9. Achour, C. B. and Rolland, C. Introducing Genericity and Modularity of Textual Scenario Interpretation in the Context of Requirements Engineering. ESPRIT Reactive Long Term Research CREWS (Cooperative Requirements Engineering With Scenarios (1997).
10. Finkelstein, C.W. A., et al. Inconsistency Handling in Multiperspective Specifications. IEEE Transaction vol. 20 no.8 (1994) : 569-577.
11. Characteristics of Good Software Requirements Specification [online]. Available from: <http://www.cs.city.ac.uk/homes/spa/project/section3.htm>. City University, UK. , 1998.
12. IEEE SA Standard Board. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998 (1998).

13. Dennis, A. and Wixom, Haley B. Systems Analysis and Design. : John Wiley & Son, 1999.
14. Nuseibeh, B., Easterbook, et al. S. Making Inconsistency Respectable in Software Development. IEEE Computer Society Press vol.33 no.4 (April 2000) : 24-29.
15. Sukhahuta, R. and Smith, D. Information Extraction Strategies for Thai Documents. International Journal of Computer Processing of Oriental Languages (IJCPOL) vol. 4 no.2 (2001).
16. Sornlertlamvanich, V. and Charoenporn, T. ORCHID: Thai Part-Of-Speech Tagged Corpus. NECTEC (1998).
17. Information Research and Development Division [online]. Available from: <http://www.links.nectec.or.th/>. National Electronics and Computer Technology Center (NECTEC), 2002.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.

แผนแบบเอกสารข้อกำหนดความต้องการซอฟต์แวร์ภาษาไทย [2]

1. บทนำ (Introduction)
 - 1.1 จุดประสงค์ (Purpose)
 - 1.2 ขอบเขต (Scope)
 - 1.3 คำจำกัดความ (Definitions)
 - 1.4 รัศพจน์ (Acronyms)
 - 1.5 คำย่อ (Abbreviations)
 - 1.6 เอกสารอ้างอิง (References)
 - 1.7 ภาพรวม (Overview)
2. รายละเอียดโดยรวม (Overall description)
 - 2.1 แนวคิดหรือภาพโดยรวมของซอฟต์แวร์ (Product perspective)
 - 2.1.1 การต่อประสานกับผู้ใช้ (User interface)
 - 2.1.2 การต่อประสานกับฮาร์ดแวร์ (Hardware interfaces)
 - 2.1.3 การต่อประสานกับซอฟต์แวร์ (Software interfaces)
 - 2.1.4 การต่อประสานกับระบบสื่อสาร (Communications interfaces)
 - 2.1.5 ข้อจำกัดของหน่วยความจำ (Memory constraints)
 - 2.1.6 การดำเนินการ (Operations)
 - 2.1.7 ความต้องการด้านการติดตั้งซอฟต์แวร์ (Site adaptation requirements)
 - 2.2 หน้าที่ของซอฟต์แวร์ (Product functions)
 - 2.3 ลักษณะของผู้ใช้ (User characteristics)
 - 2.4 ข้อจำกัด (Constraints)
 - 2.5 ข้อสันนิษฐานและการขึ้นต่อกัน (Assumptions and dependencies)
 - 2.6 ข้อจำกัดในการเตรียมกระบวนการในการพิจารณาซอฟต์แวร์ (Qualification provisions)
 - 2.7 ความสามารถในการติดตามความต้องการ (Requirements traceability)
 - 2.8 การแบ่งสับปันส่วนความต้องการ (Apportioning of requirements)

3. ความต้องการที่เฉพาะเจาะจง (Specific requirements)
 - 3.1 การต่อประสานภายนอก (External interfaces)
 - 3.2 ความต้องการด้านหน้าที่ (Functional requirements)
 - 3.2.1 ความต้องการด้านหน้าที่ที่ 1 (Functional requirement 1)
 - 3.2.1.1 บทนำ (Introduction)
 - 3.2.1.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.1.3 กระบวนการ (Processing)
 - 3.2.1.4 ข้อมูลนำออก (Outputs)
 - 3.2.2 ความต้องการด้านหน้าที่ที่ 2 (Functional requirement 2)
 -
 -
 -
 - 3.2.n ความต้องการด้านหน้าที่ที่ n (Functional requirement n)
 - 3.2.n.1 บทนำ (Introduction)
 - 3.2.n.2 ข้อมูลนำเข้า (Inputs)
 - 3.2.n.3 กระบวนการ (Processing)
 - 3.2.n.4 ข้อมูลนำออก (Outputs)
 - 3.3 ความต้องการด้านประสิทธิภาพ (Performance requirements)
 - 3.4 ข้อจำกัดในการออกแบบ (Design constraints)
 - 3.4.1 มาตรฐานที่มีใช้ในองค์กร (Standards compliance)
 - 3.4.2 ข้อจำกัดในการใช้งานซอฟต์แวร์ (Implementation constraints)
 - 3.5 ลักษณะประจำ (Attributes)
 - 3.5.1 ความเชื่อถือได้ (Reliability)
 - 3.5.2 สภาพพร้อมใช้งาน (Availability)
 - 3.5.3 ความมั่นคง (Security)
 - 3.5.4 ความสามารถในการบำรุงรักษา (Maintainability)
 - 3.5.5 ความสามารถในการใช้ได้หลายระบบ (Portability)
 - 3.6 ความต้องการอื่น ๆ (Other requirements)
 - 3.6.1 ความต้องการด้านฐานข้อมูลเชิงตรรกะ (Logical database requirements)
 - 3.6.2 ความต้องการด้านการดำเนินการ (Operational requirements)

- 3.6.3 ความต้องการด้านการใช้ทรัพยากรฮาร์ดแวร์คอมพิวเตอร์ให้เกิดประโยชน์สูงสุด (Computer hardware resource utilization requirements)
 - 3.6.4 ความต้องการด้านระบบสื่อสารคอมพิวเตอร์ (Computer communication requirements)
 - 3.6.5 ปัจจัยที่มีผลต่อคุณภาพของซอฟต์แวร์ (Software quality factors)
 - 3.6.6 ความต้องการด้านการเตรียมขั้นตอนต่าง ๆ ที่ใช้ในการพัฒนาซอฟต์แวร์ (Logistics-related requirements)
 - 3.6.7 ความต้องการด้านความปลอดภัย (Safety requirements)
 - 3.6.8 ความต้องการด้านการทดสอบ (Verification requirements)
 - 3.6.9 ความต้องการด้านการยอมรับ (Acceptance requirements)
 - 3.6.10 ความต้องการด้านเอกสาร (Documentation requirements)
 - 3.6.11 ความต้องการด้านบรรจุภัณฑ์ซอฟต์แวร์ (Packaging requirements)
 - 3.6.12 ความต้องการด้านสภาพแวดล้อม (Environment requirements)
 - 3.6.13 ความต้องการด้านการฝึกอบรมบุคคลที่เกี่ยวข้อง (Training-related requirements)
4. ข้อมูลสนับสนุน (Supporting information)
- 4.1 บันทึก (Notes)
 - 4.2 ภาคผนวก (Appendixes)
 - 4.3 ดัชนี (Index)

ภาคผนวก ข.

ยูสเคส [13]

ID number : <u>2</u>			
Use case name : <u>รับข้อมูลต่าง ๆ ของลูกค้า</u>			
Short description : <u>หน้าที่ในส่วนนี้ เป็นการรับข้อมูลลูกค้าใหม่ สร้างรหัสลูกค้า และเก็บข้อมูลลูกค้าลงในฐานข้อมูล</u>			
Trigger : <u>ผู้ใช้ใส่ข้อมูลชื่อและที่อยู่ของลูกค้าใหม่</u>			
Type :		(External)	Temporal
Major Inputs		Major Outputs	
Description	Source	Description	Destination
<u>ชื่อลูกค้า</u>	<u>ผู้ใช้</u>	<u>รหัสลูกค้า</u>	<u>ฐานข้อมูล</u>
<u>ที่อยู่ลูกค้า</u>	<u>ผู้ใช้</u>	<u>ชื่อลูกค้า</u>	<u>ฐานข้อมูล</u>
_____	_____	<u>ที่อยู่ลูกค้า</u>	<u>ฐานข้อมูล</u>
Major Steps Performed		Information for Steps	
<p>1. <u>รับข้อมูลชื่อลูกค้าและที่อยู่ลูกค้า</u></p> <p style="text-align: center;"> </p>		<p><u>ชื่อลูกค้า</u></p> <p><u>ที่อยู่ลูกค้า</u></p>	
<p>2. <u>สร้างรหัสลูกค้า</u></p> <p style="text-align: center;"> </p>		<p><u>รหัสลูกค้า</u></p>	
<p>3. <u>เก็บข้อมูลลูกค้าลงในฐานข้อมูล</u></p> <p style="text-align: center;"> </p>		<p><u>ชื่อลูกค้า</u></p> <p><u>ที่อยู่ลูกค้า</u></p> <p><u>รหัสลูกค้า</u></p>	

ภาคผนวก ค.

ตารางแสดงหน้าที่ของคำในภาษาไทย จากคลังข้อความออร์คิด

ตารางที่ ค-1 หน้าที่ของคำในภาษาไทยและตัวอย่าง

หน้าที่ของคำ	รายละเอียด	ตัวอย่าง
NPRP	Proper noun	วินโดวส์ 95, โควโรน่า, ไค้ก, พระอาทิตย์
NCNM	Cardinal number	หนึ่ง, สอง, สาม, 1, 2, 3
NONM	Ordinal number	ที่หนึ่ง, ที่สอง, ที่สาม, ที่1, ที่2, ที่3
NLBL	Label noun	1, 2, 3, 4, ก, ข, a, b
NCMN	Common noun	หนังสือ, อาหาร, อาจารย์, คน
NTTL	Title noun	ดร., พลเอก
PPRS	Personal pronoun	คุณ, เขา, ฉัน
PDMN	Demonstrative pronoun	นี้, นั่น, ที่นั่น, ที่นี่
PNTR	Interrogative pronoun	ใคร, อะไร, อย่างไร
PREL	Relative pronoun	ที่, ซึ่ง, อัน, ผู้
VACT	Active verb	ทำงาน, ร้องเพลง, กิน
VSTA	Stative verb	เห็น, รู้, คือ
VATT	Attributive verb	อ้วน, ดี, สวย
XVBM	Pre-verb auxiliary, before negator “ไม่”	เกิด, เกือบ, กำลัง
XVAM	Pre-verb auxiliary, after negator “ไม่”	ค่อย, น่า, ได้
XVMM	Pre-verb, before or after negator “ไม่”	ควร, เคย, ต้อง
XVBB	Pre-verb auxiliary, in imperative mood	กรุณา, จง, เชิญ, อย่า, ห้าม
XVAE	Post-verb auxiliary	ไป, มา, ขึ้น
DDAN	Definite determiner, after noun without classifier in between	นี้, นั่น, โน่น, ทั้งหมด
DDAC	Definite determiner, allowing classifier in between	นี้, นั่น, โน่น, อยู่นั้น
DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression	ทั้ง, อีก, เพียง

หน้าที่ ของคำ	รายละเอียด	ตัวอย่าง
DDAQ	Definite determiner, following quantitative expression	พอดี, ถ้วย
DIAC	Indefinite determiner, following noun; allowing classifier in between	ไหน, อื่น, ต่าง ๆ
DIBQ	Indefinite determiner, following quantitative expression	บาง, ประมาณ, เกือบ
DIAQ	Indefinite determiner, between noun and classifier or preceding quantitative expression	กว่า, เศษ
DCNM	Determiner, cardinal number expression	หนึ่งคน, เลือ 2 ตัว
DONM	Determiner, ordinal number expression	ที่หนึ่ง, ที่สอง, ที่สุดท้าย
ADVN	Adverb with normal form	เก่ง, เร็ว, ช้า, สม่่าเสมอ
ADVI	Adverb with iterative form	เร็ว ๆ, เสมอ ๆ, ช้า ๆ
ADVP	Adverb with prefixed form	โดยเร็ว
ADVS	Sentential adverb	โดยปกติ, ธรรมดา
CNIT	Unit classifier	ตัว, คน, เล่ม
CLTV	Collective classifier	คู่, กลุ่ม, ฝูง, เชง, ทาง, ด้าน, แบบ, รุ่น
CMTR	Measurement classifier	กิโลกรัม, แก้ว, ชั่วโมง
CFQC	Frequency classifier	ครั้ง, เทียว
CVBL	Verbal classifier	ม้วน, มัด
JCRG	Coordinating conjunction	และ, หรือ, แต่
JCMP	Comparative conjunction	กว่า, เหมือนกับ, เท่ากับ
JSBR	Subordinating conjunction	เพราะว่า, เนื่องจาก, ที่, แม้ว่า, ถ้า
RPRE	Preposition	จาก, ละ, ของ, ได้, บน
INT	Interjection	อ้อ, อ้อ, เออ, เอ้, อ้อ
FIXN	Nominal prefix	การทำงาน, ความสนุกสนาน
FIXV	Adverbial prefix	ความเร็ว
EAFF	Ending for affirmative sentence	จ้ะ, จ้ะ, ค่ะ, ครับ, นะ, น้า, เถอะ

หน้าที่ ของคำ	รายละเอียด	ตัวอย่าง
EITT	Ending for interrogative sentence	หรือ, เหนอ, ไหม, มั้ย
NEG	Negator	ไม่, ไม่ได้, ไม่ได้, มิ
PUNC	Punctuation	(,), “, ,, ;



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง.

ไวยากรณ์ภาษาไทย และรูปแบบเอกสารความต้องการด้านหน้าที่ของซอฟต์แวร์

```
grammar Srs // skips whitespace and comments
{
    func ::= funcreq [{funcreq}];
    funcreq ::= name intro input process output;
    name ::= "ความต้องการด้านหน้าที่" stn ;
    intro ::= "บทนำ" ["SPAC"] ["DASH"] [stn];
    input ::= "ข้อมูลนำเข้า" ["SPAC"] ["DASH"] [stn];
    process ::= "กระบวนการ" stn;
    output ::= "ข้อมูลนำออก" ["SPAC"] ["DASH"] [stn];

    stn ::= { [{"SPAC"}|"PUNC"]} (conj [(verbphr | nounphr | prepphr | classifiernode|
        relativecl)] | nounphr| verbphr| prepphr);

    nounphr ::= "FIXN" verbphr | nounword [{ conj (nounword|verbphr) }]
        [advword] [adjphr][classifiernode][relativecl][verbphr] [prepphr];
    nounword ::= noun ["VATT"] [{ [classifiernode]"COMM" noun ["VATT"] } ] ;

    verbphr ::= verbword [{conj verbword}] [nounphr] [classifiernode]
        [relativecl] [advphr] [prepphr];
    verbword ::= [helpverbword] verb [{verb}] [helpverbword] | helpverbword ;
    helpverbword ::= helpverb [{helpverb}];
    relativecl ::= "PREL" (verbphr | nounphr | prepphr | classifiernode| adjphr);

    prepphr ::= prepword [classifiernode] [nounphr] [verbphr] [adjphr]
        [{"SPAC"}|"PUNC"]} ;
    prepword ::= prep [{prep}];
```


advphr ::= advword ["JCMP" nounword] | pronounword | [("FIXV" "VATT")] ;

advword ::= adv [{adv}] ;

pronounword ::= pronoun [{pronoun}] ;

adjphr ::= "VATT" [advword] [nounword] | pronounword ;

classifiernode ::= mainclassifier [{mainclassifier}] ;

mainclassifier ::= "DDAN"|"DDAC"|"DDBQ"|"DDAQ"|"DIAC"|"DIBQ"|"DIAQ"|"DCNM"|"CNIT"|"CLTV"|"CMTR"|"CFQC"|"CVBL"|"NCNM"|"NONM"|"DONM";

conj ::= "JCRG"|"JCMP"|"JSBR";

noun ::= "NCMN"|"NTTL"|"NPRP"|"FIXN" verb [verb] ;

verb ::= "VACT"|"VSTA";

helpverb ::= "XVBM"|"XVAM"|"XVMM"|"XVBB"|"XVAE"|"NEG";

prep ::= "RPRE";

adv ::= "ADVN"|"ADVI"|"ADVP"|"ADVS" ;

pronoun ::= "PDMN"|"PNTR";

};

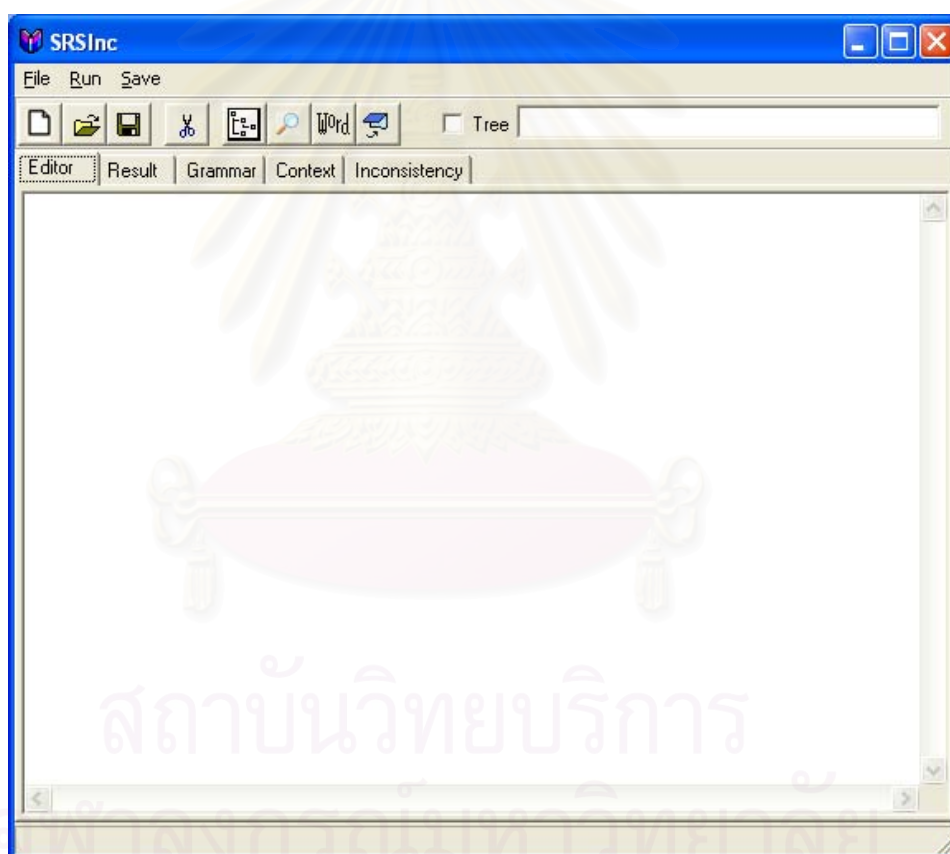
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก จ.

คู่มือประกอบการใช้เครื่องมือซอฟต์แวร์ เพื่อตรวจสอบความไม่ตรงกันใน เอกสาร

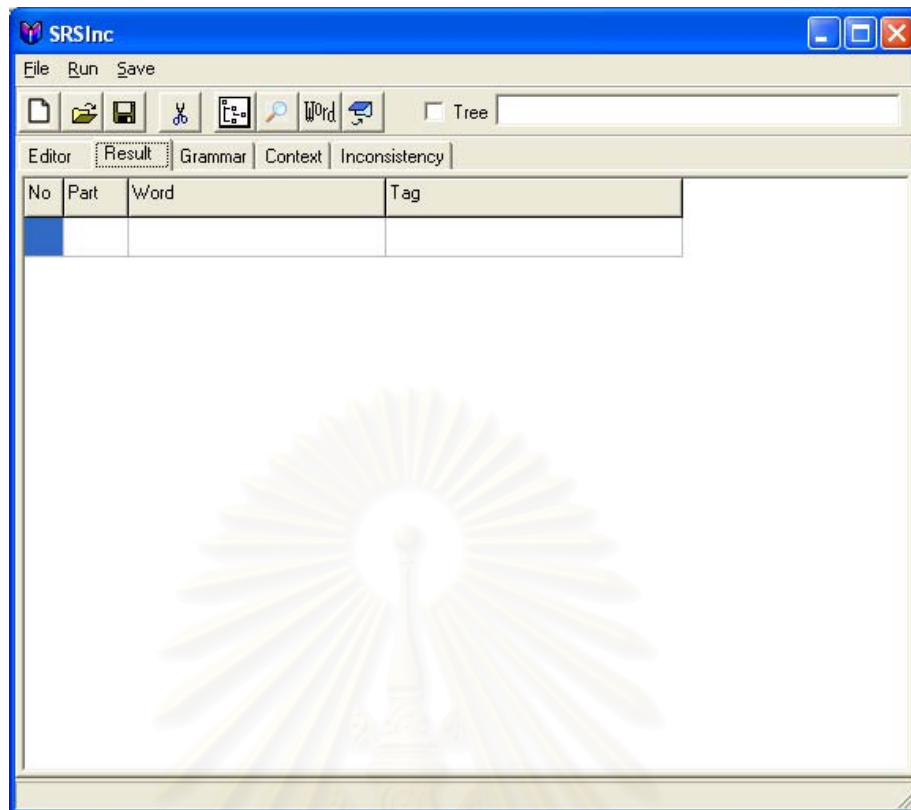
จ-1 การใช้งานหน้าจอหลักของเครื่องมือซอฟต์แวร์

เริ่มทำการรัน (Run) เครื่องมือซอฟต์แวร์ โดยเรียกโปรแกรมชื่อว่า “ChkSRS.exe” จะได้จอภาพแรกเริ่มต้นการทำงาน ดังรูปที่ จ-1 หลังจากนั้นเลือกเมนูการทำงานที่ต้องการในแถบเมนู หรือเลือกเครื่องมือที่ต้องการในแถบเครื่องมือ เพื่อให้เครื่องมือซอฟต์แวร์ทำงานตามที่ต้องการ ซึ่งจะอธิบายในส่วนถัดไป

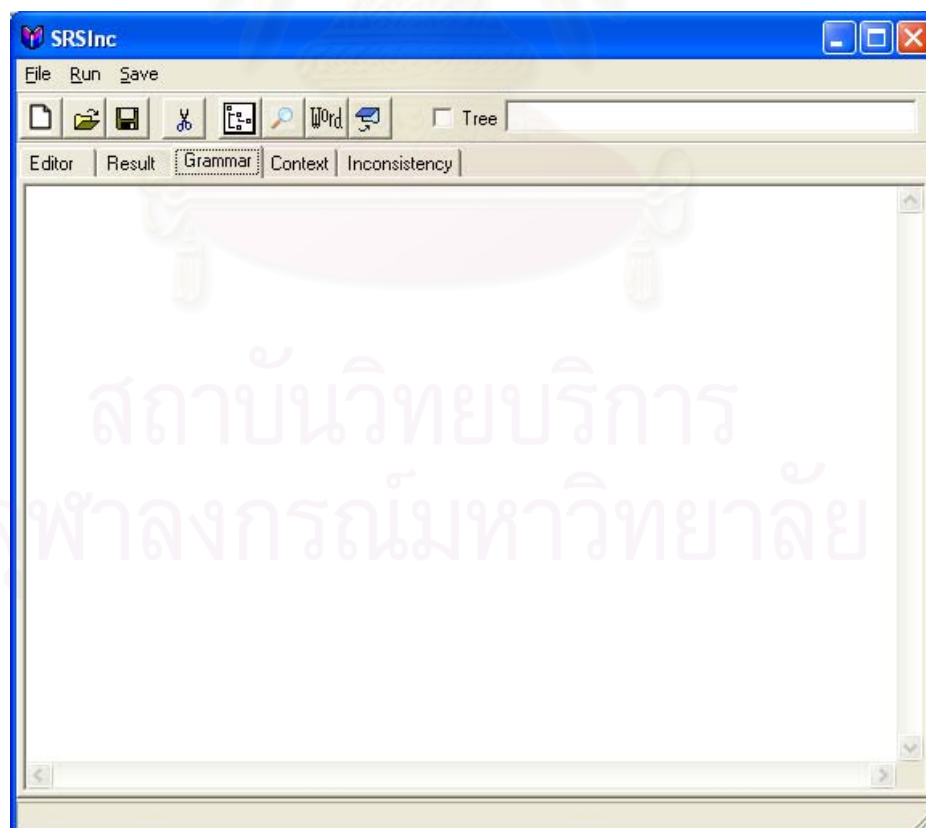


รูปที่ จ-1 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แถบเอดิเตอร์

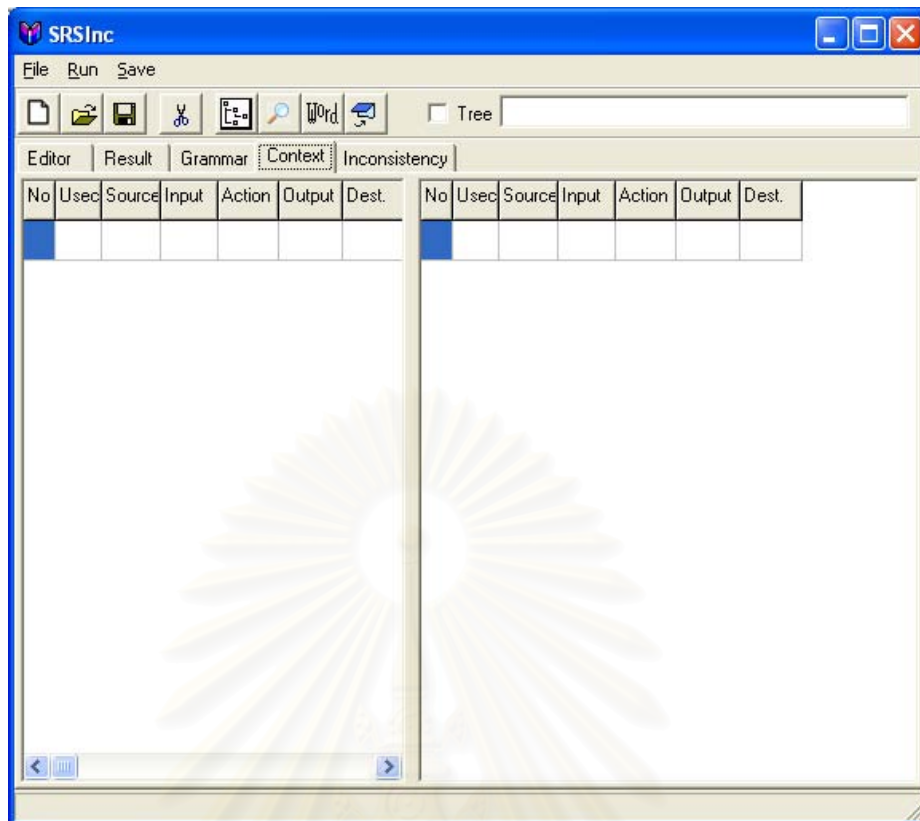
จากรูปที่ จ-1 หน้าจอหลักการทำงาน มีแถบการทำงาน 5 ส่วน สามารถเปลี่ยนแถบการทำงานได้โดยคลิกเลือกแถบการทำงานที่ต้องการ เพื่อเปลี่ยนแถบที่แสดงบนหน้าจอหลัก ผลลัพธ์จากการเปลี่ยนแถบการทำงาน แสดงดังรูปที่ จ-2 ถึงรูปที่ จ-5



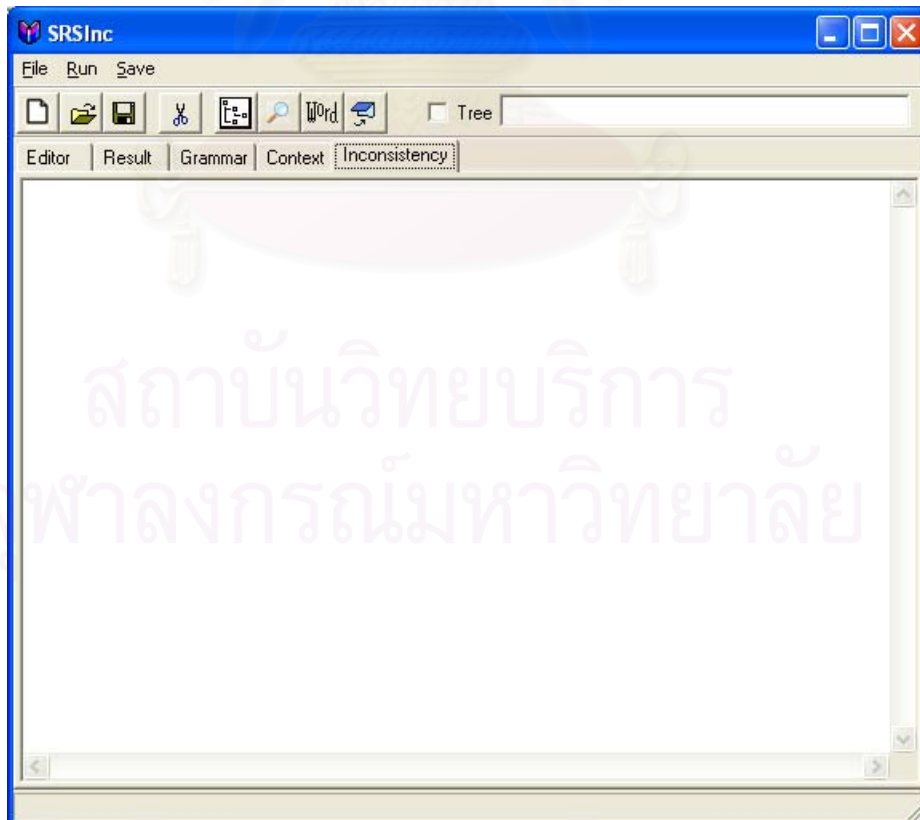
รูปที่ ๑-2 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แถบริชต์



รูปที่ ๑-3 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แถบแกรมมา



รูปที่ ๑-4 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แดบคอนเทค



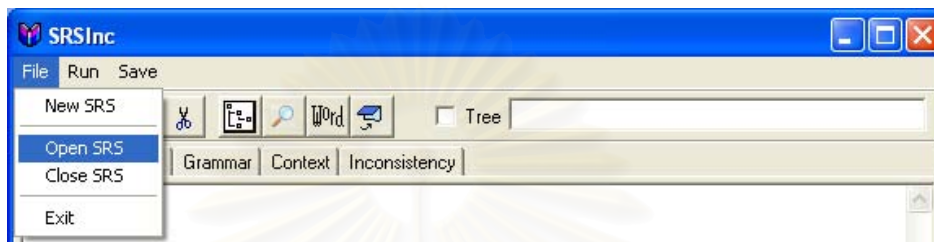
รูปที่ ๑-5 หน้าจอหลักของเครื่องมือซอฟต์แวร์ – แดบอินคอนซิสเทนซี

จากรูปที่ ๑-1 ส่วนบนของหน้าจอหลักการทำงานจะเป็นแถบเมนูและแถบเครื่องมือ สำหรับควบคุมและสั่งการให้เครื่องมือซอฟต์แวร์ทำการประมวลผล

แถบเมนู

แถบเมนูประกอบด้วย 3 เมนู คือ เมนู File เมนู Run และ เมนู Save ซึ่งมีรายละเอียดการทำงาน ดังต่อไปนี้

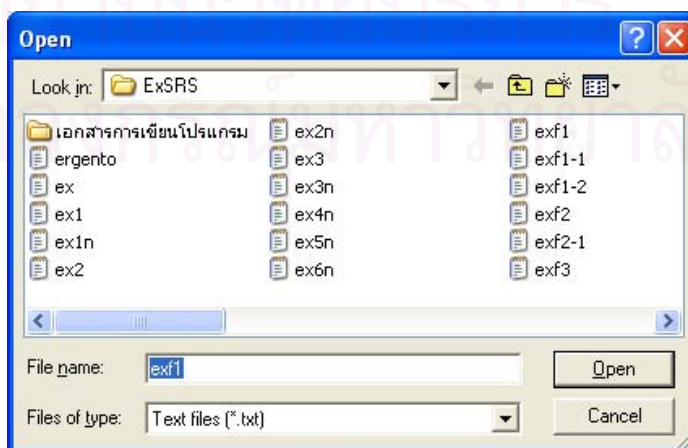
1) เมนู File ประกอบด้วยเมนูย่อยอีก 4 เมนู แสดงดังรูปที่ ๑-6



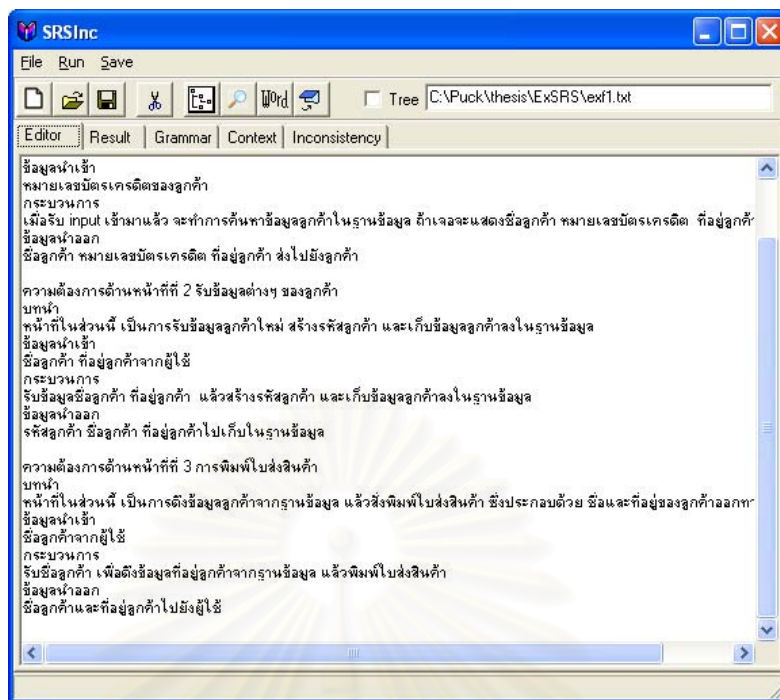
รูปที่ ๑-6 เมนู File

เมนูย่อยในเมนู File ประกอบด้วย

- เมนู New SRS เป็นเมนูเพื่อสร้างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ขึ้นมาใหม่ เมื่อเลือกเมนูนี้ จะปรากฏหน้าจอว่างในแถบเอดิเตอร์สำหรับการพิมพ์เอกสารข้อกำหนดความต้องการซอฟต์แวร์ใหม่ ผลลัพธ์การทำงานจะเหมือนหน้าจอเริ่มต้นการทำงาน แสดงดังรูปที่ ๑-1
- เมนู Open SRS เป็นเมนูเพื่อเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เมื่อเลือกเมนูนี้ จะปรากฏหน้าจอสำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ด้านหน้าที่ที่ต้องการนำมาตรวจสอบความไม่ตรงกัน เพื่อเป็นข้อมูลนำเข้าของเครื่องมือซอฟต์แวร์ แสดงดังรูปที่ ๑-7 หลังจากเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ต้องการและคลิก Open เครื่องมือซอฟต์แวร์จะเปิดเอกสารนั้นขึ้นมาทำงาน ผลลัพธ์แสดงดังรูปที่ ๑-8

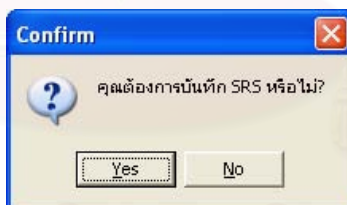


รูปที่ ๑-7 หน้าจอสำหรับเลือกเอกสารข้อกำหนดความต้องการซอฟต์แวร์ที่ต้องการ



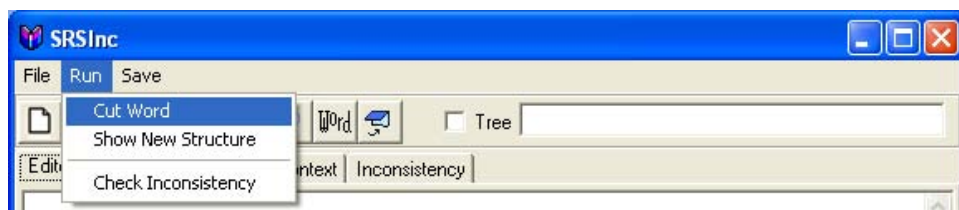
รูปที่ ๑-8 หน้าจอผลลัพธ์จากการเปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์

- เมนู Close SRS เป็นเมนูเพื่อปิดเอกสารข้อกำหนดความต้องการซอฟต์แวร์ เมื่อเลือกเมนูนี้ เครื่องมือซอฟต์แวร์จะปรากฏหน้าจอถามว่าต้องการบันทึกเอกสารที่ทำงานอยู่ก่อนปิดหรือไม่ แสดงดังรูปที่ ๑-9 ถ้าตอบ “Yes” เครื่องมือซอฟต์แวร์จะทำการบันทึกเอกสารก่อนการปิดเอกสารนั้น แต่ถ้าตอบ “No” จะไม่ทำการบันทึกเอกสาร



รูปที่ ๑-9 หน้าจอถามความต้องการในการบันทึกก่อนการปิดเอกสาร

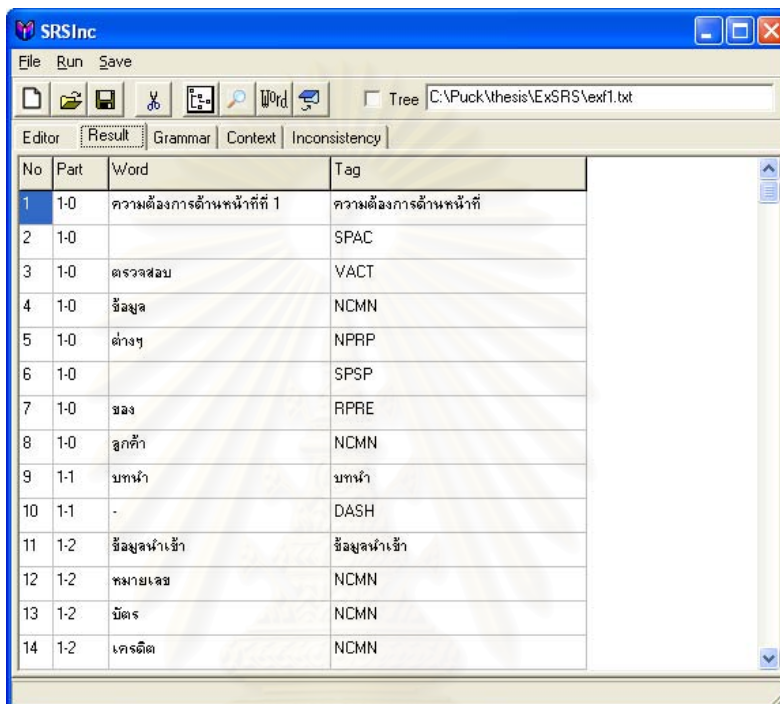
- เมนู Exit เป็นเมนูเพื่อออกจากโปรแกรม เมื่อเลือกเมนูนี้ เครื่องมือซอฟต์แวร์จะออกจากโปรแกรมที่ทำงานอยู่ โดยปิดการทำงานของเครื่องมือซอฟต์แวร์
- 2) เมนู Run ประกอบด้วยเมนูย่อยอีก 3 เมนู แสดงดังรูปที่ ๑-10



รูปที่ ๑-10 เมนู Run

เมนูย่อยในเมนู Run ประกอบด้วย

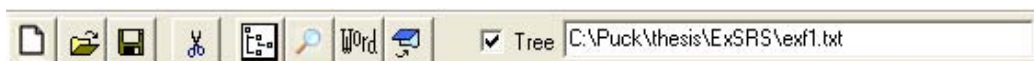
- เมนู Cut Word เป็นเมนูเพื่อการตัดคำและกำกับหน้าที่คำของคำในเอกสารข้อความ กำหนดความต้องการซอฟต์แวร์ หลังจากการเปิดเอกสารขึ้นมาทำงานหรือสร้างเอกสารใหม่ในแถบเอดิเตอร์ เมื่อเลือกเมนูนี้ ผลลัพธ์การทำงานแสดงดังรูปที่ ๑-11



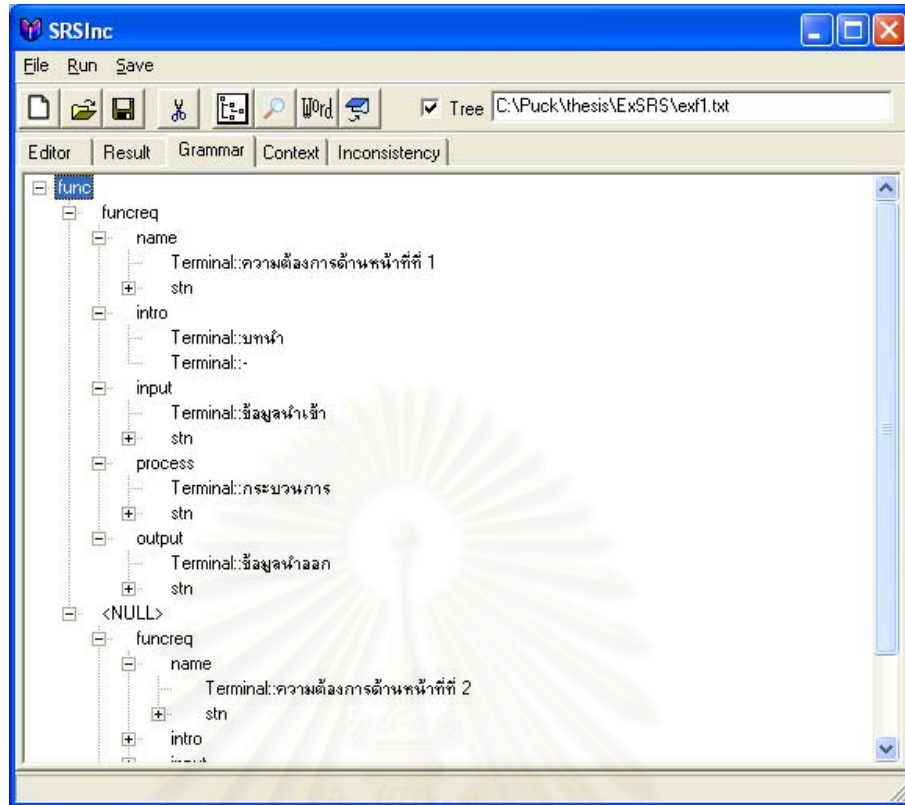
No	Part	Word	Tag
1	1-0	ความต้องการด้านหน้าที่ 1	ความต้องการด้านหน้าที่
2	1-0		SPAC
3	1-0	ตรวจสอบ	VACT
4	1-0	ข้อมูล	NCMN
5	1-0	ต่างๆ	NPRP
6	1-0		SPSP
7	1-0	ของ	RPRE
8	1-0	ลูกค้า	NCMN
9	1-1	บทนำ	บทนำ
10	1-1	-	DASH
11	1-2	ข้อมูลนำเข้า	ข้อมูลนำเข้า
12	1-2	หมายเลข	NCMN
13	1-2	บัตร	NCMN
14	1-2	เครื่องคิด	NCMN

รูปที่ ๑-11 หน้าจอผลลัพธ์จากการตัดคำและกำกับหน้าที่คำ

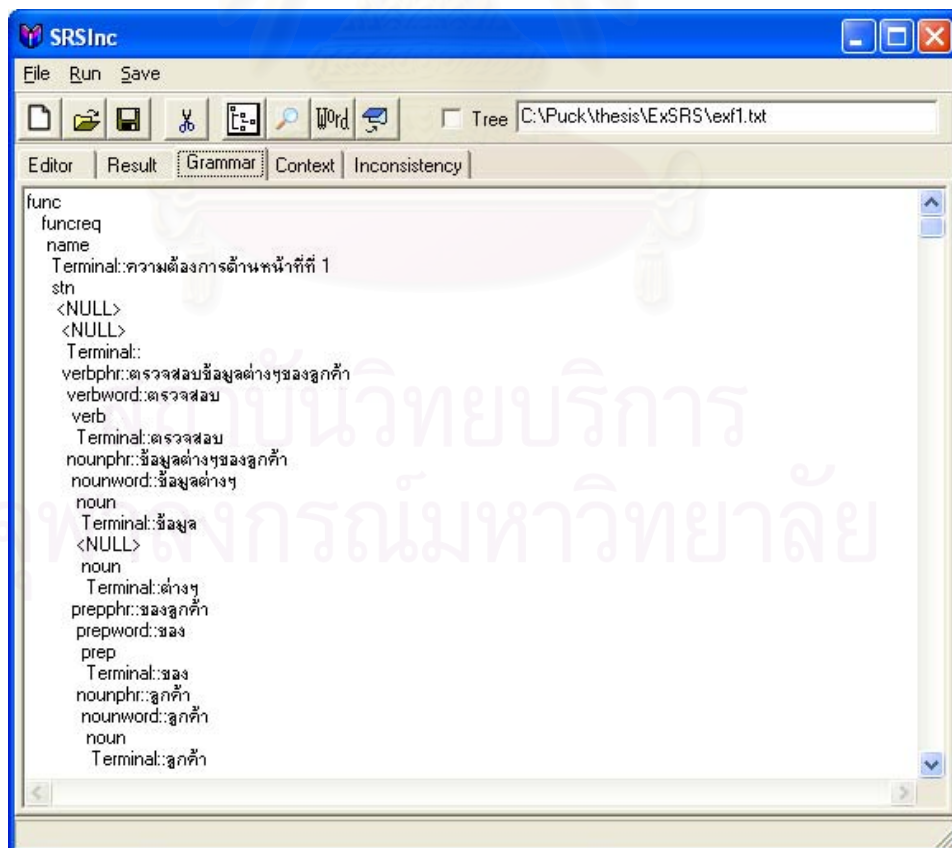
- เมนู Show New Structure เป็นเมนูเพื่อวิเคราะห์ประโยคด้วยไวยากรณ์ แล้วสร้างโครงสร้างใหม่ที่ใช้อธิบายแทนความต้องการด้านหน้าที่ หลังจากการตัดคำและกำกับหน้าที่คำ เมื่อเลือกเมนูนี้ ผลลัพธ์การทำงานของกรวิเคราะห์ประโยคด้วยไวยากรณ์จะแสดงได้ 2 ลักษณะ ขึ้นอยู่กับการเลือกลักษณะการแสดงผล โดยถ้าคลิกเลือก Tree ดังรูปที่ ๑-12 ก่อนการเลือกเมนูนี้ จะได้ผลลัพธ์การทำงานแสดงดังรูปที่ ๑-13 แต่ถ้าไม่มีการคลิกเลือก Tree จะได้ผลลัพธ์การทำงานแสดงดังรูปที่ ๑-14 หลังจากการวิเคราะห์ประโยคด้วยไวยากรณ์ จะทำการดึงส่วนสำคัญของประโยคที่เกี่ยวข้องกับคำหรือวลีสำคัญในการอธิบายความต้องการด้านหน้าที่เพื่อสร้างโครงสร้างใหม่ ผลลัพธ์การทำงานแสดงดังรูปที่ ๑-15



รูปที่ ๑-12 ออบชั่น Tree



รูปที่ จ-13 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ โครงสร้างรูปต้นไม้

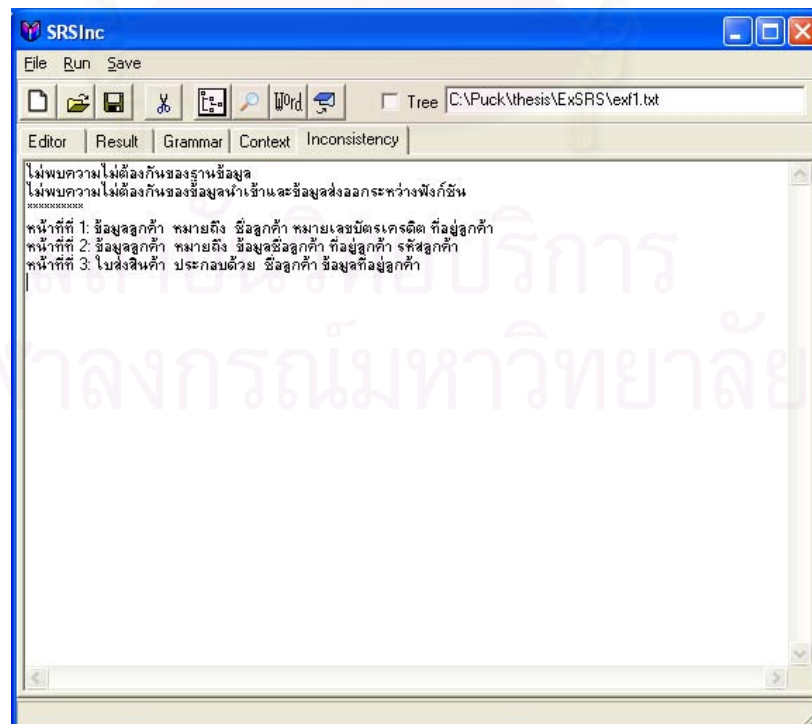


รูปที่ จ-14 หน้าจอผลลัพธ์จากการวิเคราะห์ประโยคด้วยไวยากรณ์ ลักษณะข้อความ

No	Usecas	Sour	Input	Act	Output	Dest.
1	หน้าที่ที่ 1	ลูกค้า	หมายเลข	รับหมายเลข	หมายเลข	ฟังก์ชัน 1
2	หน้าที่ที่ 1		ค้นหาข้อมูล	ฐานข้อมูล		
3	หน้าที่ที่ 1		แสดงข้อมูล	ลูกค้า		
1	หน้าที่ที่ 1	ผู้ใช้	ข้อมูล	รับข้อมูล		
2	หน้าที่ที่ 2		สร้างรหัส	ลูก		
3	หน้าที่ที่ 3		ข้อมูล	เก็บข้อมูล	ฐานข้อมูล	
1	หน้าที่ที่ 1	ผู้ใช้	ข้อมูล	รับข้อมูล		
2	หน้าที่ที่ 2		ตั้งข้อมูล	ฐานข้อมูล		
3	หน้าที่ที่ 3		ใบส่งสินค้า	พิมพ์ใบส่งสินค้า	ใบส่งสินค้า	ผู้ใช้

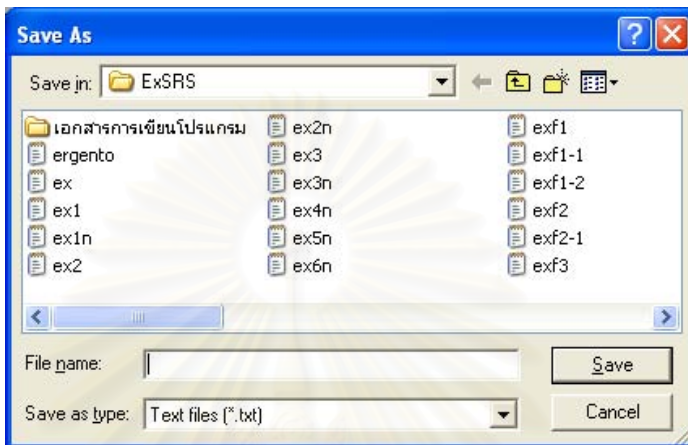
รูปที่ ๑-15 หน้าจอผลลัพธ์โครงสร้างใหม่ที่อธิบายแทนความต้องการด้านหน้าที่

- เมนู Check Inconsistency เป็นเมนูเพื่อตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความต้องการซอฟต์แวร์ โดยตรวจสอบจากโครงสร้างใหม่ที่ได้จากการประมวลผลขั้นก่อน เมื่อเลือกเมนูนี้ ผลลัพธ์การทำงานแสดงดังรูปที่ ๑-16



รูปที่ ๑-16 หน้าจอผลลัพธ์จากการตรวจสอบความไม่ตรงกัน

- 3) เมนู Save ไม่มีเมนูย่อย เมนูนี้เป็นเมนูเพื่อการบันทึกเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์ หลังจากที่เราทำการแก้ไขข้อความในเอกสารหรือสร้าง เอกสารใหม่ในแถบเอดิเตอร์ เมื่อเลือกเมนูนี้ จะปรากฏหน้าจอให้ใส่ชื่อเพิ่มข้อมูลที่ต้องการบันทึกเอกสาร แสดงดังรูปที่ ๑-17 หลังจากใส่ชื่อเพิ่มข้อมูลและคลิก Save เครื่องมือซอฟต์แวร์จะทำการบันทึกเอกสาร











รูปที่ ๑-17 หน้าจอสำหรับใส่ชื่อเพิ่มข้อมูลที่ต้องการบันทึกเอกสาร

แถบเครื่องมือ

การเลือกเครื่องมือในแถบเครื่องมือจะมีลักษณะการทำงานเหมือนการเลือกเมนูในแถบเมนูการทำงาน แต่สร้างขึ้นเพื่อให้ผู้ใช้งานเครื่องมือซอฟต์แวร์สามารถใช้เครื่องมือซอฟต์แวร์ได้สะดวกและรวดเร็วขึ้น ดังนั้นจึงจะการอธิบายในส่วนรายละเอียดไว้ แถบเครื่องมือประกอบด้วยเครื่องมือต่าง ๆ ในตารางที่ ๑-1

ตารางที่ ๑-1 หน้าทีการทำงานของแถบเครื่องมือ

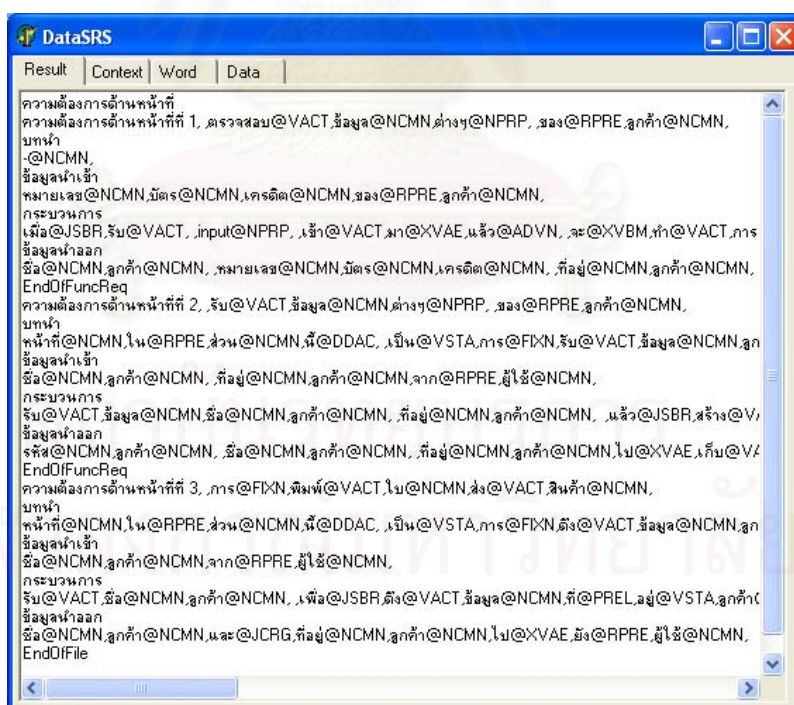
เครื่องมือ	การทำงาน	เมนูการทำงาน
	การสร้างเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์ใหม่	New SRS
	การเปิดเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์	Open SRS
	การบันทึกเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์	Save
	การตัดคำและกำกับหน้าที่คำ	Cut Word
	การวิเคราะห์ประโยคด้วยไวยากรณ์ เพื่อสร้างโครงสร้างใหม่	Show New Structure
	การตรวจสอบความไม่ตรงกันในเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์	Check Inconsistency
	การแสดงผลหน้าจอส่วนเก็บข้อมูล	-
	การปิดเอกสารข้อกำหนดความ ต้องการซอฟต์แวร์	Close SRS

จ-2 การใช้งานหน้าจอส่วนเก็บข้อมูล และฐานข้อมูล

หลังจากการเลือกเครื่องมือ Word บนหน้าจอการทำงานหลัก จะแสดงหน้าจอส่วนเก็บข้อมูลอีกหน้าจอหนึ่งขึ้นมา แสดงดังรูปที่ จ-18 หน้าจอส่วนเก็บข้อมูลมีแถบการทำงาน 4 ส่วน สามารถเปลี่ยนแถบการทำงานได้โดยคลิกเลือกแถบการทำงานที่ต้องการ ผลลัพธ์จากการเปลี่ยนแถบการทำงาน แสดงดังรูปที่ จ-19 ถึงรูปที่ จ-21

หน้าจอในส่วนนี้ทุกแถบการทำงานเป็นเพียงที่เก็บข้อมูลต่าง ๆ ระหว่างที่เครื่องมือซอฟต์แวร์ทำการประมวลผล ซึ่งผู้ใช้งานจะไม่สามารถเข้ามาทำการแก้ไขข้อมูลใด ๆ ได้ แต่สามารถเห็นผลการทำงานระหว่างที่เครื่องมือซอฟต์แวร์ทำการประมวลผลได้อย่างชัดเจน โดยข้อมูลที่เก็บในหน้าจอส่วนนี้มีการอธิบายความหมาย รายละเอียดแล้วในหัวข้อ 4.1.3.2 ของบทที่ 4 ดังนั้นหน้าจอนี้จะไม่มีการอธิบายวิธีการใช้งาน

รูปที่ จ-18 รูปที่ จ-19 และรูปที่ จ-21 ไม่สามารถเปลี่ยนแปลงข้อมูลที่แสดงผลได้ เพราะเกิดจากผลลัพธ์การทำงาน แต่สำหรับรูปที่ จ-20 เป็นข้อมูลของฐานข้อมูลที่เก็บเป็นแฟ้มข้อมูลประเภทข้อความไว้ สามารถทำการเปลี่ยนแปลง เพิ่ม ลบ แก้ไขข้อมูลได้โดยตรงจากการแก้ไขแฟ้มข้อมูล



รูปที่ จ-18 หน้าจอส่วนเก็บข้อมูล – แถบบริซัล

No	Fun	Part	Word	Tag
3	1	0	ตรวจสอบ	VACT
5	1	0	ข้อมูลต่างๆ	NCMN
7	1	0	ของ	RPRE
8	1	0	ลูกค้า	NCMN
14	1	2	หมายเลขบัตร	NCMN
15	1	2	ของ	RPRE
16	1	2	ลูกค้า	NCMN
19	1	3	รับ	VACT
21	1	3	input	NPRP
24	1	3	เข้ามา	XVAE
28	1	3	กระทำ	XVBM
30	1	3	ค้นหา	VACT
32	1	3	ข้อมูลลูกค้า	NCMN
33	1	3	โน	RPRE
34	1	3	ฐานข้อมูล	NCMN
38	1	3	เจาะละ	XVBM

รูปที่ จ-19 หน้าจอส่วนเก็บข้อมูล – แถบคอนเทค

No	Verb	RoleN
1	เก็บ	2
2	รับ	2
3	สร้าง	1
4	แสดง	1
5	ค้นหา	3
6	ดึง	3
7	พิมพ์	2
8	ตรวจสอบ	2
9	ออก	1
10	ชำระ	1
11	ทำ	1
12	สั่งซื้อ	2
13	บันทึก	2
14	จัดซื้อ	2
15	รายงาน	2
16	ปรับปรุง	2

รูปที่ จ-20 หน้าจอส่วนเก็บข้อมูล – แถบเวิร์ด

DataSRS				
Result	Context	Word	Data	
Input	Output	Locate	F	
เลขบัตรเครดิ	ชื่อลูกค้า	1.3	1	
รหัสลูกค้า	หมายเลขมี	1.3	1	
ชื่อลูกค้า	ที่อยู่ลูกค้า	1.3	1	
ที่อยู่ลูกค้า				

Func	Start	End	Data	Func	Start	End	Data	F
ฟังก์ชัน 1.2	1	1	หมายเลข	ฟังก์ชัน 1.1	1	1	หมายเลขบัตร	1
ฟังก์ชัน 2.2	2	3	ข้อมูลชื่อ	ฟังก์ชัน 2.1	2	3	ข้อมูลชื่อลูกค้า	1
ฟังก์ชัน 2.2			ที่อยู่ลูกค้า	ฟังก์ชัน 2.1			ที่อยู่ลูกค้า	1
ฟังก์ชัน 2.3	4	6	ข้อมูลชื่อ	ฟังก์ชัน 2.2	4	6	รหัสลูกค้า	1
ฟังก์ชัน 2.3			ที่อยู่ลูกค้า	ฟังก์ชัน 2.2			ชื่อลูกค้า	1
ฟังก์ชัน 2.3			รหัสลูกค้า	ฟังก์ชัน 2.2			ที่อยู่ลูกค้า	1
ฟังก์ชัน 3.2	7	7	ชื่อลูกค้า	ฟังก์ชัน 3.1	7	7	ชื่อลูกค้า	1

รูปที่ จ-21 หน้าจอส่วนเก็บข้อมูล – แถบตาต้า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก จ.

ตัวอย่างชุดคำสั่งในเครื่องมือซอฟต์แวร์

```
procedure chkFormat;
var i : integer; status : integer;
begin
  CWMain.mmFile.Lines.Add('EndOfFile');
  status := -1;
  if (StrComp(PChar('ความต้องการด้านหน้าที่'),PChar(Trim(CWMain.mmFile.Lines[0]))) = 0) or
    (StrComp(PChar('ความต้องการด้านหน้าที่ (Functional requirements)'),
    PChar(CWMain.mmFile.Lines[0])) = 0) then
    status := 0
  else
    reportError(0,0);
  i := 1;
  while CWMain.mmFile.Lines[i] <> 'EndOfFile' do
  begin
    if (Pos('ความต้องการด้านหน้าที่',CWMain.mmFile.Lines[i]) = 1) then
    begin
      if status <> 0 then
        reportError(status,i);
      status := 1;
    end;
    if (StrPos(PChar(Trim(CWMain.mmFile.Lines[i])), 'บทนำ') = 'บทนำ') or (StrPos(PChar
    (Trim(CWMain.mmFile.Lines[i])), 'บทนำ (Introduction)') then
    begin
      if status <> 1 then
        reportError(status,i);
      status := 2;
    end;
    if (StrPos(PChar(Trim(CWMain.mmFile.Lines[i])), 'ข้อมูลนำเข้า') = 'ข้อมูลนำเข้า') or (StrPos
    (PChar(Trim(CWMain.mmFile.Lines[i])), 'ข้อมูลนำเข้า (Inputs)') then
    begin
      if status <> 2 then
        reportError(status,i);
      status := 3;
    end;
    if (StrPos(PChar(Trim(CWMain.mmFile.Lines[i])), 'กระบวนการ') = 'กระบวนการ') or (StrPos
    (PChar(Trim(CWMain.mmFile.Lines[i])), 'กระบวนการ (Processing)') then
    begin
      if status <> 3 then
        reportError(status,i);
      status := 4;
    end;
    if (StrPos(PChar(Trim(CWMain.mmFile.Lines[i])), 'ข้อมูลนำออก') = 'ข้อมูลนำออก') or (StrPos
    (PChar(Trim(CWMain.mmFile.Lines[i])), 'ข้อมูลนำออก (Outputs)') then
    begin
      if status <> 4 then
        reportError(status,i);
      status := 0;
    end;
    i := i+1;
  end;
  if status <> 0 then
    reportError(status,i);
  CWMain.mmFile.Lines.delete(CWMain.mmFile.Lines.Count-1);
end;
```

รูปที่ จ-1 มอดูลการตรวจสอบรูปแบบของเอกสาร

```

procedure prePrcFile;
var lineText, frontText : string; i : integer; p : integer;
begin
  CWMain.mmFile.Lines.Add('EndOfFile');
  i := 0;
  while CWMain.mmFile.Lines[i] <> 'EndOfFile' do
  begin
    // Concate "๑"
    lineText := CWMain.mmFile.Lines[i];
    p := Pos('๑',lineText);
    frontText := "";
    while (p <> 0) do
    begin
      if Copy(lineText,p-1,1) = ' ' then
        frontText := frontText + TrimRight(Copy(lineText,0,p-1)) + '๑'
      else
        frontText := Copy(lineText,0,p);
        lineText := Copy(lineText,p+1,Length(lineText)-p);
        p := Pos('๑',lineText);
      end;
      CWMain.mmFile.Lines[i] := Trim(frontText + lineText);
      i := i + 1;
    end;
    // Remove Tab
    i := 0;
    while CWMain.mmFile.Lines[i] <> 'EndOfFile' do
    begin
      lineText := CWMain.mmFile.Lines[i];
      p := Pos(' ',lineText);
      frontText := "";
      while (p <> 0) do
      begin
        frontText := frontText + Copy(lineText,0,p-1) + ' ';
        lineText := Copy(lineText,p+1,Length(lineText)-p);
        p := Pos(' ',lineText);
      end;
      CWMain.mmFile.Lines[i] := Trim(frontText + lineText);
      i := i + 1;
    end;
    CWMain.mmFile.Lines.delete(CWMain.mmFile.Lines.Count-1);
  end;
end;

```

รูปที่ จ-2 มอดูลการแก้ไขเปลี่ยนแปลงข้อความในเอกสารโดยอัตโนมัติ

```

procedure cutTag(lineText : string; section : string);
var word : string; remain : string; i, j : integer;
begin
  j := CWMain.strgTag.RowCount;
  i := 0;
  lineText := TrimRight(lineText);
  while Pos(',',lineText) > 0 do
  begin
    CWMain.strgTag.Cells[0,CWMain.strgTag.RowCount-1] :=
    intostr(CWMain.strgTag.RowCount-1);
    CWMain.strgTag.Cells[1,CWMain.strgTag.RowCount-1] := section;
    word := Copy(lineText,0,Pos(',',lineText)-1);
    if (Pos(' ',word) = 0) and (word <> "") then //Check space and tab and delete.
    begin
      if Pos('@',word) > 0 then
      begin
        CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] :=
        Copy(word,0,Pos('@',word)-1);
        CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] :=
        Copy(word,Pos('@',word)+1,Length(word));
        CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
      end
    end
  end
end

```

รูปที่ จ-3 มอดูลการรับข้อมูลของผลลัพธ์จากโปรแกรมตัดคำ

```

else
begin
  if (Length(word) > 4) then //Keyword Except Intro
  begin
    CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] := word;
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := word;
    CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
  end
  else if Integer(word) <> 0 then //Concate string
  begin
    remain := CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-2];
    CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-2] :=
Copy(remain,0,Pos('@',remain)-1);
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-2] :=
Copy(remain,Pos('@',remain)+1,Length(remain))+word;
  end
  end;
end
// function with number
else if (Pos('ความต้องการด้านหน้าที่',word) > 0) then
begin
  if (Pos('ความต้องการด้านหน้าที่',word) > 0) then
    CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] := word;
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'ความต้องการด้านหน้าที่';
    CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
  end
  //Comma
  else if word = " then
  begin
    CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] := ',';
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'COMM';
    CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
    lineText := Copy(lineText,5,Length(lineText));
  end
  // Space and tab
  else
  begin
    CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] := ' ';
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'SPAC';
    CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
  end;
  lineText := Copy(lineText,Pos(',',lineText)+1,Length(lineText));
end;
if ((lineText <> ") and (lineText <> 'EndOfFile')) then
begin
  CWMain.strgTag.Cells[0,CWMain.strgTag.RowCount-1] := inttostr
(CWMain.strgTag.RowCount-1);
  CWMain.strgTag.Cells[1,CWMain.strgTag.RowCount-1] := section;
  CWMain.strgTag.Cells[2,CWMain.strgTag.RowCount-1] := lineText;
  if (Pos('ความต้องการด้านหน้าที่',lineText) > 0) then
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'ความต้องการด้านหน้าที่'
  else if Pos('บทนำ',lineText) > 0 then
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'บทนำ'
  else if Pos('นำเข้า',lineText) > 0 then
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'ข้อมูลนำเข้า'
  else if Pos('กระบวนการ',lineText) > 0 then
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'กระบวนการ'
  else if Pos('นำออก',lineText) > 0 then
    CWMain.strgTag.Cells[3,CWMain.strgTag.RowCount-1] := 'ข้อมูลนำออก';
  CWMain.strgTag.RowCount := CWMain.strgTag.RowCount+1;
  end;
  remSpace(i,j)
end;

```

รูปที่ จ-3 มอดูลการรับข้อมูลของผลลัพธ์จากโปรแกรมตัดคำ (ต่อ)


```

procedure recoverTag;
var i : integer; word : string;
begin
  for i := 1 to CWMain.strgTag.RowCount do // For punctuations and special character
  begin
    if (Length(CWMain.strgTag.Cells[2,i]) = 1) then
    begin
      word := CWMain.strgTag.Cells[2,i];
      if word[1] in ['[',']','\','/','{','}'] then CWMain.strgTag.Cells[3,i] := 'PUNC'
      else if word[1] in ['(',')'] then CWMain.strgTag.Cells[3,i] := 'PUNC'
      else if word[1] in ['-'] then CWMain.strgTag.Cells[3,i] := 'PUNC'
    end;
  end;
  for i:= 1 to CWMain.strgTag.RowCount do
  begin
    if (CWMain.strgTag.Cells[3,i] = 'FIXN') then
    begin
      if CWMain.strgTag.Cells[2,i] = 'การณ' then
        CWMain.strgTag.Cells[3,i+1] := 'VSTA'
      else
        CWMain.strgTag.Cells[3,i+1] := 'VACT'
    end
    else if (CWMain.strgTag.Cells[2,i] = 'ผู้') and (CWMain.strgTag.Cells[3,i+1] = 'VACT') then
    begin
      CWMain.strgTag.Cells[3,i] := 'NCMN';
      CWMain.strgTag.Cells[3,i+1] := 'NCMN'
    end
    //Compound Noun begin with Noun and follow with verb
    else if (CWMain.strgTag.Cells[3,i] = 'NCMN') then
    begin
      if (CWMain.strgTag.Cells[3,i+1] = 'VACT') and ((CWMain.strgTag.Cells[3,i+2] = 'NCMN')
      or (CWMain.strgTag.Cells[3,i+2] = 'SPAC') or (CWMain.strgTag.Cells[3,i+2] = 'SPSP'))
      then
        CWMain.strgTag.Cells[3,i+1] := 'NCMN';
      if (CWMain.strgTag.Cells[3,i+1] = 'VACT') and (CWMain.strgTag.Cells[3,i+2] = 'VACT')
      then
        begin
          CWMain.strgTag.Cells[3,i+1] := 'NCMN';
          CWMain.strgTag.Cells[3,i+2] := 'NCMN';
        end;
    end
    else if (CWMain.strgTag.Cells[3,i] = 'SPAC') or (CWMain.strgTag.Cells[3,i] = 'SPSP') or
    (CWMain.strgTag.Cells[3,i] = 'PUNC') or (Length(CWMain.strgTag.Cells[3,i])>4) or
    (CWMain.strgTag.Cells[3,i] = 'มหน้า') then
    begin
      // Compound noun (begin with PREL)
      if (CWMain.strgTag.Cells[3,i-1] = 'VACT') and (CWMain.strgTag.Cells[3,i-2] = 'PREL') then
      begin
        CWMain.strgTag.Cells[3,i-1] := 'NCMN';
        CWMain.strgTag.Cells[3,i-2] := 'RPRE';
      end
      else if (CWMain.strgTag.Cells[3,i-1] = 'PREL') then // Compound noun (end with PREL)
        CWMain.strgTag.Cells[3,i-1] := 'NCMN'
      else if (CWMain.strgTag.Cells[2,i+1] = '-') and ( (Length(CWMain.strgTag.Cells[3,i])>4) or
      (CWMain.strgTag.Cells[3,i] = 'มหน้า') ) then
        CWMain.strgTag.Cells[3,i+1] := 'DASH';
      end
      else if (CWMain.strgTag.Cells[3,i] = 'PREL') and (CWMain.strgTag.Cells[2,i] = 'ที่') and
      (CWMain.strgTag.Cells[2,i-1] = 'ข้อมูล') then
      begin
        CWMain.strgTag.Cells[3,i] := 'NCMN';
        CWMain.strgTag.Cells[3,i+1] := 'NCMN';
      end;
    end;
  end;
end;
end;

```

รูปที่ ๑-4 มอดูลการแก้ไขหน้าที่คำ

```

procedure remSpace(i : integer; j : integer);
var back, front, countFront, countBack, k, l : integer; wordBack, wordFront : string;
begin
for i := j to CWMain.strgTag.RowCount - 1 do
if (CWMain.strgTag.Cells[3,i] = 'SPAC') then
begin
back := i+1;
front := i-1;
wordBack := Trim(CWMain.strgTag.Cells[2,back]);
wordFront := Trim(CWMain.strgTag.Cells[2,front]);
// Conjunction and Preposition
if Pos('J',PChar(CWMain.strgTag.Cells[3,front])) = 1 then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if Pos('R',PChar(CWMain.strgTag.Cells[3,back])) = 1 then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (Length(wordBack) = 1) or (Length(wordfront) = 1) then
begin
if wordBack = '(' then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if wordFront = ')' then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordFront = ',') or (wordBack = ',') then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordFront[1] in ['0'..'9']) or (wordBack[1] in ['0'..'9']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
end
else
if wordFront[1] in ['0'..'9'] then // Number
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if wordBack[1] in ['0'..'9'] then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordBack[1] in ['n'..'ซ','ม','แ','ใ','ไ','ใ']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordBack[1] in ['น'..'ซ','ม','แ','ใ','ไ','ใ']) and (wordFront[1] in ['a'..'z']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
// Compound English
else if (wordFront[1] in ['A'..'Z']) and (wordBack[1] in ['a'..'z']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordFront[1] in ['a'..'z']) and (wordBack[1] in ['a'..'z']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (wordFront[1] in ['a'..'z']) and (wordBack[1] in ['A'..'Z']) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
// Capital letter begin sentence
else if ((wordFront[1] in ['n'..'ซ','ม','แ','ใ','ไ','ใ']) and (wordBack[1] in ['A'..'Z'])) then
begin
countFront := 0;
countBack := 0;
k := i-1;
while Pos('V',PChar(CWMain.strgTag.Cells[3,k])) <> 1 do
begin
k := k-1;
countFront := countFront + 1;
if k = 0 then break;
end;
l := i+1;
while Pos('V',PChar(CWMain.strgTag.Cells[3,l])) <> 1 do
begin
l := l+1;
countBack := countBack + 1;
if l = CWMain.strgTag.RowCount then break;
end;
if (k = 0) or (l = CWMain.strgTag.RowCount) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
else if (countFront < countBack) then
CWMain.strgTag.Cells[3,i] := 'SPSP'
end
end
end
end
end
end

```

รูปที่ จ-5 มอดูลการกำหนดขอบเขตของประโยค

```

else if ((wordBack[1] in ['น','ฮ','เ','ใ','ใ','ใ']) and (wordFront[1] in ['A'..'Z'])) then
begin
  countFront := 0;
  countBack := 0;
  k := i-1;
  while Pos('V',PChar(CWMain.strgTag.Cells[3,k])) <> 1 do
  begin
    k := k-1;
    countFront := countFront + 1;
    if k = 0 then break;
  end;
  l := j+1;
  while Pos('V',PChar(CWMain.strgTag.Cells[3,l])) <> 1 do
  begin
    l := l+1;
    countBack := countBack + 1;
    if l = CWMain.strgTag.RowCount then break;
  end;
  if (k = 0) or (l = CWMain.strgTag.RowCount) then
    CWMain.strgTag.Cells[3,i] := 'SPSP'
  else if (countFront > countBack) then
    CWMain.strgTag.Cells[3,i] := 'SPSP'
  end;
end;
end;
end;

```

รูปที่ จ-5 มอดูลการกำหนดขอบเขตของประโยค (ต่อ)

```

procedure getContext(first:integer; last:integer);
var i, j, k : integer; tempStr : string; numName, numIntro, numIn,numProc, numOut, numStruct : integer;
begin
  numName := DataSRS.strgVName.RowCount - 1;
  numIntro := DataSRS.strgVIntro.RowCount - 1;
  numIn := DataSRS.strgVIn.RowCount - 1;
  numProc := DataSRS.strgVProc.RowCount - 1;
  numOut := DataSRS.strgVOut.RowCount - 1;
  numStruct := DataSRS.strgStruct.RowCount - 1;
  i := first ;
  while i < last do
    if DataSRS.strgContext.Cells[4,i] = 'VACT' then
      begin
        case StrToInt(DataSRS.strgContext.Cells[2,i]) of
          0: begin
              DataSRS.strgVName.Cells[0,DataSRS.strgVName.RowCount-1] := IntroStr(i);
              DataSRS.strgVName.Cells[1,DataSRS.strgVName.RowCount-1] := "";
              Repeat
                DataSRS.strgVName.Cells[1,DataSRS.strgVName.RowCount-1] :=
DataSRS.strgVName.Cells[1,DataSRS.strgVName.RowCount-1] + DataSRS.strgContext.Cells[3,i];
                i := i+1;
              until ((DataSRS.strgContext.Cells[4,i] <> 'VACT') and (DataSRS.strgContext.Cells[4,i] <>
'VSTA')) or (i = DataSRS.strgContext.RowCount-1) or (StrToInt(DataSRS.strgContext.Cells[5,i]) <
StrToInt(DataSRS.strgContext.Cells[5,i-1]));
                j := i-1;
                while (Copy(DataSRS.strgContext.Cells[4,i],0,1) <> 'N') and (i <
DataSRS.strgContext.RowCount-1) and (StrToInt(DataSRS.strgContext.Cells[5,i]) >
StrToInt(DataSRS.strgContext.Cells[5,j])) do
                  l := i+1;
                  DataSRS.strgVName.Cells[2,DataSRS.strgVName.RowCount-1] :=
DataSRS.strgContext.Cells[3,i];
                  // Multiple Data
                  if (DataSRS.strgContext.Cells[5,i+1] = '1') and (Pos('V',DataSRS.strgContext.Cells[4,i+1]) <> 1)
and (DataSRS.strgContext.Cells[2,i] = DataSRS.strgContext.Cells[2,i+1]) then
                    begin
                      k := i+1;
                      tempStr := "";

```

รูปที่ จ-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค

```

Repeat
  if DataSRS.strgContext.Cells[5,k] <= DataSRS.strgContext.Cells[5,k-1] then
    tempStr := tempStr + ' ';
    tempStr := tempStr + DataSRS.strgContext.Cells[3,k];
    k := k+1;
    until ((DataSRS.strgContext.Cells[4,k] = 'VACT') and (DataSRS.strgContext.Cells[5,k+1] <>
'1') and (Copy(DataSRS.strgContext.Cells[4,k+1],0,1) <> 'R')) or ((DataSRS.strgContext.Cells[4,k] =
'VACT') and (DataSRS.strgContext.Cells[5,k] = '1')) or ((DataSRS.strgContext.Cells[4,k] = 'VACT') and
(DataSRS.strgContext.Cells[5,k] = '1')) or (k = DataSRS.strgContext.RowCount-1) or
(DataSRS.strgContext.Cells[2,k-1] <> DataSRS.strgContext.Cells[2,k]);
    DataSRS.strgVName.Cells[2,DataSRS.strgVName.RowCount-1] :=
DataSRS.strgVName.Cells[2,DataSRS.strgVName.RowCount-1] + tempStr;
    i := k;
  end;
  // Find Agent
  j := i+1;
  while (DataSRS.strgContext.Cells[4,j] <> 'VACT') and (j <= DataSRS.strgContext.RowCount-1) do
  begin
    if (chkAgent(DataSRS.strgContext.Cells[3,j]) = 1) and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ') then
      begin
        DataSRS.strgVName.Cells[5,DataSRS.strgVName.RowCount-1] :=
DataSRS.strgContext.Cells[3,j];
        break;
      end;
      j := j+1;
    end;
    DataSRS.strgVName.Cells[3,DataSRS.strgVName.RowCount-1] := '0';
    DataSRS.strgVName.Cells[4,DataSRS.strgVName.RowCount-1] := '2';
    DataSRS.strgVName.RowCount := DataSRS.strgVName.RowCount+1;
  end;
1: begin
  DataSRS.strgVIntro.Cells[0,DataSRS.strgVIntro.RowCount-1] := InttoStr(i);
  DataSRS.strgVIntro.Cells[1,DataSRS.strgVIntro.RowCount-1] := ";
  Repeat
    DataSRS.strgVIntro.Cells[1,DataSRS.strgVIntro.RowCount-1] :=
DataSRS.strgVIntro.Cells[1,DataSRS.strgVIntro.RowCount-1] + DataSRS.strgContext.Cells[3,i];
    i := i+1;
    until ((DataSRS.strgContext.Cells[4,i] <> 'VACT') and (DataSRS.strgContext.Cells[4,i] <>
'VSTA') or (i = DataSRS.strgContext.RowCount-1) or (StrToInt(DataSRS.strgContext.Cells[5,i]) <
StrToInt(DataSRS.strgContext.Cells[5,i-1])));
    j := i-1;
    while (Copy(DataSRS.strgContext.Cells[4,i],0,1) <> 'N') and (i <
DataSRS.strgContext.RowCount-1) and (StrToInt(DataSRS.strgContext.Cells[5,i]) >
StrToInt(DataSRS.strgContext.Cells[5,j])) do i := i+1;
    DataSRS.strgVIntro.Cells[2,DataSRS.strgVIntro.RowCount-1] :=
DataSRS.strgContext.Cells[3,i];
    // Multiple Data
    if (DataSRS.strgContext.Cells[5,i+1] = '1') and (Pos('V',DataSRS.strgContext.Cells[4,i+1]) <> 1)
and (DataSRS.strgContext.Cells[2,i] = DataSRS.strgContext.Cells[2,i+1]) and
(Pos('R',DataSRS.strgContext.Cells[4,i+1]) <> 1) then
      begin
        k := i+1;
        tempStr := ";
        Repeat
          if DataSRS.strgContext.Cells[5,k] <= DataSRS.strgContext.Cells[5,k-1] then
            tempStr := tempStr + ' ';
            tempStr := tempStr + DataSRS.strgContext.Cells[3,k];
            k := k+1;
            until ((DataSRS.strgContext.Cells[4,k] = 'VACT') and (DataSRS.strgContext.Cells[5,k+1] <>
'1') and (Copy(DataSRS.strgContext.Cells[4,k+1],0,1) <> 'R')) or ((DataSRS.strgContext.Cells[4,k] =
'VACT') and (DataSRS.strgContext.Cells[5,k] = '1')) or (k = DataSRS.strgContext.RowCount-1) or
(DataSRS.strgContext.Cells[2,k-1] <> DataSRS.strgContext.Cells[2,k]) or (DataSRS.strgContext.
Cells[4,k] = 'RPRE');

```

รูปที่ จ-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)

```

DataSRS.strgVIntro.Cells[2,DataSRS.strgVIntro.RowCount-1] :=
DataSRS.strgVIntro.Cells[2,DataSRS.strgVIntro.RowCount-1] + tempStr;
i := k;
end;
// Find Agent
j := i+1;
while (DataSRS.strgContext.Cells[4,j] <> 'VACT') and (j <= DataSRS.strgContext.RowCount-1)
do
begin
if (chkAgent(DataSRS.strgContext.Cells[3,j]) = 1) and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ') then
begin
DataSRS.strgVIntro.Cells[5,DataSRS.strgVIntro.RowCount-1] :=
DataSRS.strgContext.Cells[3,j];
break;
end;
j := j+1;
end;
DataSRS.strgVIntro.Cells[3,DataSRS.strgVIntro.RowCount-1] := '0';
DataSRS.strgVIntro.Cells[4,DataSRS.strgVIntro.RowCount-1] := '3';
DataSRS.strgVIntro.RowCount := DataSRS.strgVIntro.RowCount+1;
end;
2: begin
// Noun before verb
if (DataSRS.strgContext.Cells[5,i-1] = '1') or (DataSRS.strgContext.Cells[5,i] <> '1') then
begin
j := i;
Repeat
j := j-1;
until (DataSRS.strgContext.Cells[4,j] = 'NCMN') or (DataSRS.strgContext.Cells[5,j] = '1');
if (DataSRS.strgContext.Cells[4,j] = 'NCMN') then
begin
tempStr := DataSRS.strgContext.Cells[3,j];
j := j-1;
while (DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[2,j]
= '4') do
begin
tempStr := DataSRS.strgContext.Cells[3,j] + ' ' + tempStr;
j := j-1;
end;
end;
DataSRS.strgVIn.Cells[6,DataSRS.strgVIn.RowCount-1] := tempStr;
end;
DataSRS.strgVIn.Cells[0,DataSRS.strgVIn.RowCount-1] := IntroStr(i);
DataSRS.strgVIn.Cells[1,DataSRS.strgVIn.RowCount-1] := "";
Repeat
DataSRS.strgVIn.Cells[1,DataSRS.strgVIn.RowCount-1] :=
DataSRS.strgVIn.Cells[1,DataSRS.strgVIn.RowCount-1] + DataSRS.strgContext.Cells[3,i];
i := i+1;
until ((DataSRS.strgContext.Cells[4,i] <> 'VACT') and (DataSRS.strgContext.Cells[4,i] <>
'VSTA')) or (i = DataSRS.strgContext.RowCount-1) or (StrToInt(DataSRS.strgContext.Cells[5,i]) <
StrToInt(DataSRS.strgContext.Cells[5,i-1]));
j := i-1;
while (Copy(DataSRS.strgContext.Cells[4,i],0,1) <> 'N') and (i <
DataSRS.strgContext.RowCount-1) and (StrToInt(DataSRS.strgContext.Cells[5,i]) >
StrToInt(DataSRS.strgContext.Cells[5,j]) ) do
i := i+1;
DataSRS.strgVIn.Cells[2,DataSRS.strgVIn.RowCount-1] :=
DataSRS.strgContext.Cells[3,i];
// Multiple Data
if (DataSRS.strgContext.Cells[5,i+1] = '1') and (Pos('V',DataSRS.strgContext.Cells[4,i+1]) <> 1)
and (DataSRS.strgContext.Cells[2,i] = DataSRS.strgContext.Cells[2,i+1]) then
begin
k := i+1;
tempStr := "";

```

รูปที่ ๖-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)

```

Repeat
  if DataSRS.strgContext.Cells[5,k] <= DataSRS.strgContext.Cells[5,k-1] then
    tempStr := tempStr + ' ';
    tempStr := tempStr + DataSRS.strgContext.Cells[3,k];
    k := k+1;
  until ((DataSRS.strgContext.Cells[4,k] = 'VACT') and (DataSRS.strgContext.Cells[5,k+1] <>
'1') and (Copy(DataSRS.strgContext.Cells[4,k+1],0,1) <> 'R')) or ((DataSRS.strgContext.Cells[4,k] =
'VACT') and (DataSRS.strgContext.Cells[5,k] = '1')) or (k = DataSRS.strgContext.RowCount-1) or
(DataSRS.strgContext.Cells[2,k-1] <> DataSRS.strgContext.Cells[2,k]);
  DataSRS.strgVIn.Cells[2,DataSRS.strgVIn.RowCount-1] :=
DataSRS.strgVIn.Cells[2,DataSRS.strgVIn.RowCount-1] + tempStr;
  i := k;
  end;
  // Find Agent
  j := i+1;
  while (DataSRS.strgContext.Cells[4,j] <> 'VACT') and (j <= DataSRS.strgContext.RowCount-1) do
begin
  if (chkAgent(DataSRS.strgContext.Cells[3,j]) = 1) and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ') then
begin
  DataSRS.strgVIn.Cells[5,DataSRS.strgVIn.RowCount-1] := DataSRS.strgContext.Cells[3,j];

  break;
end;
j := j+1;
end;
DataSRS.strgVIn.Cells[3,DataSRS.strgVIn.RowCount-1] := '0';
DataSRS.strgVIn.Cells[4,DataSRS.strgVIn.RowCount-1] := '1';
DataSRS.strgVIn.RowCount := DataSRS.strgVIn.RowCount+1;
end;
3: begin
DataSRS.strgVProc.Cells[0,DataSRS.strgVProc.RowCount-1] := InttoStr(i);
DataSRS.strgVProc.Cells[1,DataSRS.strgVProc.RowCount-1] := "";
Repeat
  DataSRS.strgVProc.Cells[1,DataSRS.strgVProc.RowCount-1] := DataSRS.
strgVProc.Cells[1,DataSRS.strgVProc.RowCount-1] + DataSRS.strgContext.Cells[3,i];
  i := i+1;
  until ((DataSRS.strgContext.Cells[4,i] <> 'VACT') and (DataSRS.strgContext.Cells[4,i] <>
'VSTA')) or (i = DataSRS.strgContext.RowCount-1) or (StrToInt(DataSRS.strgContext.Cells[5,i]) <
StrToInt(DataSRS.strgContext.Cells[5,i-1]));
  j := i-1;
  while (Copy(DataSRS.strgContext.Cells[4,i],0,1) <> 'N') and (i <
DataSRS.strgContext.RowCount-1) and (StrToInt(DataSRS.strgContext.Cells[5,i]) >
StrToInt(DataSRS.strgContext.Cells[5,j])) do i := i+1;
  DataSRS.strgVProc.Cells[2,DataSRS.strgVProc.RowCount-1] :=
DataSRS.strgContext.Cells[3,i];
  // Multiple Data
  if (DataSRS.strgContext.Cells[5,i+1] = '1') and (Pos('V',DataSRS.strgContext.Cells[4,i+1]) <>
1) and (DataSRS.strgContext.Cells[2,i] = DataSRS.strgContext.Cells[2,i+1]) then
begin
  k := i+1;
  tempStr := "";
  Repeat
    if DataSRS.strgContext.Cells[5,k] <= DataSRS.strgContext.Cells[5,k-1] then
      tempStr := tempStr + ' ';
      tempStr := tempStr + DataSRS.strgContext.Cells[3,k];
      k := k+1;
    until ((DataSRS.strgContext.Cells[4,k] = 'VACT') and (DataSRS.strgContext.Cells[5,k+1] <>
'1') and (Copy(DataSRS.strgContext.Cells[4,k+1],0,1) <> 'R')) or ((DataSRS.strgContext.Cells[4,k] =
'VACT') and (DataSRS.strgContext.Cells[5,k] = '1')) or (k = DataSRS.strgContext.RowCount-1) or
(DataSRS.strgContext.Cells[2,k-1] <> DataSRS.strgContext.Cells[2,k]);
    DataSRS.strgVProc.Cells[2,DataSRS.strgVProc.RowCount-1] :=
DataSRS.strgVProc.Cells[2,DataSRS.strgVProc.RowCount-1] + tempStr;
    i := k;
  end;
end;

```

รูปที่ ๑-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)

```

// Find Agent
j := i+1;
while (DataSRS.strgContext.Cells[4,j] <> 'VACT') and (j <= DataSRS.strgContext.RowCount-1) do
begin
if (chkAgent(DataSRS.strgContext.Cells[3,j]) = 1) and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ') then
begin
DataSRS.strgVProc.Cells[5,DataSRS.strgVProc.RowCount-1] :=
DataSRS.strgContext.Cells[3,j];
break;
end;
j := j+1;
end;
DataSRS.strgVProc.Cells[3,DataSRS.strgVProc.RowCount-1] := '0';
DataSRS.strgVProc.Cells[4,DataSRS.strgVProc.RowCount-1] := '4';
DataSRS.strgVProc.RowCount := DataSRS.strgVProc.RowCount+1;
end;
4: begin
if (DataSRS.strgContext.Cells[5,i-1] = '1') or (DataSRS.strgContext.Cells[5,i] <> '1') then
begin // Noun before verb
j := i;
Repeat
j := j-1;
until (DataSRS.strgContext.Cells[4,j] = 'NCMN') or (DataSRS.strgContext.Cells[5,j] = '1');
if (DataSRS.strgContext.Cells[4,j] = 'NCMN') then
begin
tempStr := DataSRS.strgContext.Cells[3,j];
j := j-1;
while (DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[2,j]
= '4') do
begin
tempStr := DataSRS.strgContext.Cells[3,j] + ' ' + tempStr;
j := j-1;
end;
end;
DataSRS.strgVOut.Cells[6,DataSRS.strgVOut.RowCount-1] := tempStr;
end;
DataSRS.strgVOut.Cells[0,DataSRS.strgVOut.RowCount-1] := InttoStr(i);
DataSRS.strgVOut.Cells[1,DataSRS.strgVOut.RowCount-1] := "";
Repeat
DataSRS.strgVOut.Cells[1,DataSRS.strgVOut.RowCount-1] := DataSRS.
strgVOut.Cells[1,DataSRS.strgVOut.RowCount-1] + DataSRS.strgContext.Cells[3,i];
i := i+1;
until ((DataSRS.strgContext.Cells[4,i] <> 'VACT') and (DataSRS.strgContext.Cells[4,i] <>
'VSTA')) or (i = DataSRS.strgContext.RowCount-1) or (StrToInt(DataSRS.strgContext.Cells[5,i]) <
StrToInt(DataSRS.strgContext.Cells[5,i-1]));
j := i-1;
while (Copy(DataSRS.strgContext.Cells[4,i],0,1) <> 'N') and (i <
DataSRS.strgContext.RowCount-1) and (StrToInt(DataSRS.strgContext.Cells[5,i]) >
StrToInt(DataSRS.strgContext.Cells[5,j])) do i := i+1;
DataSRS.strgVOut.Cells[2,DataSRS.strgVOut.RowCount-1] :=
DataSRS.strgContext.Cells[3,i];
if (DataSRS.strgContext.Cells[5,i+1] = '1') and (Pos('V',DataSRS.strgContext.Cells[4,i+1]) <>
1) and (DataSRS.strgContext.Cells[2,i] = DataSRS.strgContext.Cells[2,i+1]) then
begin // Multiple Data
k := i+1;
tempStr := "";
Repeat
if DataSRS.strgContext.Cells[5,k] <= DataSRS.strgContext.Cells[5,k-1] then
tempStr := tempStr + ' ';
tempStr := tempStr + DataSRS.strgContext.Cells[3,k];
k := k+1;
until ((DataSRS.strgContext.Cells[4,k] = 'VACT') and (DataSRS.strgContext.Cells[5,k+1] <>
'1') and (Copy(DataSRS.strgContext.Cells[4,k+1],0,1) <> 'R')) or ((DataSRS.strgContext.Cells[4,k] =
'VACT') and (DataSRS.strgContext.Cells[5,k] = '1')) or (k = DataSRS.strgContext.RowCount-1) or
(DataSRS.strgContext.Cells[2,k-1] <> DataSRS.strgContext.Cells[2,k]);

```

รูปที่ ๖-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)

```

DataSRS.strgVOut.Cells[2,DataSRS.strgVOut.RowCount-1] :=
DataSRS.strgVOut.Cells[2,DataSRS.strgVOut.RowCount-1] + tempStr;
i := k;
end;
// Find Agent
j := i+1;
while (DataSRS.strgContext.Cells[4,j] <> 'VACT') and (j <= DataSRS.strgContext.RowCount-1)
do
begin
if (chkAgent(DataSRS.strgContext.Cells[3,j]) = 1) and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ') then
begin
DataSRS.strgVOut.Cells[5,DataSRS.strgVOut.RowCount-1] :=
DataSRS.strgContext.Cells[3,j];
break;
end;
j := j+1;
end;
DataSRS.strgVOut.Cells[3,DataSRS.strgVOut.RowCount-1] := '0';
DataSRS.strgVOut.Cells[4,DataSRS.strgVOut.RowCount-1] := '1';
DataSRS.strgVOut.RowCount := DataSRS.strgVOut.RowCount+1;
end;
end;
else if (chkAgent(DataSRS.strgContext.Cells[3,i]) = 1) then
begin
case StrToInt(DataSRS.strgContext.Cells[2,i]) of
2: begin
DataSRS.strgVIn.Cells[0,DataSRS.strgVIn.RowCount-1] := DataSRS.strgContext.Cells[0,i];
DataSRS.strgVIn.Cells[5,DataSRS.strgVIn.RowCount-1] := DataSRS.strgContext.Cells[3,i];
j := i;
Repeat
j := j-1;
until ((DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ')) or (DataSRS.strgContext.Cells[2,j] <> '2');
if (DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[5,j] <=
DataSRS.strgContext.Cells[5,i]) then
begin
k := j-1;
tempStr := DataSRS.strgContext.Cells[3,j];
while (DataSRS.strgContext.Cells[4,k] = 'NCMN') and (DataSRS.strgContext.Cells[2,k]
= '2') do
begin
tempStr := DataSRS.strgContext.Cells[3,k] + '' + tempStr;
k := k-1;
end;
DataSRS.strgVIn.Cells[2,DataSRS.strgVIn.RowCount-1] := tempStr;
end;
DataSRS.strgVIn.RowCount := DataSRS.strgVIn.RowCount+1;
end;
4: begin
DataSRS.strgVOut.Cells[0,DataSRS.strgVOut.RowCount-1] :=
DataSRS.strgContext.Cells[0,i];
DataSRS.strgVOut.Cells[5,DataSRS.strgVOut.RowCount-1] :=
DataSRS.strgContext.Cells[3,i]; j := i;
Repeat
j := j-1;
until ((DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[3,j-1] <>
'ของ')) or (DataSRS.strgContext.Cells[2,j] <> '4');
if (DataSRS.strgContext.Cells[4,j] = 'NCMN') and (DataSRS.strgContext.Cells[5,j] <=
DataSRS.strgContext.Cells[5,i]) then
begin
k := j-1;
tempStr := DataSRS.strgContext.Cells[3,j];
while (DataSRS.strgContext.Cells[4,k] = 'NCMN') and (DataSRS.strgContext.Cells[2,k] =
'4') do

```

รูปที่ ๖-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)


```

begin
    tempStr := DataSRS.strgContext.Cells[3,k] + '' + tempStr;
    k := k-1;
end;
DataSRS.strgVOut.Cells[2,DataSRS.strgVOut.RowCount-1] := tempStr;
end;
DataSRS.strgVOut.RowCount := DataSRS.strgVOut.RowCount+1;
end;
end;
i := i + 1;
end
else
    i := i + 1;
weighStn(numName, numIntro, numIn, numProc, numOut);
printStn(numName, numIntro, numIn, numProc, numOut);
recoverStn(numStruct);
getAgent(numIn, numOut, numStruct);
getCase(numStruct);
end;

```

รูปที่ จ-6 มอดูลการเลือกส่วนสำคัญของประโยคและเลือกประโยค (ต่อ)

```

procedure getCase(nS : integer);
var i,j : integer; str1, str2, num : string; fn1, fn2 : integer; nC, nA : integer;
begin
    nC := CWMain.strgCase.RowCount - 1;
    nA := CWMain.strgCaseAll.RowCount - 1;
    for i:=nS to DataSRS.strgStruct.RowCount-1 do
        for j:=i+1 to DataSRS.strgStruct.RowCount-1 do
            if DataSRS.strgStruct.Cells[2,i] = DataSRS.strgStruct.Cells[2,j] then
                begin
                    if Copy(DataSRS.strgStruct.Cells[3,i],1,6) = 'ข้อมูล' then
                        str1 := Copy(DataSRS.strgStruct.Cells[3,i],7,Length(DataSRS.strgStruct.Cells[3,i]));
                    if Copy(DataSRS.strgStruct.Cells[3,j],1,6) = 'ข้อมูล' then
                        str2 := Copy(DataSRS.strgStruct.Cells[3,j],7,Length(DataSRS.strgStruct.Cells[3,j]));
                    if (Pos(str1,str2) > 0) and (str1 <> str2) then
                        DataSRS.strgStruct.Cells[5,i] := '0'
                    else if Pos(str2,str1) > 0 then
                        begin
                            DataSRS.strgStruct.Cells[5,j] := '0';
                            if (DataSRS.strgStruct.Cells[1,i] = "") and (DataSRS.strgStruct.Cells[1,j] <> "") then
                                DataSRS.strgStruct.Cells[1,i] := DataSRS.strgStruct.Cells[1,j];
                            end
                        end
                    else
                        if DataSRS.strgStruct.Cells[5,j] > DataSRS.strgStruct.Cells[5,i] then
                            DataSRS.strgStruct.Cells[5,i] := '0'
                        else
                            DataSRS.strgStruct.Cells[5,j] := '0';
                        end;
                end
            //Define number of function
        for i:= nS to DataSRS.strgStruct.RowCount-1 do
            begin
                if DataSRS.strgStruct.Cells[5,i] <> '0' then
                    begin
                        if (i = nS) or (CWMain.strgCase.RowCount = 2) then
                            CWMain.strgCase.Cells[0,CWMain.strgCase.RowCount-1] := '1'
                        else
                            CWMain.strgCase.Cells[0,CWMain.strgCase.RowCount-1] :=
                                IntroStr(StrToInt(CWMain.strgCase.Cells[0,CWMain.strgCase.RowCount-2])+1);
                            CWMain.strgCase.Cells[1,CWMain.strgCase.RowCount-1] := 'หน้าที่ที่ ' +
                                DataSRS.strgStruct.Cells[0,i];
                            CWMain.strgCase.Cells[4,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[2,i] +
                                DataSRS.strgStruct.Cells[3,i];
                            CWMain.strgCase.Cells[5,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[3,i];
                            CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[1,i];
                    end
                end
            end

```

รูปที่ จ-7 มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมให้โครงสร้างใหม่สมบูรณ์ขึ้น

```

if (DataSRS.strgStruct.Cells[1,i] <> "") and (DataSRS.strgStruct.Cells[4,i] = '2') then
begin
  CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[1,i]
  DataSRS.strgStruct.Cells[1,i];
end
else if DataSRS.strgStruct.Cells[4,i] = '2' then
  CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[3,i]
else if DataSRS.strgStruct.Cells[4,i] = '3' then
  CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] := "";
  CWMain.strgCase.Cells[2,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[6,i];
  CWMain.strgCase.Cells[6,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.Cells[7,i];
  CWMain.strgCase.Cells[7,CWMain.strgCase.RowCount-1] := IntToStr(i);
  CWMain.strgCase.RowCount := CWMain.strgCase.RowCount+1;
end;
end;
if CWMain.strgCase.RowCount > 2 then
  CWMain.strgCase.RowCount := CWMain.strgCase.RowCount-1;
  //Check duplicate of Source and Destination
  for i:=nC+1 to CWMain.strgCase.RowCount - 1 do
  begin
    j := nC-1;
    Repeat
      j := j+1
    until ((CWMain.strgCase.Cells[1,j] = CWMain.strgCase.Cells[1,i]) and (CWMain.strgCase.Cells[2,i] =
CWMain.strgCase.Cells[2,j])) or (j=i);
    if j<i then
    begin
      CWMain.strgCase.Cells[2,i] := "";
      if Copy(CWMain.strgCase.Cells[3,i],1,6) = 'ข้อมูล' then
      begin
        str1 := Copy(CWMain.strgCase.Cells[3,i],7,Length(CWMain.strgCase.Cells[3,i]));
        if (Pos(str1,CWMain.strgCase.Cells[3,j]) >0) then
          CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j]
        else if j = nC then
          CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j]
        else
          CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j] + ' ' +
CWMain.strgCase.Cells[3,i];
        end
        else if Pos(CWMain.strgCase.Cells[3,i],CWMain.strgCase.Cells[3,j]) > 0 then
          CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j]
        else if Copy(CWMain.strgCase.Cells[3,j],1,6) = 'ข้อมูล' then
        begin
          str2 := Copy(CWMain.strgCase.Cells[3,j],7,Length(CWMain.strgCase.Cells[3,j]));
          if (Pos(str2,CWMain.strgCase.Cells[3,i]) >0) then
            CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j]
          else
            CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j] + ' ' +
CWMain.strgCase.Cells[3,i];
          end
          else if j = nC then
            CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j]
          else
            CWMain.strgCase.Cells[3,j] := CWMain.strgCase.Cells[3,j] + ' ' +
CWMain.strgCase.Cells[3,i]
          end;
        end;
      end;
    for i:= nC to CWMain.strgCase.RowCount - 1 do
      if DataSRS.strgStruct.Cells[4,StrToInt(CWMain.strgCase.Cells[7,i])] = '3' then
        CWMain.strgCase.Cells[3,i] := "";
    for i := nC to CWMain.strgCase.RowCount-1 do
    begin
      j := CWMain.strgCase.RowCount;
      Repeat
        j := j-1;
      until ((CWMain.strgCase.Cells[1,j] = CWMain.strgCase.Cells[1,i]) and (CWMain.strgCase.Cells[6,i] =
CWMain.strgCase.Cells[6,j])) or (j=i);

```

รูปที่ ๗-7 มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมให้โครงสร้างใหม่สมบูรณ์ขึ้น (ต่อ)

```

if j>i then
begin
  CWMain.strgCase.Cells[6,i] := "";
  if Copy(CWMain.strgCase.Cells[5,j],1,6) = 'ข้อมูล' then
  begin
    str1 := Copy(CWMain.strgCase.Cells[5,j],7,Length(CWMain.strgCase.Cells[5,j]));
    if Pos(str1,CWMain.strgCase.Cells[5,i]) > 0 then
      CWMain.strgCase.Cells[5,j] := CWMain.strgCase.Cells[5,i]
    else
      CWMain.strgCase.Cells[5,j] := CWMain.strgCase.Cells[5,i] + ' ' +
CWMain.strgCase.Cells[5,i];
    end
    else if Pos(CWMain.strgCase.Cells[5,i],CWMain.strgCase.Cells[5,j]) > 0 then
      CWMain.strgCase.Cells[5,j] := CWMain.strgCase.Cells[5,i];
    end;
  end;
  //Output from Output Part
  if StrToInt(CWMain.strgCase.Cells[7,CWMain.strgCase.RowCount-1]) <
DataSRS.strgStruct.RowCount-1 then
  if Pos(DataSRS.strgStruct.Cells[2,StrToInt(CWMain.strgCase.Cells[7,CWMain.strgCase. RowCount-
1])),DataSRS.strgStruct.Cells[2,StrToInt(CWMain.strgCase.Cells[7, CWMain.strgCase.RowCount-1])+1])
> 0 then
    CWMain.strgCase.Cells[5,CWMain.strgCase.RowCount-1] := DataSRS.strgStruct.
Cells[1,StrToInt(CWMain.strgCase.Cells[7,CWMain.strgCase.RowCount-1])+1];
    chkDupFunc(nC);
    for i:= nC to CWMain.strgCase.RowCount-1 do
    begin
      CWMain.strgCaseAll.Cells[0,CWMain.strgCaseAll.RowCount-1] := CWMain.strgCase.Cells[0,i];
      CWMain.strgCaseAll.Cells[1,CWMain.strgCaseAll.RowCount-1] := CWMain.strgCase.Cells[1,i];
      if (CWMain.strgCase.Cells[2,i] = "") and (i <> 1) then
      begin
        if (CWMain.strgCase.Cells[6,i-1] = "") and (Pos('.',CWMain.strgCase.Cells[6,i-1])=0) then
          CWMain.strgCaseAll.Cells[2,CWMain.strgCaseAll.RowCount-1] := 'ฟังก์ชัน ' +
Copy(CWMain.strgCase.Cells[1,i],Pos(' ',CWMain.strgCase.Cells[1,i])+1,
Length(CWMain.strgCase.Cells[1,i])) + '.'+CWMain.strgCase.Cells[0,i-1]
        else
          CWMain.strgCaseAll.Cells[2,CWMain.strgCaseAll.RowCount-1] :=
CWMain.strgCase.Cells[6,i-1];
        end
        else
          CWMain.strgCaseAll.Cells[2,CWMain.strgCaseAll.RowCount-1] := CWMain.strgCase.Cells[2,i];
          CWMain.strgCaseAll.Cells[3,CWMain.strgCaseAll.RowCount-1] :=
Trim(CWMain.strgCase.Cells[3,i]);
          CWMain.strgCaseAll.Cells[4,CWMain.strgCaseAll.RowCount-1] := CWMain.strgCase.Cells[4,i];
          CWMain.strgCaseAll.Cells[5,CWMain.strgCaseAll.RowCount-1] :=
Trim(CWMain.strgCase.Cells[5,i]);
          if (CWMain.strgCase.Cells[6,i] = "") and (i <> CWMain.strgCase.RowCount-1 ) then
            CWMain.strgCaseAll.Cells[6,CWMain.strgCaseAll.RowCount-1] := 'ฟังก์ชัน ' +
Copy(CWMain.strgCase.Cells[1,i],Pos(' ',CWMain.strgCase.Cells[1,i])+1,
Length(CWMain.strgCase.Cells[1,i])) + '.'+CWMain.strgCase.Cells[0,i+1]
          else
            CWMain.strgCaseAll.Cells[6,CWMain.strgCaseAll.RowCount-1] := CWMain.strgCase.Cells[6,i];
            CWMain.strgCaseAll.RowCount := CWMain.strgCaseAll.RowCount+1;
          end;
        if CWMain.strgCaseAll.RowCount > 2 then
          CWMain.strgCaseAll.RowCount := CWMain.strgCaseAll.RowCount-1;
          for i:= nA to CWMain.strgCaseAll.RowCount-1 do //Check and input value for null value
            if CWMain.strgCaseAll.Cells[3,i] = "" then
              begin
                if (Pos('.',CWMain.strgCaseAll.Cells[2,i]) > 0) and (Pos('.',CWMain.strgCaseAll.Cells[6,i-1]) > 0)
then
                  begin
                    fn1 := StrToInt(Copy(CWMain.strgCaseAll.Cells[2,i],Pos('.',CWMain.strgCaseAll.Cells[2,i])
+1,Length(CWMain.strgCaseAll.Cells[2,i]));
                    fn2 := StrToInt(Copy(CWMain.strgCaseAll.Cells[6,i-1],Pos('.',CWMain.strgCaseAll.Cells[6,i-
1])+1,Length(CWMain.strgCaseAll.Cells[6,i-1]));

```

รูปที่ จ-7 มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมให้โครงสร้างใหม่สมบูรณ์ขึ้น (ต่อ)

```

if fn1+1 = fn2 then
begin
  CWMain.strgCaseAll.Cells[3,i] := CWMain.strgCaseAll.Cells[5,i-1];
  if Pos(CWMain.strgCaseAll.Cells[3,i],CWMain.strgCaseAll.Cells[5,i]) = 0 then
  begin
    if Copy(CWMain.strgCaseAll.Cells[5,i],1,6) = 'ข้อมูล' then
    begin
      if (chkAgent(Copy(CWMain.strgCaseAll.Cells[5,i],7,Length(CWMain.strgCaseAll.
Cells[5,i]))) = 1) or (DataSRS.strgStruct.Cells[4, StrToInt(CWMain.strgCase.Cells[7,i])] = '3') then
      begin
        if (DataSRS.strgStruct.Cells[4, StrToInt(CWMain.strgCase.Cells[7,i])] = '3') and
(DataSRS.strgStruct.Cells[2, StrToInt(CWMain.strgCase.Cells[7,i])] = 'ตั้ง') then
          CWMain.strgCaseAll.Cells[3,i+1] := CWMain.strgCaseAll.Cells[3,i] + ' ' +
CWMain.strgCaseAll.Cells[5,i];
          CWMain.strgCaseAll.Cells[5,i] := CWMain.strgCaseAll.Cells[3,i]
        end;
      end
    else if DataSRS.strgStruct.Cells[4, StrToInt(CWMain.strgCase.Cells[7,i])] = '3' then
    begin
      CWMain.strgCaseAll.Cells[5,i] := CWMain.strgCaseAll.Cells[3,i];
    end
    else
      CWMain.strgCaseAll.Cells[5,i] := CWMain.strgCaseAll.Cells[3,i] + ' ' +
CWMain.strgCaseAll.Cells[5,i];
    end;
  end;
end
else if DataSRS.strgStruct.Cells[4, StrToInt(CWMain.strgCase.Cells[7,i])] = '3' then
begin
  CWMain.strgCaseAll.Cells[3,i] := CWMain.strgCaseAll.Cells[5,i-1];
  CWMain.strgCaseAll.Cells[5,i] := CWMain.strgCaseAll.Cells[3,i];
end
else CWMain.strgCaseAll.Cells[3,i] := CWMain.strgCaseAll.Cells[5,i]
end
else if (Pos(CWMain.strgCaseAll.Cells[3,i],CWMain.strgCaseAll.Cells[5,i]) = 0) and (i <> nA) then
begin
  if (Copy(Trim(CWMain.strgCaseAll.Cells[3,i]),1,6) = 'ข้อมูล') and (Pos(Copy
(CWMain.strgCaseAll.Cells[3,i],7,Length(CWMain.strgCaseAll.Cells[3,i])),CWMain.strgCaseAll.Cells[5,i]
) > 0) then
    CWMain.strgCaseAll.Cells[3,i] := CWMain.strgCaseAll.Cells[5,i]
  else if CWMain.strgCase.Cells[3,i] = CWMain.strgCase.Cells[5,i] then
  else
    CWMain.strgCaseAll.Cells[3,i] := CWMain.strgCaseAll.Cells[3,i] + ' '
+CWMain.strgCaseAll.Cells[5,i];
  end;
  //Output from Output Part
  if CWMain.strgCase.Cells[5,CWMain.strgCase.RowCount-1] <> " then
  begin
    CWMain.strgCaseAll.Cells[5,CWMain.strgCaseAll.RowCount-1] :=
CWMain.strgCase.Cells[5,CWMain.strgCase.RowCount-1];
    if CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] = " then
      CWMain.strgCase.Cells[3,CWMain.strgCase.RowCount-1] :=
CWMain.strgCase.Cells[5,CWMain.strgCase.RowCount-1];
    end;
  // Declare No. of function
  for i:= nA to CWMain.strgCaseAll.RowCount-1 do
  begin
    if nA = 1 then
      num := '1'
    else
      num := InttoStr(StrToInt(Copy(CWMain.strgCaseAll.Cells[0,nA-
1],1,Pos('.',CWMain.strgCaseAll.Cells[0,nA-1])-1)) + 1 );
      CWMain.strgCaseAll.Cells[0,i] := num + '.' + CWMain.strgCaseAll.Cells[0,i];
    end;
  end;
end;
end;

```

รูปที่ ๗-7 มอดูลการสร้างโครงสร้างใหม่และเพิ่มเติมให้โครงสร้างใหม่สมบูรณ์ขึ้น (ต่อ)

```

procedure chkConsist;
var i, j, k, chkData : integer; ft, lt : integer; str1, str2 : string;
begin
  createData; //Check Database
  for i:=1 to DataSRS.strgOut.RowCount-1 do
  begin
    DataSRS.strgOut.Cells[2,i] := '0';
    if (chkDataIn(DataSRS.strgOut.Cells[0,i]) = 1) then
      DataSRS.strgOut.Cells[2,i] := '1'
    else if (Copy(DataSRS.strgOut.Cells[0,i],1,6) = 'ข้อมูล') and (chkAgent
(Copy(DataSRS.strgOut.Cells[0,i],7,Length(DataSRS.strgOut.Cells[0,i]))) = 1) then
      DataSRS.strgOut.Cells[2,i] := '1'
    end;
    chkData := 0;
    if (DataSRS.strgOut.Cells[0,1] <> "") then
      for i:=1 to DataSRS.strgOut.RowCount-1 do
        if (DataSRS.strgOut.Cells[2,i] = '0') and (DataSRS.strgOut.Cells[0,i] <> 'รายงานผล') then
          begin
            CWMain.mmIncon.Lines.Add('ฟังก์ชัน ' + DataSRS.strgOut.Cells[1,i] + ' ข้อมูล: ' +
DataSRS.strgOut.Cells[0,i] + ' ไม่พบการเก็บลงฐานข้อมูล');
            chkData := 1;
          end;
        if (chkData = 0) or (DataSRS.strgOut.Cells[2,DataSRS.strgOut.RowCount-1] = "") then
          CWMain.mmIncon.Lines.Add('ไม่พบความไม่ต้องกันของฐานข้อมูล');
        chkAction;
        createInOut; //Input Output
        // In must Out
        for i:= 1 to CWMain.strgCaseAll.RowCount - 1 do
          begin
            if Copy(CWMain.strgCaseAll.Cells[6,i],1,8) = 'ฟังก์ชัน' then
              for k:= 1 to DataSRS.strgFuncIn.RowCount-1 do
                if (DataSRS.strgFuncIn.Cells[0,k] = CWMain.strgCaseAll.Cells[6,i]) and
(DataSRS.strgFuncIn.Cells[5,k] = 'ฟังก์ชัน ' + CWMain.strgCaseAll.Cells[0,i]) then
                  begin
                    ft := StrToInt(DataSRS.strgFuncIn.Cells[1,k]);
                    while ft <= StrToInt(DataSRS.strgFuncIn.Cells[2,k]) do
                      begin
                        for j:= 1 to DataSRS.strgFuncOut.RowCount-1 do
                          if (DataSRS.strgFuncOut.Cells[0,j] = DataSRS.strgFuncIn.Cells[5,ft]) and
(DataSRS.strgFuncOut.Cells[5,j] = DataSRS.strgFuncIn.Cells[0,ft]) then
                            begin
                              lt := StrToInt(DataSRS.strgFuncOut.Cells[1,j]);
                              while lt <= StrToInt(DataSRS.strgFuncOut.Cells[2,j]) do
                                begin
                                  if Copy(DataSRS.strgFuncIn.Cells[3,ft],1,6) = 'ข้อมูล' then
                                    str1 := Copy(DataSRS.strgFuncIn.Cells[3,ft],7,
Length(DataSRS.strgFuncIn.Cells[3,ft]));
                                  else
                                    str1 := DataSRS.strgFuncIn.Cells[3,ft];
                                  if Copy(DataSRS.strgFuncOut.Cells[3,lt],1,6) = 'ข้อมูล' then
                                    str2 := Copy(DataSRS.strgFuncOut.Cells[3,lt],7,
Length(DataSRS.strgFuncOut.Cells[3,lt]));
                                  else
                                    str2 := DataSRS.strgFuncOut.Cells[3,lt];
                                  if str1 = str2 then
                                    DataSRS.strgFuncIn.Cells[4,ft] := '1';
                                    lt := lt+1;
                                  end;
                                end;
                              break;
                            end;
                          ft := ft+1;
                        end;
                      end;
                    break;
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

รูปที่ จ-8 มอดูลการตรวจสอบความไม่ต้องกัน

```

//Out must In
if Copy(CWMain.strgCaseAll.Cells[2,i],1,8) = 'ฟังก์ชัน' then
  for k:= 1 to DataSRS.strgFuncOut.RowCount-1 do
    if (DataSRS.strgFuncOut.Cells[0,k] = CWMain.strgCaseAll.Cells[2,i]) and
(DataSRS.strgFuncOut.Cells[5,k] = 'ฟังก์ชัน '+ CWMain.strgCaseAll.Cells[0,i]) then
      begin
        ft := StrToInt(DataSRS.strgFuncOut.Cells[1,k]);
        while ft <= StrToInt(DataSRS.strgFuncOut.Cells[2,k]) do
          begin
            for j:= 1 to DataSRS.strgFuncIn.RowCount-1 do
              if (DataSRS.strgFuncIn.Cells[0,j] = DataSRS.strgFuncOut.Cells[5,ft]) and
(DataSRS.strgFuncIn.Cells[5,j] = DataSRS.strgFuncOut.Cells[0,ft]) then
                begin
                  lt := StrToInt(DataSRS.strgFuncIn.Cells[1,j]);
                  while lt <= StrToInt(DataSRS.strgFuncIn.Cells[2,j]) do
                    begin
                      if Copy(DataSRS.strgFuncOut.Cells[3,ft],1,6) = 'ข้อมูล' then
                        str1 := Copy(DataSRS.strgFuncOut.Cells[3,ft],7,Length
(DataSRS.strgFuncOut.Cells[3,ft]))
                      else
                        str1 := DataSRS.strgFuncOut.Cells[3,ft];
                      if Copy(DataSRS.strgFuncIn.Cells[3,lt],1,6) = 'ข้อมูล' then
                        str2 := Copy(DataSRS.strgFuncIn.Cells[3,lt],7,Length
(DataSRS.strgFuncIn.Cells[3,lt]))
                      else
                        str2 := DataSRS.strgFuncIn.Cells[3,lt];
                      if str1 = str2 then
                        DataSRS.strgFuncOut.Cells[4,ft] := '1';
                        lt := lt+1;
                      end;
                    break;
                  end;
                end;
              ft := ft+1;
            end;
          break;
        end;
      end;
    chkData := 0;
    for i:=1 to DataSRS.strgFuncIn.RowCount - 1 do
      if DataSRS.strgFuncIn.Cells[4,i] = '0' then
        begin
          CWMain.mmIncon.Lines.Add(DataSRS.strgFuncIn.Cells[0,i] + ' ข้อมูล: ' +
DataSRS.strgFuncIn.Cells[3,i] + ' เป็นข้อมูลนำออกจาก '+ DataSRS.strgFuncIn.Cells[5,i] + ' แต่ไม่พบการเป็น
ข้อมูลนำเข้า');
          chkData := 1;
        end;
      for i:=1 to DataSRS.strgFuncOut.RowCount - 1 do
        if DataSRS.strgFuncOut.Cells[4,i] = '0' then
          begin
            CWMain.mmIncon.Lines.Add(DataSRS.strgFuncOut.Cells[5,i] + ' ข้อมูล: ' +
DataSRS.strgFuncOut.Cells[3,i] + ' เป็นข้อมูลนำเข้าจาก '+ DataSRS.strgFuncOut.Cells[0,i] + ' แต่ไม่พบการ
เป็นข้อมูลนำออก');
            chkData := 1;
          end;
        if chkData = 0 then
          CWMain.mmIncon.Lines.Add('ไม่พบความไม่ตรงกันของข้อมูลนำเข้าและข้อมูลส่งออกระหว่างฟังก์ชัน');
          chkWordWrite;
        end;
      end;
    end;
  end;

```

รูปที่ จ-8 มอดูลการตรวจสอบความไม่ตรงกัน (ต่อ)

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวจุฑาพัทธ์ ปักซี่ เกิดเมื่อวันที่ 27 พฤศจิกายน พ.ศ. 2521 สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์ จาก คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2542 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2543



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย