

การสร้างชุดคำสั่งปีเพลสำหรับกฎธุรกิจจากตารางการตัดสินใจ

นาย ธนาวุฒิ เอื้อชัยกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2551

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

GENERATING BPEL SCRIPTS FOR BUSINESS RULES FROM A DECISION TABLE

Mr. Thanawut Auechaikul

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2008

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การสร้างชุดคำสั่งพีเพิลสำหรับกฎธุรกิจจากตารางการตัดสินใจ

โดย

นาย ธนาวุฒิ เอื้อชัยกุล

สาขาวิชา

วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศศิริวงษ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.ภัทรชัย ลลิตโรจน์วงศ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์)

ธนาวุฒิ เชื้อชัยกุล : การสร้างชุดคำสั่งบีเพลสำหรับกฎธุรกิจจากตารางการตัดสินใจ. (GENERATING BPEL SCRIPTS FOR BUSINESS RULES FROM A DECISION TABLE) อ. ที่ปรึกษาวิทยานิพนธ์
 หลัก: ผศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 138 หน้า.

งานวิจัยนี้นำเสนอเครื่องมือ และวิธีการที่ช่วยให้นักวิเคราะห์ผู้มีความรู้เกี่ยวกับกฎธุรกิจสามารถสร้าง
 แก๊ซ และจัดการกฎธุรกิจได้ด้วยตัวเอง งานวิจัยนี้เลือกใช้ตารางการตัดสินใจและออกแบบภาษานิยามกฎธุรกิจ
 เพื่อช่วยเหลือนักวิเคราะห์ ตารางตัดสินใจเป็นตารางแบบหน่วยขยายและสามารถทำงานแบบวนซ้ำได้ เครื่องมือ
 บรรณาธิกรณกฎธุรกิจที่ออกแบบเป็นเว็บแอปพลิเคชันสามารถสร้าง แก๊ซ จัดประเภท และตรวจสอบความ
 กำกวมของตารางการตัดสินใจ นอกจากนี้ยังสามารถสร้างตารางการตัดสินใจจากเงื่อนไขที่กำหนดได้

งานวิจัยนี้ออกแบบภาษากำหนดกฎธุรกิจหรือภาษาบีอาร์เอ็มแอลเพื่อกำหนดกฎธุรกิจ ภาษานี้แบ่งเป็น
 3 กลุ่ม คือ กลุ่มแท็กกำหนดข้อเท็จจริง กลุ่มแท็กกำหนดเซตกฎธุรกิจ และกลุ่มแท็กกำหนดกฎธุรกิจ ทั้ง 3 กลุ่ม
 ถูกใช้ในไฟล์ 3 ประเภท คือ ไฟล์ข้อเท็จจริง ไฟล์เซตกฎธุรกิจและไฟล์กฎธุรกิจ ตามลำดับ ไฟล์ข้อเท็จจริงเก็บแผ่น
 แบบเงื่อนไขและการกระทำเพื่อเป็นวัตถุดิบสำหรับสร้างกฎธุรกิจ ไฟล์เซตกฎธุรกิจทำหน้าที่เก็บกฎธุรกิจทั้งหมด
 ในเซตกฎธุรกิจหรือตารางการตัดสินใจที่สร้างขึ้น ไฟล์เซตกฎธุรกิจจะระบุแผ่นแบบเงื่อนไขและการกระทำที่ใช้ในกฎ
 ธุรกิจและค่าของตัวแปรปรับแต่งที่ผู้ใช้กำหนด โดยรับข้อมูลจากผู้ใช้ผ่านเครื่องมือบรรณาธิกรณกฎธุรกิจ

ไฟล์กฎธุรกิจจะทำหน้าที่เก็บข้อมูลเกี่ยวกับไฟล์เซตกฎธุรกิจและไฟล์ข้อเท็จจริงทั้งหมด จากนั้นกฎธุรกิจ
 ทั้งหมดจะถูกแปลงเป็นไฟล์บีเพลและไฟล์บรรยายเว็บเซอร์วิส ซึ่งสามารถนำไปประมวลผลด้วยเครื่องประมวล-
 ผลบีเพลเพื่อให้ตารางการตัดสินใจสามารถทำงานให้บริการในรูปแบบของเว็บเซอร์วิสธุรกิจ ซึ่งช่วยสนับสนุนการ
 ทำงานของกระบวนการธุรกิจอื่นๆ ได้

งานวิจัยนี้ทดสอบด้วยกรณีศึกษาเกี่ยวกับกระบวนการธุรกิจ 2 กรณีศึกษา เพื่อให้ครอบคลุมการทำงานของ
 ของกฎธุรกิจและกระบวนการธุรกิจโดยทั่วไปได้ หลังจากการสร้างกฎธุรกิจแล้วนำไฟล์บีเพลไปทดสอบกับโปรแกรม
 NetBeans IDE 6.1 พบว่ากฎธุรกิจที่ทำงานในรูปแบบเว็บเซอร์วิสสามารถทำงานร่วมกับกระบวนการธุรกิจและเว็บ
 เซอร์วิสอื่นๆ ได้ และสามารถทำงานให้ผลลัพธ์ตามที่กำหนดในกฎธุรกิจได้อย่างถูกต้อง

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....
 สาขาวิชา....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา 2551

4970347221 : MAJOR COMPUTER ENGINEERING

KEY WORD: WEB SERVICES / BUSINESS RULES / BPEL / DECISION TABLES

THANAWUT AUECHAIKUL: GENERATING BPEL SCRIPTS FOR BUSINESS RULES FROM A DECISION TABLE. THESIS PRINCIPAL ADVISOR: ASST. PROF. WIWAT VATANAWOOD, Ph.D., 138 pp.

This research presents a tool and methods that enable business analysts to manipulate business rules by themselves. The research applies a decision table and designs a Business Rules Definition Language (BRDL) to support the business analysts. The decision table is an extended-entry table and can work as iteration loop. Business analysts can define business rules via a web application that can create, modify, categorize and check ambiguity of decision tables. Furthermore, it can generate a decision table from the defined conditions.

This research designs the BRDL to define business rules. The tags in BRDL are categorized into 3 groups i.e. Predefined Facts definition tags, Ruleset definition tags and Business Rules definition tags. These groups are applied in three types of files i.e. Predefined Facts files, Ruleset files and Business Rules files, respectively. The Predefined Facts files specify condition and action templates as material for business rule definition. Meanwhile, the Ruleset files save all business rules from a decision table. The Business Rules files save all information about the Ruleset files and Predefined Facts files. Finally, all business rules are transformed into BPEL files and WSDL files that can run as web services to support business processes.

In this research, two case studies are examined to test the correctness and efficiency of the business rules. The result BPEL files are deployed and tested as web services with NetBeans IDE 6.1. It shows that the web services collaborate with other business processes and web services correctly. The result of web services match with the business rules defined in decision tables.

Department...Computer Engineering..... Student's signature.....
 Field of study...Computer Engineering..... Principal Advisor's signature.....
 Academic year.....2008.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีเพราะความช่วยเหลือของและสนับสนุนจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.ดร.วิวัฒน์ วัฒนาวุฒิ ซึ่งได้สละเวลาให้คำแนะนำปรึกษา เสนอแนะข้อคิดเห็น แนวทางการวิจัย และได้ให้ความช่วยเหลือดูแลด้านต่างๆ เป็นอย่างดี ซึ่งผู้วิจัยขอขอบพระคุณอย่างสูงในความกรุณาเป็นอย่างยิ่ง

ขอขอบพระคุณ รศ.ดร.ธราทิพย์ สุวรรณศาสตร์ ประธานกรรมการสอบวิทยานิพนธ์และกรรมการสอบวิทยานิพนธ์ ผศ.ดร.ภัทรชัย ลลิตโรจน์วงศ์ ผศ.ดร.อาทิตย์ ทองทักษ์ ที่ให้ความกรุณาเสนอคำแนะนำ ข้อคิดเห็น และชี้แนะแนวทางการพัฒนางานวิจัย ตลอดจนจนถึงตรวจสอบและแก้ไขข้อบกพร่องต่างๆ ของวิทยานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ สมาชิกในห้องปฏิบัติการ CGCI ที่คอยให้คำแนะนำ ความช่วยเหลือ และความสุจริตต่อการวิจัย

ขอขอบคุณผู้ที่มีส่วนช่วยเหลือและให้กำลังใจทุกท่านที่ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

ขอขอบคุณทุนอุดหนุนการทำวิทยานิพนธ์ของธนาคารกสิกรไทย จำกัด (มหาชน) และโครงการการวิศวกรรมซอฟต์แวร์แผนใหม่สำหรับวิสาหกิจโดยสถาบันส่งเสริมบริการ ที่อนุเคราะห์ทุนค่าอุปกรณ์ต่างๆ ที่ใช้ในวิทยานิพนธ์

ท้ายที่สุดขอกราบขอบพระคุณบิดา มารดา คุณอา ครอบครั้วและญาติพี่น้องทุกท่าน ที่อุปการะเลี้ยงดู เป็นกำลังใจ สนับสนุนในด้านการศึกษาและด้านต่างๆ ตลอดมา จนวิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	4
1.3 ขอบเขตของการวิจัย.....	5
1.4 คำจำกัดความที่ใช้ในการวิจัย.....	6
1.5 ขั้นตอนและวิธีดำเนินการวิจัย.....	6
1.6 โครงสร้างของวิทยานิพนธ์นี้.....	6
1.7 บทความที่ได้รับการตีพิมพ์.....	7
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	8
2.1 ทฤษฎีที่เกี่ยวข้อง.....	8
2.1.1 กฎธุรกิจ.....	8
2.1.2 เอกซ์เอ็มแอล.....	9
2.1.3 ตารางการตัดสินใจ.....	10
2.1.4 เว็บเซอร์วิส.....	14
2.1.3 ภาษาบรรยายเว็บเซอร์วิส.....	16
2.1.3 เอ็กส์พาท.....	17
2.1.7 กระบวนการธุรกิจ.....	18
2.1.8 ภาษาพีเพิล.....	19
2.1.9 ภาษาโดเมนเฉพาะ.....	20
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	21
2.2.1 งานวิจัย “Hybrid Web Service Composition: Business Processes Meet	

บทที่	หน้า
Business Rules” โดย Anis Charfi และคณะ.....	21
2.2.2 งานวิจัย “Orchestrating Supply Chain Interactions Using Emerging Process Description Language and Business Rules” โดย Costello และคณะ.....	21
2.2.3 งานวิจัย “Web Services Based Execution of Business Rules” โดย R. Schmidt.....	21
2.2.4 งานวิจัย “Integration of the Decision Table Formalism with a Relational Database Environment” โดย J. Vanthienen และคณะ.....	22
2.2.5 งานวิจัย “A Domain Specific Language for Securing Distributed Systems” โดย H. Hamdi และคณะ.....	22
2.2.6 งานวิจัย “MTrans, a DSL for Model Transformation” โดย M. Peltier.....	22
3 การออกแบบระบบ.....	23
3.1 ภาษานิยามกฎธุรกิจ.....	23
3.2 สถาปัตยกรรมระบบ.....	38
3.3 การออกแบบการทำงานของระบบ.....	49
3.4 แผนภาพคลาส.....	56
4 การพัฒนาและการทดสอบ.....	59
4.1 การพัฒนา.....	59
4.2 กรณีทดสอบที่ 1 : การคำนวณเงินทดแทน.....	59
4.2.1 ลักษณะเงินทดแทน.....	60
4.2.2 กระบวนการธุรกิจ.....	60
4.2.3 ตารางการตัดสินใจ.....	63
4.2.4 การนำตารางการตัดสินใจมาใช้ซ้ำ.....	68
4.2.5 การทดสอบความถูกต้อง.....	69
4.3 กรณีทดสอบที่ 2 : การสั่งซื้อสินค้าผ่านเว็บไซต์.....	75
4.3.1 กระบวนการธุรกิจ.....	75
4.3.2 ตารางการตัดสินใจ.....	77
4.3.4 การนำตารางการตัดสินใจมาใช้ซ้ำ.....	79
4.3.3 การทดสอบความถูกต้อง.....	80

บทที่	หน้า
4.4 การทดสอบประสิทธิภาพการให้บริการของเว็บเซอร์วิสภาครัฐกิจ.....	84
4.4.1 สภาพแวดล้อมในการทดสอบ.....	84
4.4.2 การทดสอบเวลาที่ใช้ในทำงานของเว็บเซอร์วิส.....	84
4.4.3 ผลการทดสอบ.....	85
4.5 สรุปผลการทดสอบ.....	86
5. สรุปผลการวิจัยและข้อเสนอแนะ.....	87
5.1 สรุปผลการวิจัย.....	88
5.2 ประโยชน์ที่คาดว่าจะได้รับ.....	88
5.3 ข้อจำกัด.....	88
5.4 ข้อเสนอแนะ.....	88
รายการอ้างอิง.....	89
ภาคผนวก.....	92
ภาคผนวก ก.....	93
ภาคผนวก ข.....	107
ภาคผนวก ค.....	110
ภาคผนวก ง.....	116
ภาคผนวก จ.....	127
ประวัติผู้เขียนวิทยานิพนธ์.....	138

สารบัญตาราง

ตารางที่	หน้า
2.1	ส่วนประกอบของตารางการตัดสินใจ..... ..10
2.2	ตารางการตัดสินใจของกฎการจ่ายค่าสินค้าโดยไม่ใช้เงินสด..... ..11
2.3	ตารางการตัดสินใจแบบลดรูป..... ..11
2.4	ตารางการตัดสินใจแบบหน่วยขยาย..... ..12
2.5	ตารางย่อยเงื่อนไขประเมินว่าลูกค้าเป็นลูกค้าชั้นดีหรือไม่..... ..13
2.6	ตารางย่อยการกระทำการส่งสินค้า..... ..13
2.7	ตารางการตัดสินใจที่มีความกำกวม..... ..14
2.8	ตารางการตัดสินใจที่เงื่อนไขเป็นการออร์..... ..14
3.1	กฎธุรกิจที่สร้างจากเงื่อนไขที่กำหนดโดยผู้ใช้..... ..54
4.1	ผลการทดสอบเวลาที่ใช้ในการเรียกใช้เว็บเซอร์วิสกฎธุรกิจ..... ..84
ก-1	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Fact..... ..93
ก-2	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Action..... ..94
ก-3	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Activity..... ..94
ก-4	รายละเอียดของคุณสมบัติและเมท็อดของคลาส InvokeWSActivity..... ..95
ก-5	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Parameter..... ..95
ก-6	รายละเอียดของคุณสมบัติและเมท็อดของคลาส SubRulesetParameter..... ..96
ก-7	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Variable..... ..96
ก-8	รายละเอียดของคุณสมบัติและเมท็อดของคลาส VariableEnum..... ..97
ก-9	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Rule..... ..97
ก-10	รายละเอียดของคุณสมบัติและเมท็อดของคลาส ValueOfCondition..... ..99
ก-11	รายละเอียดของคุณสมบัติและเมท็อดของคลาส ValueOfAction..... ..100
ก-12	รายละเอียดของคุณสมบัติและเมท็อดของคลาส RuleSet..... ..101
ก-13	รายละเอียดของคุณสมบัติและเมท็อดของคลาส LoopCondition..... ..101
ก-14	รายละเอียดของคุณสมบัติและเมท็อดของคลาส RuleGroup..... ..103
ก-15	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Bpel..... ..103
ก-16	รายละเอียดของคุณสมบัติและเมท็อดของคลาส Category..... ..104
ก-17	รายละเอียดของคุณสมบัติและเมท็อดของคลาส ReturnVariable..... ..104

ตารางที่	หน้า
ก-18 รายละเอียดของคุณสมบัติและเมท็อดของคลาส DtParameter.....	104
ก-19 รายละเอียดของคุณสมบัติและเมท็อดของคลาส PredefinedFacts.....	105
ก-20 รายละเอียดของคุณสมบัติและเมท็อดของคลาส BusinessRules.....	105
ข-1 รายละเอียดของเมท็อดของคลาส RuleSetDAM.....	107
ข-2 รายละเอียดของเมท็อดของคลาส RuleSetDAM.....	108
ข-3 รายละเอียดของเมท็อดของคลาส PredefinedFactsDAM.....	109
ข-4 รายละเอียดของเมท็อดของคลาส BpelDAM.....	109

สารบัญภาพ

รูปที่	หน้า
1.1	การพัฒนากระบวนการธุรกิจในปัจจุบัน.....1
1.2	เว็บเซอร์วิสกฎธุรกิจถูกเรียกใช้จากกระบวนการธุรกิจ Loan Flow BPEL Process 3
2.1	องค์ประกอบของกฎธุรกิจ.....9
2.2	การทำงานของเว็บเซอร์วิส..... 14
2.3	สถาปัตยกรรมเว็บเซอร์วิส..... 15
2.4	การเรียกใช้เว็บเซอร์วิส..... 15
2.5	ตัวอย่างเอกซ์เอ็มแอลสคีมา..... 17
2.6	ตัวอย่างเอกสารเอกซ์เอ็มแอล..... 18
2.7	กระบวนการธุรกิจตรวจสอบใบสั่งซื้อสินค้า..... 19
3.1	ตัวอย่างแผ่นแบบเงื่อนไข..... 24
3.2	ความสัมพันธ์ของตัวแปรที่กำหนดค่าโดยผู้ใช้กับประโยคในรูปภาษาธรรมชาติ..... 25
3.3	โครงสร้างภายในแท็ก condition..... 25
3.4	ตัวอย่างแผ่นแบบการกระทำ..... 26
3.5	โครงสร้างภายในแท็ก action..... 27
3.6	ตัวอย่างแผ่นแบบเงื่อนไขและแผ่นแบบการกระทำ..... 28
3.7	ตัวอย่างแท็ก variable..... 28
3.8	ตัวอย่างแท็ก invoke..... 29
3.9	ตัวอย่างแท็ก return..... 29
3.10	แผ่นแบบการกระทำสำหรับเรียกตารางย่อย..... 30
3.11	ตัวอย่างไฟล์เซตกฎธุรกิจ..... 31
3.12	โครงสร้างภายในแท็ก ruleset ของแผ่นแบบการกระทำทั่วไป..... 33
3.13	ตัวอย่างแท็ก conditions..... 34
3.14	โครงสร้างภายในแท็ก conditions..... 35
3.15	ตัวอย่างแท็ก variableValue ภายในแท็ก action..... 36
3.16	โครงสร้างภายในแท็ก actions..... 37
3.17	ตัวอย่างไฟล์กฎธุรกิจ..... 37
3.18	โครงสร้างภายในแท็ก businessRules..... 38

รูปที่	หน้า
3.19 องค์ประกอบของระบบสำหรับการพัฒนาธุรกิจ.....	39
3.20 เครื่องมือบรรณาธิกรณั้ธุรกิจแสดงตารางการตัดสินใจ.....	39
3.21 เครื่องมือบรรณาธิกรณั้ธุรกิจ.....	40
3.22 ตัวอย่างแผ่นแบบเงื่อนไขในไฟล์ข้อเท็จจริง.....	41
3.23 ตัวอย่างไฟล์เซตธุรกิจของตารางการตัดสินใจ.....	42
3.24 ตัวอย่างเอกซ์เอ็มแอลสคีมาไฟล์บรรยายเว็บเซอร์วิสจากตารางการตัดสินใจ.....	44
3.25 ตัวอย่างไฟล์บีเพลที่ประกอบด้วยข้อความ If-Then.....	45
3.26 ตัวอย่างไฟล์บีเพลที่เป็นการทำงานแบบวนซ้ำ.....	46
3.27 ตัวอย่างไฟล์บีเพลที่มีเรียกใช้เว็บเซอร์วิส.....	48
3.28 แผนภาพยูสเคสแสดงการทำงานของงานวิจัย.....	49
3.29 แผนภาพกิจกรรมการสร้างเซตธุรกิจ.....	50
3.30 แผนภาพกิจกรรมการสร้างธุรกิจ.....	51
3.31 แผนภาพกิจกรรมการสร้างเงื่อนไขในธุรกิจ.....	52
3.32 แผนภาพกิจกรรมการสร้างการกระทำในธุรกิจ.....	52
3.33 แผนภาพกิจกรรมการสร้างธุรกิจจากเงื่อนไขที่กำหนด.....	55
3.34 ความสัมพันธ์ของแพ็กเกจของคลาส.....	56
3.35 ความสัมพันธ์และคุณสมบัติของคลาสในแพ็กเกจ Data Access.....	57
3.36 ความสัมพันธ์และคุณสมบัติของคลาสในแพ็กเกจ Business Logic.....	58
4.1 แผนภาพดีพลอยเมนต์.....	59
4.2 กระบวนการธุรกิจคำนวณหาจำนวนเงินทดแทน.....	61
4.3 ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน.....	63
4.4 แผ่นแบบเงื่อนไขจังหวัดที่ลูกจ้างทำงาน.....	63
4.5 แผ่นแบบการกระทำกำหนดค่าแรงขั้นต่ำ.....	64
4.6 ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้.....	64
4.7 แผ่นแบบการกระทำกำหนดค่าทดแทนจากจำนวนวันที่ไม่สามารถทำงานได้.....	65
4.8 ตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด.....	66
4.9 การส่งพารามิเตอร์และรับข้อมูลระหว่างตารางการตัดสินใจ การกระทำและเว็บเซอร์วิส	

รูปที่	หน้า
ของการกระทำคำนวณค่าทดแทนจากความสูญเสียถาวร.....	67
4.10 การส่งพารามิเตอร์และรับข้อมูลระหว่างผู้เรียกใช้ตารางการตัดสินใจ ตารางการตัดสินใจ และการกระทำของการกระทำคำนวณค่าทดแทนทั้งหมด.....	68
4.11 การเรียกใช้ตารางการตัดสินใจของกระบวนการธุรกิจคำนวณเงินทดแทนและกระบวนการธุรกิจคำนวณค่าสินไหมทดแทน.....	69
4.12 ตัวอย่างข้อความไขพของพารามิเตอร์ของตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน.....	69
4.13 ตัวอย่างข้อความไขพของผลลัพธ์ของตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน.....	70
4.14 ตัวอย่างข้อความไขพของพารามิเตอร์ของตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้.....	70
4.15 ตัวอย่างข้อความไขพของผลลัพธ์ของตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้.....	71
4.16 ตัวอย่างข้อความไขพของพารามิเตอร์ของตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด ..	71
4.17 ตัวอย่างข้อความไขพของผลลัพธ์ของตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด.....	72
4.18 ตัวอย่างข้อความไขพของพารามิเตอร์ของกระบวนการธุรกิจคำนวณหาจำนวนเงินทดแทน.....	73
4.19 ตัวอย่างข้อความไขพของผลลัพธ์ของกระบวนการธุรกิจคำนวณหาจำนวนเงินทดแทน....	73
4.20 กระบวนการธุรกิจการสั่งซื้อสินค้าผ่านเว็บไซต์.....	75
4.21 ตารางการตัดสินใจกำหนดค่าส่งหลัก.....	76
4.22 ตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 1 วัน.....	77
4.23 ความสัมพันธ์ของทั้งตารางการตัดสินใจกำหนดค่าส่งสินค้าทั้ง 4 ตาราง.....	77
4.24 ตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม.....	78
4.25 การใช้ตารางการตัดสินใจร่วมกันของกระบวนการธุรกิจสั่งซื้อสินค้าและกระบวนการธุรกิจคำนวณต้นทุนสินค้า.....	79
4.26 ตัวอย่างข้อความไขพของพารามิเตอร์ของตารางการตัดสินใจกำหนดค่าส่งหลัก.....	79
4.27 ตัวอย่างข้อความไขพของผลลัพธ์ของตารางการตัดสินใจกำหนดค่าส่งหลัก.....	80
4.28 ตัวอย่างข้อความไขพของพารามิเตอร์ของตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม.....	80

รูปที่	หน้า
4.29 ตัวอย่างข้อความชีพของผลลัพธ์ของตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม.....	81
4.30 ตัวอย่างข้อความชีพของพารามิเตอร์ของกระบวนการธุรกิจสั่งซื้อสินค้าผ่านเว็บไซต์.....	82
4.31 ตัวอย่างข้อความชีพของผลลัพธ์ของกระบวนการธุรกิจสั่งซื้อสินค้าผ่านเว็บไซต์.....	82
4.32 กราฟแสดงการเปรียบเทียบระหว่างจำนวนผู้ใช้บริการพร้อมกันและเวลาเฉลี่ยต่อการเรียกใช้เว็บเซอวิส 1 ครั้ง.....	85
ค-1 เอกซ์เอ็มแอลสคีมาของไฟล์ข้อเท็จจริง.....	110
ค-2 เอกซ์เอ็มแอลสคีมาของไฟล์เซตกฎธุรกิจ.....	112
ค-3 เอกซ์เอ็มแอลสคีมาของไฟล์กฎธุรกิจ.....	115
ง-1 ตัวอย่างการสร้างหมวดหมู่ของตารางการตัดสินใจ.....	116
ง-2 ตัวอย่างการสร้างตารางการตัดสินใจ.....	117
ง-3 ตัวอย่างการสร้างตารางการตัดสินใจจากเงื่อนไข.....	117
ง-4 ตัวอย่างตารางการตัดสินใจที่สร้างจากเงื่อนไข.....	118
ง-5 การสร้างและแก้ไขกฎธุรกิจที่ละกฎ.....	119
ง-6 ระบุตัวดำเนินการเปรียบเทียบและค่าของตัวแปรที่เกี่ยวกับเงื่อนไข.....	119
ง-7 การแก้ไขการกระทำด้วยเครื่องมือบรรณาธิกรณกฎธุรกิจ.....	120
ง-8 การกำหนดค่าพารามิเตอร์ของการกระทำด้วยพารามิเตอร์ของตารางการตัดสินใจ.....	121
ง-9 การแสดงค่าพารามิเตอร์ของการกระทำ.....	121
ง-10 การนำผลลัพธ์จากการกระทำมากำหนดค่าให้ผลลัพธ์ของตารางการตัดสินใจ.....	122
ง-11 การเพิ่มการกระทำแบบเรียกตารางย่อยเข้าสู่กฎธุรกิจ.....	122
ง-12 การกำหนดค่าพารามิเตอร์ของการกระทำด้วยพารามิเตอร์ของตาราง.....	122
ง-13 การนำผลลัพธ์จากตารางย่อยมากำหนดค่าให้ผลลัพธ์ของตารางการตัดสินใจ.....	123
ง-14 การกำหนดเงื่อนไขการวนซ้ำของตารางการตัดสินใจ.....	125
ง-15 ตัวอย่างการตรวจสอบความกำกวมของตารางการตัดสินใจ.....	125
ง-16 ตัวอย่างการตรวจสอบความขัดแย้งกันของเงื่อนไขในกฎธุรกิจ.....	126
ง-17 ตัวอย่างการแยกตารางการตัดสินใจ.....	126

บทที่ 1

บทนำ

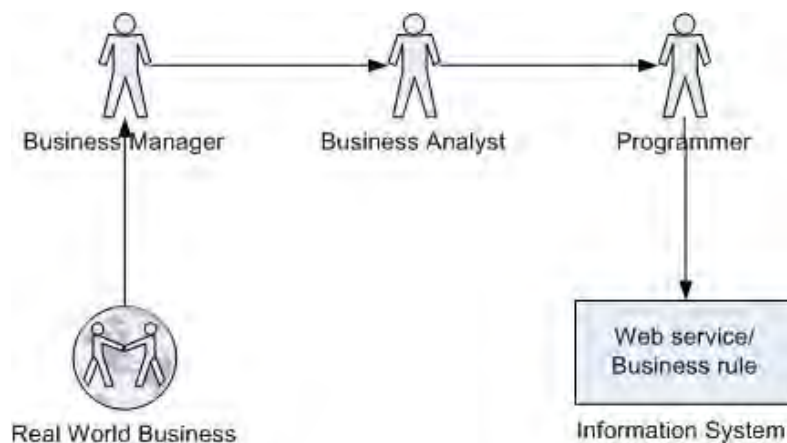
1.1 ความเป็นมาและความสำคัญของปัญหา

องค์กรต่างๆ มีกฎธุรกิจ (Business Rules) มากมาย [1] เช่น กฎการให้ส่วนลดแก่ลูกค้าประเภทต่างๆ กฎการอนุมัติงบประมาณ กฎการอนุมัติประกันภัย เป็นต้น กฎเหล่านี้ล้วนเกี่ยวข้องกับพนักงาน และลูกค้าขององค์กรทั้งสิ้น กฎเหล่านี้เป็นแกนหลักของกระบวนการธุรกิจ (Business Process) ขององค์กร เพราะใช้ในการตัดสินใจ และตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น นอกจากนี้ยังเป็นตัวกำหนด หรือข้อบังคับในการทำงานของพนักงานในองค์กร เพื่อให้ผลลัพธ์ที่ได้จากบุคลากรเหล่านี้สอดคล้องกัน กฎธุรกิจจึงเป็นตัวขับเคลื่อนองค์กรที่สำคัญอย่างยิ่ง ด้วยเหตุนี้ กฎเหล่านี้จึงควรจัดการ และกำหนดเป็นการเฉพาะ โดยแยกกฎออกจากการทำงานอื่นๆ

ปัจจุบันองค์กรต่างๆ ได้พัฒนาระบบสารสนเทศ (Information System) ขึ้นเพื่อสนับสนุนการทำงานขององค์กร หลายอย่างในองค์กรถูกเปลี่ยนให้อยู่ในรูปอิเล็กทรอนิกส์ และระบบอัตโนมัติ (Automatic System) โดยใช้เครื่องคอมพิวเตอร์ ไม่เว้นแม้แต่กฎธุรกิจ เช่น กฎการให้ส่วนลดลูกค้า กฎการอนุมัติเงินกู้ เป็นต้น

อย่างไรก็ตามโดยทั่วไปกฎธุรกิจผูกติดอยู่ในกระบวนการธุรกิจหรือภายในโปรแกรม เช่น คำสั่งจาวา (Java code) เป็นต้น ซึ่งพัฒนาด้วยการเขียนโปรแกรม (Programming)

ในปัจจุบันนี้ [2] ปกติแล้ว ในการพัฒนาเว็บเซอร์วิส และกระบวนการธุรกิจ ผู้จัดการ (Business Manager) ทำหน้าที่กำหนดกระบวนการโดยถ่ายทอดความรู้ผ่านนักวิเคราะห์ (Business Analyst) ไปสู่นักเขียนโปรแกรม เพื่อพัฒนาซอฟต์แวร์สนับสนุนการดำเนินงานทางธุรกิจ นักเขียนโปรแกรมจะพัฒนากฎธุรกิจ และเว็บเซอร์วิสต่างๆ เพื่อดำเนินการในกระบวนการธุรกิจที่ใช้จริง ดังรูปที่ 1.1



รูปที่ 1.1 การพัฒนากระบวนการธุรกิจในปัจจุบัน

จากรูปที่ 1.1 จะเห็นได้ว่าการกำหนดกฎธุรกิจด้วยแนวทางข้างต้นนั้นมีข้อเสีย [3] คือ

- 1) ผู้สร้างกฎธุรกิจในที่นี้คือผู้จัดการ ไม่ได้เป็นผู้เขียนกฎที่ใช้ในระบบด้วยตัวเองโดยตรง ผลที่ตามมาคือกฎที่กำหนดขึ้นอาจไม่ตรงกับความต้องการที่แท้จริงขององค์กร
- 2) การปรับเปลี่ยนแก้ไขกฎธุรกิจค่อนข้างช้า เพราะความรู้เกี่ยวกับกฎธุรกิจถ่ายทอดผ่านหลายบุคคลกว่ากฎธุรกิจไปถึงตัวระบบที่ทำงานจริงๆ มีผลให้องค์กรขาดความคล่องตัวในการรับมือความเปลี่ยนแปลงด้านธุรกิจ
- 3) กฎธุรกิจจะจัดกระจายในระบบสารสนเทศ บางกฎอาจขัดแย้งหรือซ้ำซ้อนกัน
- 4) กฎธุรกิจที่ฝังในตัวโปรแกรมไม่สามารถนำมาใช้ซ้ำได้

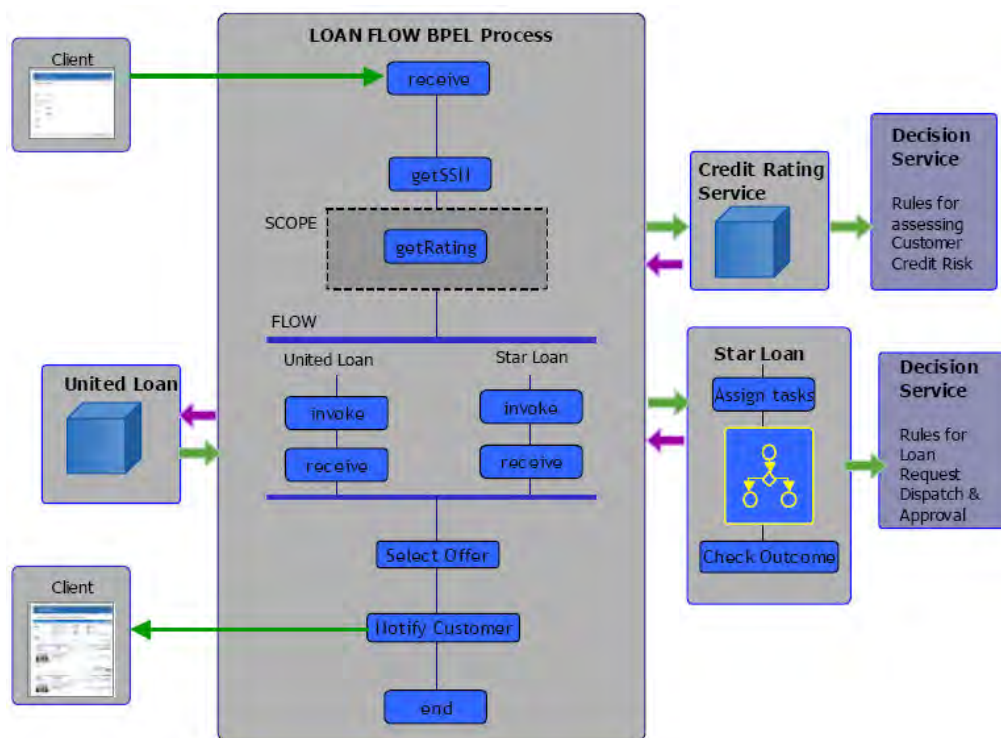
จึงเป็นที่มาของงานวิจัยที่จะลดช่องว่างระหว่างนักวิเคราะห์ซึ่งได้รับความรู้เกี่ยวกับกฎธุรกิจจากผู้จัดการ กับกฎที่ใช้ในระบบสารสนเทศจริงๆ โดยการแยกกฎออกจากตัวโปรแกรม เปลี่ยนให้กฎทำงานในรูปแบบของเว็บเซอวิสซึ่งตัวกฎธุรกิจเองก็สามารถเรียกใช้เว็บเซอวิสอื่นๆ ได้เช่นกัน และให้นักวิเคราะห์เป็นผู้กำหนดกฎด้วยตัวเองโดยใช้ตารางการตัดสินใจและภาษาโดเมนเฉพาะ (Domain-Specific Language) โดยงานวิจัยนี้ได้พัฒนาภาษาโดเมนเฉพาะที่ชื่อภาษานิยามกฎธุรกิจหรือภาษาปีอาร์ดีแอล (BRDL: Business Rules Definition Language)

เพื่อให้กฎธุรกิจสามารถทำงานร่วมกับเว็บเซอวิสได้ จำเป็นต้องมีภาษาหรือคำสั่งที่ใช้ควบคุม หรือเรียกใช้เว็บเซอวิส ภาษาที่มีคุณสมบัติดังกล่าวและเป็นที่ยอมรับในปัจจุบันคือภาษาพีเพิล [4] อย่างไรก็ตามภาษาพีเพิลเป็นเทคโนโลยีที่ค่อนข้างใหม่ นักวิเคราะห์ส่วนใหญ่ไม่มีความรู้ในการเขียนภาษาพีเพิลโปรแกรม ดังนั้นการที่จะให้นักวิเคราะห์เป็นผู้ระบุกฎธุรกิจด้วยตัวเอง จึงต้องใช้ภาษาหรือรูปแบบที่คนทั่วไปเข้าใจได้ง่าย งานวิจัยนี้จึงเลือกใช้ตารางการตัดสินใจ (Decision Table) และภาษาโดเมนเฉพาะเป็นตัวแทนกฎที่นำเสนอแก่นักวิเคราะห์ ภาษาโดเมนเฉพาะเป็นภาษาสำหรับงานเฉพาะอย่าง งานวิจัยจึงพัฒนาภาษาโดเมนเฉพาะเพื่อใช้พัฒนากฎธุรกิจ ส่วนตารางการตัดสินใจเป็นรูปแบบการนำเสนอกฎธุรกิจที่เรียบง่าย เข้าใจง่าย และสามารถใช้ตรวจสอบกฎธุรกิจได้อย่างมีประสิทธิภาพ นักวิเคราะห์สามารถสร้าง และแก้ไขกฎธุรกิจผ่านภาษาโดเมนเฉพาะ จากนั้นตรวจสอบหรือดูกฎธุรกิจที่สร้างขึ้นจากตารางการตัดสินใจได้ การพัฒนากฎธุรกิจด้วยภาษาโดเมนเฉพาะและตารางการตัดสินใจสะดวกกว่าการพัฒนากฎธุรกิจจากภาษาพีเพิลโดยตรง และผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับภาษาพีเพิลแต่อย่างใด

งานวิจัยนี้จึงได้พัฒนาเครื่องมือ เพื่อแปลงกฎธุรกิจที่อยู่ในรูปภาษาโดเมนเฉพาะและตารางการตัดสินใจให้เป็นภาษาพีเพิล เพื่อให้กฎธุรกิจสามารถทำงานร่วมกับกระบวนการธุรกิจ และเว็บเซอวิสอื่นๆ ได้ ด้วยแนวทางที่ว่ามานี้ช่วยให้ นักวิเคราะห์หรือผู้ใช้ทั่วไปซึ่งมีความรู้

เกี่ยวกับกฎธุรกิจขององค์กรสามารถกำหนดกฎธุรกิจได้ด้วยตัวเอง ช่วยลดขั้นตอนในการสร้างหรือแก้ไขกฎ นอกจากนี้กฎธุรกิจที่ทำงานในรูปแบบของเว็บเซอร์วิสนี้ยังสามารถถูกเรียกใช้จากเว็บเซอร์วิส และโปรแกรมประยุกต์อื่นๆ ได้ด้วย จึงช่วยให้กฎธุรกิจที่สร้างขึ้นสามารถใช้ซ้ำ (Reuse) ได้

ขอยกตัวอย่างการแยกกฎธุรกิจออกจากกระบวนการธุรกิจ และการเรียกใช้กฎธุรกิจนั้นในรูปแบบของเว็บเซอร์วิสเพื่อเพิ่มความเข้าใจ ดังแสดงดังรูปที่ 1.2



รูปที่ 1.2 เว็บเซอร์วิสกฎธุรกิจถูกเรียกใช้จากกระบวนการธุรกิจ Loan Flow BPEL Process [5]

กระบวนการธุรกิจ Loan Flow BPEL Process ข้างต้น (ภายในกรอบสี่เหลี่ยมใหญ่ตรงกลางของรูปที่ 1.2) เป็นกระบวนการธุรกิจของระบบนายหน้าเงินกู้ (Loan Broker) มีขั้นตอนสำคัญ [5] ดังนี้

1. Loan Flow BPEL Process รับคำขอกู้จากลูกค้า (Client)
2. วิเคราะห์ความเสี่ยงลูกค้าโดยเรียกเว็บเซอร์วิส Credit Rating Service (ด้านขวาของรูปที่ 1.2) ซึ่งเป็นกฎธุรกิจที่ทำงานในรูปแบบเว็บเซอร์วิส มีกฎธุรกิจ ดังนี้

- ถ้าลูกค้ามีเครดิตน้อยกว่า 500 และมีหนี้ค้างชำระมากกว่า 50,000 บาท และเงินที่ขอกู้มากกว่า 40,000 บาท ให้ความเสี่ยงเป็น “สูง”
- ถ้าลูกค้ามีเครดิตมากกว่า 500 และมีหนี้ค้างชำระน้อยกว่า 50,000 บาท และเงินที่ขอกู้น้อยกว่า 40,000 บาท ให้ความเสี่ยงเป็น “ต่ำ”

3. เรียกใช้เว็บเซอวิสของบริษัทปล่อยกู้ 2 บริษัท คือ United Loan และ Star Loan โดยส่งความเสี่ยงของลูกค้าที่ได้จากขั้นตอนที่แล้วเป็นพารามิเตอร์ เพื่อขอผลการอนุมัติเงินกู้ เว็บเซอวิสทั้ง 2 นี้ก็เป็นกฎธุรกิจซึ่งอาจมีกฎในการอนุมัติที่ต่างกัน โดยขอยกตัวอย่างกฎธุรกิจของบริษัท Star Loan ดังนี้

- ถ้าลูกค้ามีความเสี่ยง “สูง” และเงินที่ขอกู้น้อยกว่า 50,000 บาท ให้ “อนุมัติ” ค่าขอกู้ พร้อมดอกเบี้ย 5
- ถ้าลูกค้ามีความเสี่ยง “สูง” และเงินที่ขอกู้อยู่ระหว่าง 50,000 ถึง 75,000 บาท ให้ “เลื่อน” การอนุมัติเพื่อให้ผู้จัดการตรวจสอบก่อน
- ถ้าลูกค้ามีความเสี่ยง “สูง” และเงินที่ขอกู้มากกว่า 75,000 บาท ให้ “ปฏิเสธ” ค่าขอกู้
- ถ้าลูกค้ามีความเสี่ยง “ต่ำ” และเป็นลูกค้าเก่า ให้ “อนุมัติ” ค่าขอกู้ พร้อมดอกเบี้ย 4% ถ้าลูกค้ามีความเสี่ยง “ต่ำ” ให้ “อนุมัติ” ค่าขอกู้ พร้อมดอกเบี้ย 4.5%

4. Loan Flow BPEL Process เลือกข้อเสนอที่ดีที่สุดจากทั้ง 2 บริษัท และส่งกลับไปยังลูกค้า

เว็บเซอวิสที่กล่าวข้างต้นล้วนเป็นกฎธุรกิจที่ถูกแยกออกมาทำงานอิสระ และถูกเรียกใช้จากกระบวนการธุรกิจ Loan Flow BPEL Process กระบวนการธุรกิจจึงทำหน้าที่ตัวกลางที่คอยควบคุม และประสานงานเว็บเซอวิสต่างๆ ให้ทำงานร่วมกันเพื่อให้ได้ผลลัพธ์ใดๆ ออกมา เว็บเซอวิสเหล่านี้สามารถให้บริการแก่กระบวนการธุรกิจ Loan Flow BPEL Process ของบริษัทนายหน้าใดๆ ได้ หรือให้บริการแก่กระบวนการธุรกิจอื่นๆ ตามแต่ที่องค์กรต้องการ ซึ่งช่วยให้ผู้สร้างกฎธุรกิจสามารถนำกฎเหล่านี้มาใช้ซ้ำได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อสร้างเครื่องมือสำหรับสร้าง และแก้ไขกฎธุรกิจ ที่ทำงานร่วมกับเว็บเซอวิส และกระบวนการธุรกิจอื่นๆ ได้

1.3 ขอบเขตของการวิจัย

- 1) ออกแบบภาษาโดเมนเฉพาะสำหรับสร้างกฎธุรกิจ
- 2) ออกแบบตารางการตัดสินใจสำหรับแสดงกฎธุรกิจที่สร้างจากภาษาโดเมนเฉพาะ
- 3) ภาษาโดเมนเฉพาะมีลักษณะดังนี้
 - 3.1) ภาษาโดเมนเฉพาะสามารถแก้ไขกฎธุรกิจที่สร้างจากภาษาในข้อ 1
 - 3.2) สามารถใช้ตัวปฏิบัติการแอนด์ (AND) และตัวปฏิบัติการออร์ (OR) ได้ตัวหนึ่งเพียงตัวเดียวในเงื่อนไขของกฎธุรกิจ
 - 3.3) ภาษาสนับสนุนการใช้ Nested-If
 - 3.4) ภาษาสามารถกำหนดลำดับของการกระทำได้
 - 3.5) ภาษาสามารถแบ่งกลุ่มกฎ
 - 3.6) ภาษาแสดงในรูปภาษาอังกฤษ
- 4) ตารางการตัดสินใจมีลักษณะดังนี้
 - 4.1) ตารางการตัดสินใจสามารถแสดงกฎธุรกิจที่สร้างจากภาษาโดเมนเฉพาะได้
 - 4.2) เป็นตารางแบบหน่วยขยาย
 - 4.3) ตารางการตัดสินใจสามารถอ้างถึงตารางย่อยได้
 - 4.4) ตารางการตัดสินใจสามารถถูกแบ่ง (split) เป็น 2 ตารางได้
 - 4.5) ตารางการตัดสินใจสามารถกำหนดลำดับของการกระทำได้
 - 4.6) ตารางการตัดสินใจแบ่งกลุ่มกฎได้
 - 4.7) ตารางการตัดสินใจสามารถตรวจสอบความกำกวมของตารางได้
 - 4.8) ตารางการตัดสินใจแสดงข้อมูลในรูปภาษาอังกฤษ
- 5) ตัวแปรในเงื่อนไขต้องเป็นประเภทสายอักขระ หรือจำนวน
- 6) ตัวแปรในเงื่อนไขประเภทจำนวนสามารถกำหนดพิสัยได้
- 7) เงื่อนไขของกฎธุรกิจไม่สามารถแบ่งเป็นเงื่อนไขย่อยได้อีก หรือเป็นเงื่อนไขเดี่ยว
- 8) กฎธุรกิจที่สร้างขึ้นสามารถมีการวนซ้ำ (Iteration) ได้
- 9) กฎธุรกิจต้องเป็นกฎที่สามารถเขียนในรูป IF-THEN ได้ คือเป็นกฎที่มีทั้งเงื่อนไข และการกระทำ
- 10) แปลงกฎธุรกิจที่สร้างจากภาษาในข้อ 1 ให้อยู่ในรูปภาษาพีเพิล ที่พร้อมนำไปควบคุมการทำงานร่วมกันของเว็บเซอร์วิส
- 11) ทดสอบการทำงานของภาษาโดเมนเฉพาะและตารางการตัดสินใจ กับกรณีศึกษา 2 กรณี

1.4 คำจำกัดความที่ใช้ในการวิจัย

เซตกฎธุรกิจ (Ruleset) คือ กลุ่มของกฎธุรกิจที่อยู่ในตารางการตัดสินใจเดียวกัน ความจริงแล้วตารางการตัดสินใจเป็นรูปแบบหนึ่งของการแสดงเซตกฎธุรกิจ (รายละเอียดเกี่ยวกับตารางการตัดสินใจอธิบายในบทที่ 2 หัวข้อ 2.1.3) และงานวิจัยนี้แสดงเซตกฎธุรกิจในรูปของตารางการตัดสินใจเท่านั้น ดังนั้นในงานวิจัยนี้ตั้งแต่บทที่ 3 เป็นต้นไปจะใช้คำว่า “เซตกฎธุรกิจ” และ “ตารางการตัดสินใจ” ในความหมายเดียวกัน และใช้แทนกันและกัน

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาทฤษฎี และความรู้ที่เกี่ยวข้องโดยรวม
2. ศึกษาวิธีการนำเสนอกฎธุรกิจให้เข้าใจง่าย และใกล้เคียงภาษามนุษยชาติ
3. ออกแบบตารางการตัดสินใจ การแสดงกฎบนตารางการตัดสินใจ แนวทางการสร้างกฎจากภาษาโดเมนเฉพาะ และการแปลงข้อมูลจากภาษาโดเมนเฉพาะ ให้อยู่ในรูปแบบภาษาบีเพล
4. ออกแบบไฟล์เอกซ์เอ็มแอลเพื่อเก็บเงื่อนไข และการกระทำสำหรับนำมาสร้างตารางการตัดสินใจ และแสดงกฎธุรกิจในภาษาโดเมนเฉพาะ
5. พัฒนาเครื่องมือเพื่อช่วยในการสร้าง และแก้ไขกฎด้วยภาษาโดเมนเฉพาะ เพื่อนำกฎที่ได้ไปสร้างเป็นเว็บเซอร์วิส เพื่อสนับสนุนกระบวนการธุรกิจต่อไป
6. ทดสอบการทำงาน
7. วิเคราะห์การทำงาน แก้ไข ปรับปรุง
8. สรุปผล และเรียบเรียงวิทยานิพนธ์

1.6 โครงสร้างของวิทยานิพนธ์นี้

วิทยานิพนธ์ฉบับนี้มีทั้งหมด 5 บท บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย คำจำกัดความที่ใช้ในการวิจัย และขั้นตอนและวิธีดำเนินการวิจัย ดังที่ได้กล่าวไปแล้ว สำหรับบทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง บทที่ 3 กล่าวถึงสถาปัตยกรรมของระบบที่วิจัย การออกแบบและการทำงานของระบบ และแผนภาพคลาส บทที่ 4 เป็นการการพัฒนา การทดสอบและผลการทดสอบ และบทสุดท้ายบทที่ 5 เป็นการสรุปผลการวิจัยและข้อเสนอแนะ

1.7 บทความที่ได้รับการตีพิมพ์

- บทความที่ได้รับการตีพิมพ์ในงานประชุมวิชาการมี 2 บทความ คือ บทความเรื่อง “A Development of Business Rules with Decision Tables for Business Processes” และบทความเรื่อง “Generating BPEL Scripts for Business Rules from a Decision Table and Domain-Specific Language” รายละเอียดแสดงในภาคผนวก จ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 กฎธุรกิจ (Business Rules) [1]

กฎธุรกิจ หมายถึง นโยบาย ขั้นตอน หรือคำจำกัดความ ที่เกี่ยวกับวิธีการที่องค์กรปฏิบัติ ต่อลูกค้า และหุ้นส่วน กฎธุรกิจแทนความรู้ (Knowledge) เกี่ยวกับธุรกิจขององค์กร โดยทั่วไปแล้ว องค์กรต่างๆ ไม่ได้เก็บรักษากฎธุรกิจในรูปแบบที่ชัดเจน แต่อยู่ในรูปของเอกสารต่างๆ เช่น หนังสือสัญญา แผนการตลาด นโยบายในการกำหนดราคา และข้อตกลง หรือเอกสารทางกฎหมายต่างๆ เป็นต้น นอกจากนี้กฎธุรกิจยังอยู่ในรูปโปรแกรมได้ด้วย

กฎธุรกิจแบ่งได้เป็น 3 ส่วนตามรูปที่ 2.1 ดังนี้

1) คำ (Term) คือ คำนาม หรือวลีนามที่อยู่ในภาษามนุษย์ คำสามารถกำหนดในรูปแบบดังนี้

- ความคิด (Concept) เช่น ลูกค้าย เป็นต้น
- คุณสมบัติของความคิด (Property) เช่น เพศของลูกค้า เป็นต้น
- ค่า (Value) เช่น หญิง/ชาย เป็นต้น
- กลุ่มของค่า (Value set) เช่น วันหยุด (เสาร์ อาทิตย์) เป็นต้น

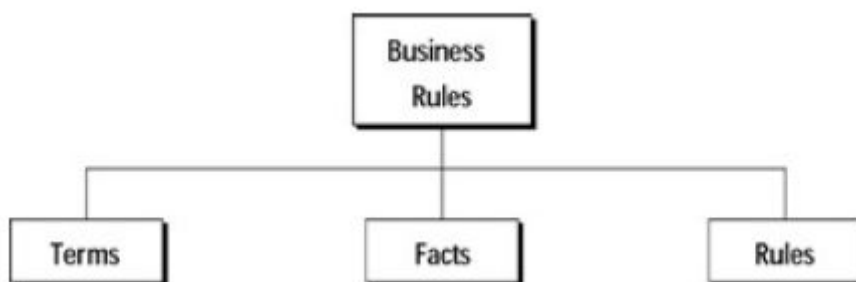
2) ข้อเท็จจริง (Fact) คือ ประโยคที่เชื่อมค่าเข้าไว้ด้วยกัน โดยใช้คำกริยา หรือบุพบท เช่น ลูกค้าสั่งสินค้า เป็นต้น ข้อเท็จจริงแสดงความสัมพันธ์ระหว่างค่า

3) กฎ (Rule) คือ ข้อความเกี่ยวกับแนวทางการตัดสินใจ หรือข้อกำหนดต่างๆ ในธุรกิจ มีเป้าหมายเพื่อควบคุม และแนะนำการทำงานของระบบธุรกิจขององค์กร รวมทั้งช่วยควบคุม และเป็นแนวทางในการตัดสินใจของพนักงานด้วย กฎทุกกฎมีพื้นฐานมาจากข้อเท็จจริง และส่วนใหญ่สามารถเขียนได้ในรูป IF-THEN

ถ้า <เงื่อนไขเกี่ยวกับข้อเท็จจริง> แล้ว <การกระทำ>

(IF <condition involving facts> THEN <Actions>)

การกระทำ (Action) ซึ่งอาจเป็นการทำงานบางอย่าง หรือการคืนค่าผลลัพธ์ออกมา จะเกิดขึ้นเมื่อเงื่อนไขเป็นจริง



รูปที่ 2.1 องค์ประกอบของกฎธุรกิจ [1]

กฎธุรกิจแบ่งเป็น 4 ประเภท [6] ดังนี้

1) กฎเงื่อนไขบังคับ (Constraint Rule) เป็นข้อความแสดงสภาพการณ์ซึ่งต้องทำโดยไม่มีเงื่อนไข หรือทางเลือก เช่น ลูกค้าต้องใส่รหัสอย่างน้อย 6 ตัว เป็นต้น

2) กฎการอนุญาต (Action Enabler Rule) เป็นข้อความที่ตรวจสอบเงื่อนไข และทำงานบางอย่างถ้าเงื่อนไขเป็นจริง เช่น ถ้าสินค้าไม่พอ ให้แจ้งเตือนกลับไปยังลูกค้า เป็นต้น การกระทำของกฎประเภทนี้ จะไม่มีผลต่อข้อเท็จจริงเดิมของระบบ จากตัวอย่างกฎที่ยกมาข้างต้น การกระทำไม่มีการเปลี่ยนข้อมูลลูกค้า สินค้า หรือรายการสั่งสินค้า แต่อย่างไร การกระทำส่วนใหญ่อยู่ในรูปการส่งข้อความ (Message) กลับไป

3) กฎการคำนวณ (Computation Rule) เป็นข้อความที่บอกขั้นตอนวิธี (Algorithm) เพื่อให้ได้มาซึ่งค่าของค่า เช่น ราคารวมของรายการสั่งสินค้า คือ ผลรวมของราคาสินค้าทั้งหมดในรายการบวกภาษี 7% เป็นต้น

4) กฎการอนุมาน (Inference Rule) เป็นข้อความที่ตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริงจะกระทำบางอย่างซึ่งมีผลให้เกิดข้อเท็จจริงใหม่ในระบบ เช่น ถ้าลูกค้าเป็นลูกค้าชั้นดี จะได้รับส่วนลด 20% เป็นต้น จะเห็นว่าการให้ส่วนลดลูกค้า ก่อให้เกิดการเปลี่ยนแปลงของราคารวมของรายการสั่งสินค้า

2.1.2 เอกซ์เอ็มแอล (The Extensible Markup Language: XML) [7]

เอกซ์เอ็มแอลเป็นภาษามาร์คอัพ (Markup) อยู่ในรูปข้อความ (Text) ข้อดีของเอกซ์เอ็มแอล คือสามารถกำหนดแท็ก (Tag) ได้อย่างอิสระแล้วทำการส่งเอกซ์เอ็มแอลชุดนี้ไปประมวลผลยังแอปพลิเคชันใด ๆ ที่สามารถใช้ข้อมูลในเอกซ์เอ็มแอลนี้

เอกซ์เอ็มแอลประกอบด้วยกลุ่มแท็กซึ่งใช้เพื่อสร้าง และบรรยายข้อมูล เอกซ์เอ็มแอลนั้นสามารถกำหนดแท็กได้อย่างอิสระ และอยู่ในรูปแบบที่มนุษย์ หรือโปรแกรมประยุกต์ (Application) เข้าใจได้ง่าย ดังตัวอย่างหน้าถัดไป


```
<?xml version="1.0" encoding="UTF-8" ?>
<address category="friend">
  <name>Bill Frankenfiller</name>
  <street>3243 West 1st Ave.</street>
  <city>Madison</city>
  <state>WI</state>
  <zip>53591</zip>
</address>
```

เอกซ์เอ็มแอลเป็นปัจจัยสำคัญที่ทำให้การใช้งานเว็บเซอร์วิสไม่ขึ้นกับระบบซอฟต์แวร์ เนื่องจากเอกซ์เอ็มแอลถูกกำหนดให้เป็นมาตรฐานที่องค์กรต่างๆ ยอมรับ และอยู่ในรูปข้อความ ซึ่งแอปพลิเคชันสามารถนำไปประมวลผลได้ง่าย

2.1.3 ตารางการตัดสินใจ (Decision Tables) [8]

ตารางการตัดสินใจ เป็นตารางที่แสดงเซตกฎธุรกิจ หรือกฎธุรกิจที่เกี่ยวข้องกัน ตารางการตัดสินใจเป็นรูปแบบหนึ่งของการแสดงเซตกฎธุรกิจ ตารางการตัดสินใจประกอบด้วยแถวและคอลัมน์ แบ่งเป็น 4 ส่วน ดังตารางที่ 2.1

ตารางที่ 2.1 ส่วนประกอบของตารางการตัดสินใจ

เงื่อนไข (Conditions)	หน่วยเงื่อนไข (Condition Entries)
การกระทำ (Actions)	หน่วยการกระทำ (Actions Entries)

ช่องบนซ้ายเป็นเงื่อนไขของกฎที่กำหนดการกระทำ ช่องบนขวาเป็นค่าของเงื่อนไขด้านซ้าย ช่องล่างซ้ายเป็นการกระทำต่างๆ ที่จะเกิดขึ้นตามเงื่อนไขที่กำหนด และช่องล่างขวาเป็นการเลือกการกระทำที่เกิดขึ้นเมื่อมีเงื่อนไขตามด้านบน ตัวอย่างตารางที่ 2.2 เป็นกฎการจ่ายค่าสินค้าโดยไม่ใช้เงินสด

จากตารางที่ 2.2 ถ้าราคาสินค้ารวมต่ำกว่า 50 เหรียญ และลูกค้าจ่ายเงินด้วยเช็ค พนักงานรับจ่ายเงินสามารถรับเช็คจากลูกค้า และบันทึกเข้าเครื่องได้ทันที แต่ถ้าจ่ายด้วยบัตรเครดิตต้องตรวจสอบจากฐานข้อมูลท้องถิ่นก่อน

จะเห็นได้ว่าตารางการตัดสินใจช่วยให้ผู้ใช้สามารถเข้าใจกฎธุรกิจที่ซับซ้อนได้ง่ายขึ้น โดยแสดงความสัมพันธ์ของเงื่อนไขและการกระทำของแต่ละกฎในรูปแบบแถวและคอลัมน์

ตารางที่ 2.2 ตารางการตัดสินใจของกฎการจ่ายค่าสินค้าโดยไม่ใช้เงินสด

เงื่อนไข	หน่วยเงื่อนไข			
ราคาสินค้ารวมต่ำกว่า 50 เหรียญ	จริง	จริง	เท็จ	เท็จ
จ่ายด้วยเช็ค	จริง	เท็จ	จริง	เท็จ
จ่ายด้วยบัตรเครดิต	เท็จ	จริง	เท็จ	จริง
การกระทำ	หน่วยการกระทำ			
พนักงานรับเงินได้ทันที	X	-	-	-
ตรวจสอบจากฐานข้อมูลท้องถิ่น	-	X	-	-
เรียกผู้จัดการร้าน	-	-	X	-
ตรวจสอบจากฐานข้อมูลบัตรเครดิต	-	-	-	X

ตารางการตัดสินใจแบ่งเป็น 2 ประเภท [9] ดังนี้

1) ตารางการตัดสินใจแบบขยาย (Expanded Decision Table) คือ ทุกคอลัมน์ของเงื่อนไขมีการระบุค่าชัดเจน เช่น จริง หรือเท็จ เป็นต้น ดังตารางที่ 2.2

2) ตารางการตัดสินใจแบบลดรูป (Contraction Decision Table) คือ บางคอลัมน์ที่มีการกระทำที่เหมือนกันจะถูกลดรูป หรือรวมกัน เงื่อนไขที่ไม่มีผลต่อการกระทำจะอยู่ในรูปเครื่องหมาย “-” ซึ่งหมายถึง ไม่สนใจค่าของเงื่อนไขนั้น (Don't care) ดังตารางที่ 2.3

ตารางที่ 2.3 ตารางการตัดสินใจแบบลดรูป [9]

เงื่อนไข	หน่วยเงื่อนไข				
Credit Limit ?	Ok		Not Ok		
>>CustomerGood	-		Y		N
Stock Sufficient ?	Y	N	Y	N	-
การกระทำ	หน่วยการกระทำ				
>>ExecuteOrder	X	-	X	-	-
Refuse Order	-	-	-	-	X
Put On Waiting List	-	X	-	X	-

ถ้าแบ่งตารางการตัดสินใจตามลักษณะของค่าในหน่วยเงื่อนไขและหน่วยการกระทำแล้วสามารถแบ่งได้ 3 แบบ [10] ดังนี้

1) ตารางแบบหน่วยจำกัด (Limited-entry table) คือ ตารางที่ในหน่วยเงื่อนไขมีเฉพาะค่า "จริง" (เงื่อนไขเป็นจริง) "เท็จ" (เงื่อนไขเป็นเท็จ) และ "-" (ไม่สนใจเงื่อนไขนั้น) และในหน่วยการกระทำมีเฉพาะค่า "X" (ทำการกระทำนั้น) และ "-" (ไม่ทำการกระทำนั้น) ดังตารางที่ 2.2

2) ตารางแบบหน่วยขยาย (Extended-entry table) คือ ตารางที่ในหน่วยเงื่อนไขและในหน่วยการกระทำสามารถมีค่าใดๆ ได้ โดยขึ้นอยู่กับประเภทของข้อมูลของเงื่อนไขและการกระทำ เช่น เงื่อนไขคือประเภทลูกค้า ค่าในหน่วยเงื่อนไขต้องเป็นข้อมูลประเภทสายอักขระ (String) หรือถ้าเงื่อนไขคือจำนวนสินค้าที่สั่ง ค่าในหน่วยเงื่อนไขต้องเป็นข้อมูลประเภทจำนวนเต็ม (Integer) เป็นต้น ดังตารางที่ 2.4

3) ตารางแบบผสม (Mixed-entry table) คือ ตารางที่มีลักษณะของตารางทั้ง 2 แบบข้างต้นอยู่ในตารางเดียวกัน นั่นคือตารางที่มีบางหน่วยเงื่อนไขและบางหน่วยการกระทำมีค่าแบบจำกัด และมีบางหน่วยเงื่อนไขและบางหน่วยการกระทำมีค่าแบบขยาย

ตารางที่ 2.4 ตารางการตัดสินใจแบบหน่วยขยาย

เงื่อนไข	หน่วยเงื่อนไข					
Customer type	bookstore	bookstore	library	library	library	library
Order size	>=6	<6	>=50	20-49	6-19	<6
การกระทำ	หน่วยการกระทำ					
Allow ?% Discount	25	0	15	10	5	0

นอกจากนี้ ตารางการตัดสินใจอาจอ้างอิงถึงตารางย่อย (Subtable) ได้ ตารางย่อยคือ ตารางการตัดสินใจปกติทั่วไป แต่เมื่อถูกเรียกใช้จากตารางอื่นจะถูกมองเป็นตารางย่อย ดังตารางที่ 3 ส่วนที่อ้างอิงถึงตารางย่อยจะมีเครื่องหมาย ">>" จากตารางที่ 3 อ้างอิงถึงตารางย่อย CustomerGood และ ExecuteOrder

ตารางย่อยมี 2 ประเภท [9] คือ

- 1) ตารางย่อยเงื่อนไข คือ ตารางหาค่าของผลลัพธ์ของเงื่อนไข เช่น ตารางที่ 2.5 ที่ช่วยประเมินว่าลูกค้าเป็นลูกค้าชั้นดีหรือไม่
- 2) ตารางย่อยการกระทำ คือ ตารางที่ระบุการกระทำอื่นๆ เพิ่มเติม โดยสามารถมีเงื่อนไขได้เช่นเดียวกับตารางการตัดสินใจทั่วไป เช่น ตารางที่ 2.6

ตารางที่ 2.5 ตารางย่อยเงื่อนไขประเมินว่าลูกค้าเป็นลูกค้าชั้นดีหรือไม่ [9]

เงื่อนไข	หน่วยเงื่อนไข			
	< 1 Year		>=1 Year	
Age Of Account ?	< 1 Year		>=1 Year	
Travel Distance ?	<50	50-100 OR >100	<50 OR 50-100	>100
การกระทำ	หน่วยการกระทำ			
CustomerGood:=N	X	-	X	-
CustomerGood:=Y	-	X	-	X

ตารางที่ 2.6 ตารางย่อยการกระทำการสั่งซื้อสินค้า [9]

เงื่อนไข	หน่วยเงื่อนไข				
	Q<10	10<=Q<15			Q>=15
Quantity Ordered ?	Q<10	10<=Q<15			Q>=15
Travel Distance	-	D<50	50<=D<100	D>=100	-
การกระทำ	หน่วยการกระทำ				
No Discount	X	-	-	-	-
Discount 2%	-	-	-	X	-
Discount 2%	-	-	X	-	-
Railway Transport	-	-	-	-	X
Road Transport	X	X	X	X	-
Bill Type A	X	X	X	X	-
Bill Type B	-	-	-	-	X

ถ้าแบ่งตารางการตัดสินใจตามความกำกวมสามารถแบ่งได้เป็น 2 ประเภท [9] คือ

1. ซิงเกิลฮิต (Single hit) - ทุกคอลัมน์ไม่มีส่วนร่วมกัน (Disjoint) ไม่กำกวม นั่นคือทุกค่าที่เป็นไปได้ของเงื่อนไขในแต่ละคอลัมน์ไม่ซ้ำกันเลย
2. มัลติเพิลฮิต (Multiple hit) - มีเงื่อนไขเหลื่อมกันระหว่างคอลัมน์ (Overlap) กำกวม ดังตารางที่ 2.7 ถ้า Age เท่ากับ 5 เงื่อนไขของทั้ง 2 คอลัมน์จะเป็นจริง ดังนั้นจึงเกิดความกำกวมว่าจะต้องทำการกระทำของคอลัมน์ใด

ตารางที่ 2.7 ตารางการตัดสินใจที่มีความกำกวม

Age	<10	<20
Action1	1	2
Action2	-	1
Action3	2	-

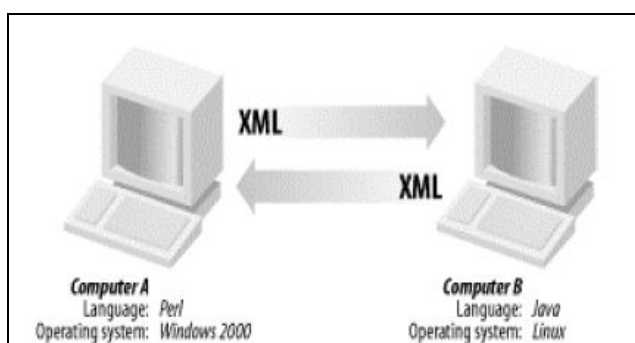
โดยปกติแล้วเงื่อนไขในแต่ละคอลัมน์ของตารางการตัดสินใจจะแอนด์กัน แต่ถ้ามีกฎที่เงื่อนไขเป็นการออร์ การแสดงกฎแบบนี้ในตารางการตัดสินใจต้องแยกกฎเป็น 2 คอลัมน์ แต่ละคอลัมน์จะกำหนดค่าที่เงื่อนไข ส่วนค่าของเงื่อนไขที่เหลือจะไม่สนใจ เช่นมีกฎว่า ถ้าราคาสินค้ารวมมากกว่า 1,000 บาท หรือเป็นสมาชิก ให้ส่วนลด 10% กฎนี้จะแสดงได้ดังตารางที่ 2.8

ตารางที่ 2.8 ตารางการตัดสินใจที่เงื่อนไขเป็นการออร์

ราคาสินค้ารวม	>1000	-
เป็นสมาชิก	-	True
Discount 10%	X	X
Discount 0%	-	-

2.1.4 เว็บเซอร์วิส (Web Services) [7]

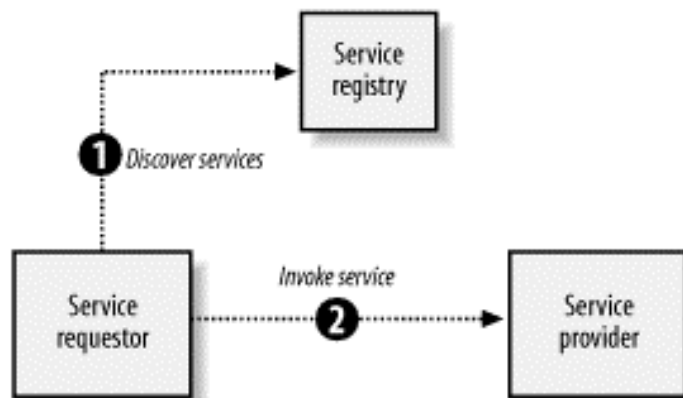
เว็บเซอร์วิสคือ โปรแกรมประยุกต์ หรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่งผ่านอินเทอร์เน็ตในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากโปรแกรมประยุกต์อื่นๆ ในรูปแบบอาร์พีซี (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เพื่อในการแลกเปลี่ยนคือ เอกซ์เอ็มแอล ทำให้เครื่องลูกข่ายสามารถเรียกใช้ส่วนโปรแกรม (Component) ใด ๆ ก็ได้ในแพลตฟอร์ม (platform) ใด ๆ ก็ได้ ดังรูปที่ 2.2 และสามารถทำงานบนโปรโตคอลหลายประเภท เช่น HTTP SMTP เป็นต้น



รูปที่ 2.2 การทำงานของเว็บเซอร์วิส [11]

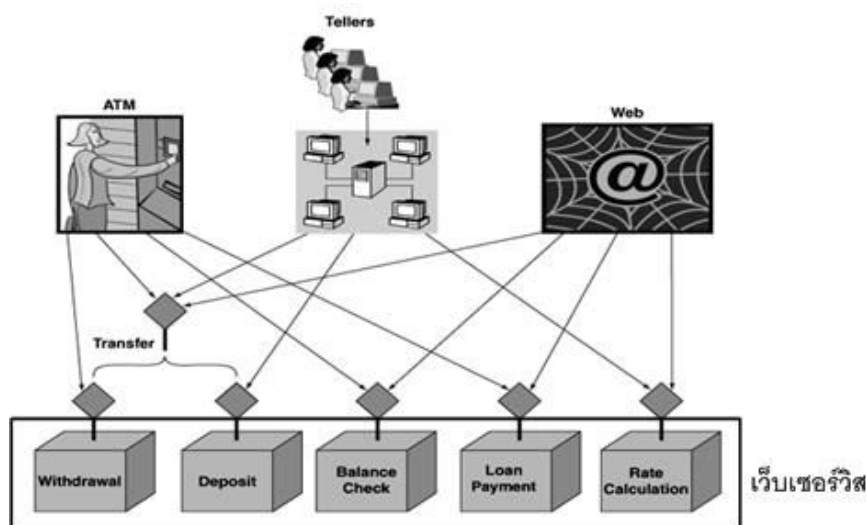
สถาปัตยกรรมเว็บเซอร์วิส (Web Services Architecture) ประกอบด้วย 3 ส่วนตามรูปที่ 2.3 ดังนี้ [11]

- 1) ผู้ให้บริการ (Service provider) มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจากผู้ใช้บริการที่เรียกเข้ามาขอใช้
- 2) ผู้ใช้บริการ (Service requester) เป็นผู้ร้องขอบริการจากผู้ให้บริการผ่านอินเทอร์เน็ต และการส่งข้อความเอกซ์เอ็มแอล
- 3) ส่วนผู้ดูแลบริการ (Service registry) เป็นที่รวบรวมข้อมูลเกี่ยวกับเว็บเซอร์วิส โดย ผู้ให้บริการจะนำข้อมูลเกี่ยวกับเว็บเซอร์วิสมาเก็บที่สำนักทะเบียน ส่วนผู้ให้บริการจะค้นหาเว็บเซอร์วิสที่ต้องการ และนำข้อมูลเกี่ยวกับเว็บเซอร์วิสไปเรียกใช้เว็บเซอร์วิสนั้น



รูปที่ 2.3 สถาปัตยกรรมเว็บเซอร์วิส [11]

ประโยชน์ที่สำคัญของเว็บเซอร์วิส คือ การนำกลับมาใช้ใหม่ได้โดยไม่ขึ้นกับระบบซอฟต์แวร์ที่เรียกใช้หรือระบบซอฟต์แวร์ที่มันทำงานอยู่ ตัวอย่างเช่น ในระบบธนาคาร ลูกค้าอาจใช้บริการผ่านพนักงานธนาคารซึ่งใช้แอปพลิเคชันจาวาหรือทำผ่านตู้เอทีเอ็มหรือผ่านเว็บแอปพลิเคชัน ดังรูปที่ 2.4 ถึงแม้วิธีการใช้บริการจะต่างกัน แต่สามารถเรียกใช้เว็บเซอร์วิส (กล่องสี่เหลี่ยมด้านล่างของรูปที่ 2.4) เดียวกันได้ เนื่องจากเว็บเซอร์วิสไม่ขึ้นกับระบบซอฟต์แวร์ที่เรียกใช้ [12]



รูปที่ 2.4 การเรียกใช้เว็บเซอร์วิส [12]

2.1.5 ภาษาบรรยายเว็บเซอร์วิส (Web Services Description Language: WSDL) [11]

ภาษาบรรยายเว็บเซอร์วิส คือ ภาษาเอกซ์เอ็มแอลที่บรรยายลักษณะของเว็บเซอร์วิส เป็นเหมือนสัญญาหรือข้อตกลงระหว่างผู้ใช้บริการกับผู้ให้บริการ ภาษาบรรยายเว็บเซอร์วิสมีข้อมูลที่สำคัญดังนี้

- 1) ชื่อเว็บเซอร์วิส
- 2) การดำเนินการ (Operation) ของเว็บเซอร์วิส เป็นบริการที่เว็บเซอร์วิสจัดให้แก่ผู้ใช้บริการ ใน 1 เว็บเซอร์วิสอาจมีการดำเนินการหลายอัน เช่น เว็บเซอร์วิสระบบคลังสินค้า (Inventory system web service) อาจมีการดำเนินการ เช่น เช็คปริมาณสินค้า เช็คราคาสินค้า กำหนดจำนวนสินค้า เป็นต้น ซึ่งผู้ใช้บริการเว็บเซอร์วิสต้องระบุให้ชัดเจนในแต่ละครั้งที่เรียกใช้เว็บเซอร์วิสว่า จะใช้การดำเนินการใด
- 3) ชนิด และลักษณะข้อมูลที่ใช้สื่อสารกัน กำหนดด้วยเอกซ์เอ็มแอลสคีมา (XML schema) เป็นกำหนดลักษณะของพารามิเตอร์และผลลัพธ์ของเว็บเซอร์วิส ซึ่งอาจเป็นประเภทข้อมูลที่ซับซ้อน (Complex type) ซึ่งประกอบด้วยข้อมูลอิลิเมนต์ (Element) ดังตัวอย่างสคีมาดังรูปที่ 2.5 พารามิเตอร์เป็นประเภท discount_book_rules_ParamType ซึ่งประกอบด้วยอิลิเมนต์คือ OrderSize และ CustomerType และผลลัพธ์ของเว็บเซอร์วิส เป็นประเภท discount_book_rules_ResultType ซึ่งประกอบด้วยข้อมูลอิลิเมนต์คือ Discount

```

<schema>
  <element name="discount_book_rules_ParamType">
    <complexType><sequence>
      <element name="OrderSize" type="integer" />
      <element name="CustomerType" type="string" />
    </sequence></complexType>
  </element>
  <element name="discount_book_rules_ResultType">
    <complexType><sequence>
      <element name="Discount" type="double" />
    </sequence></complexType>
  </element>
</schema>

```

รูปที่ 2.5 ตัวอย่างเอกซ์เอ็มแอลสคีมา

- 4) วิธีการส่งข้อมูลระหว่างกัน (Binding information) เช่น ส่งผ่านโปรโตคอล HTTP เป็นต้น
- 5) ที่อยู่ของเว็บเซอวิส โดยระบุเป็นที่อยู่ไอพี หรือชื่อเครื่องในกรณีที่อยู่ในวงแลนเดียวกัน
- 6) ข้อมูลพาร์ทเนอร์ (Partner) แสดงลักษณะเฉพาะถึงความสัมพันธ์เกี่ยวกับการสนทนา (Conversational relationship) ระหว่างกระบวนการบีเพลกับเว็บเซอวิสที่บีเพลเรียกใช้และบทบาทหน้าที่ (Role) ของเว็บเซอวิสที่มีต่อกระบวนการบีเพล

2.1.6 เอกซ์พาท (XPath) [13]

เอกซ์พาทเป็นภาษานิพจน์ (Expression language) ที่ใช้ในการประมวลผลและเข้าถึงข้อมูลในเอกสารเอกซ์เอ็มแอล ช่วยในการอ้างถึงโหนด แอตทริบิวต์และค่าภายในเอกสารเอกซ์เอ็มแอล ตัวอย่างเช่นเอกสารเอกซ์เอ็มแอลดังรูปที่ 2.6 ถ้าเขียนนิพจน์ “bookstore/book” หมายถึงเลือกอิลิเมนต์ book ทั้งหมดที่เป็นแท็กลูกของแท็ก bookstore ถ้าเขียนนิพจน์ “bookstore/book[1]” หมายถึงเลือกอิลิเมนต์ book อันแรกของแท็ก bookstore


```

<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>

```

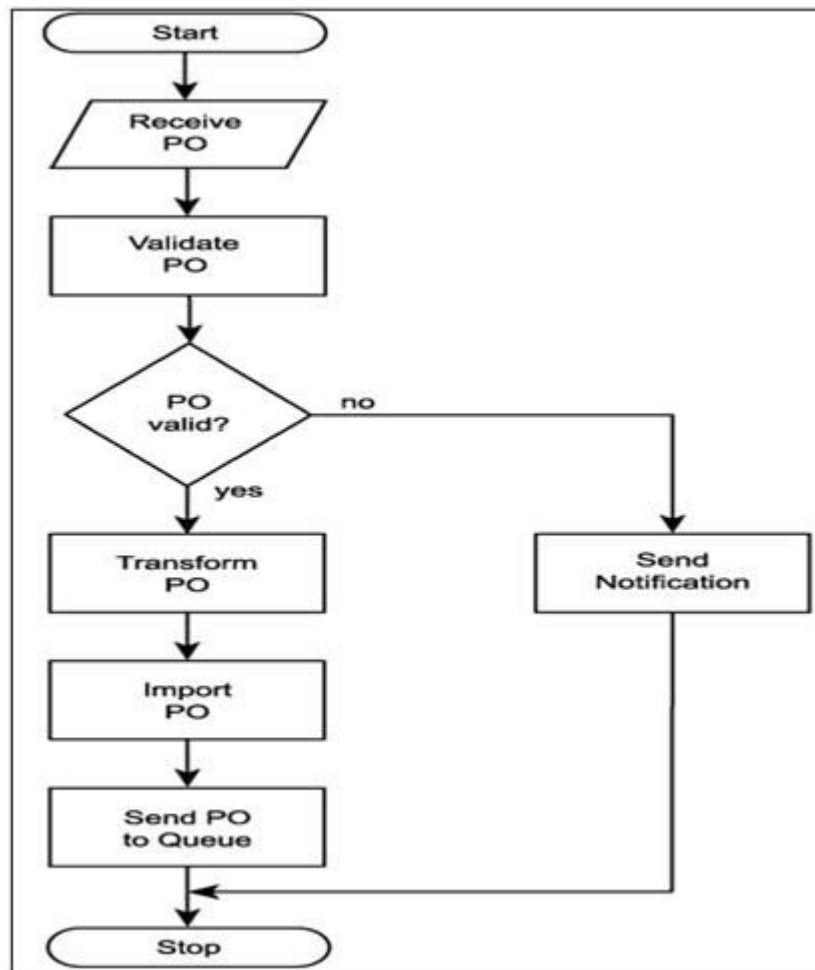
รูปที่ 2.6 ตัวอย่างเอกสารเอกซ์เอ็มแอล

2.1.7 กระบวนการธุรกิจ (Business Process) [14]

กระบวนการธุรกิจ คือ กลุ่มของกิจกรรม (Activity) ที่ทำงานร่วมกัน เพื่อเป้าหมายบางอย่าง ผลลัพธ์ของกระบวนการธุรกิจอาจเป็นข้อมูล สินค้าหรือบริการเพื่อสนองตอบต่อผู้ใช้หรือลูกค้า เช่น กระบวนการธุรกิจตรวจสอบใบสั่งซื้อสินค้า ดังรูปที่ 2.5 เป็นต้น

กระบวนการธุรกิจอาจมีการกระทำที่แปรตามเงื่อนไข หรือกฎธุรกิจได้ ดังรูปที่ 7 ถ้าใบสั่งซื้อสินค้าถูกต้องก็จะแปลงข้อมูลใบสั่งซื้อสินค้า แต่ถ้าไม่ถูกต้องก็จะส่งข้อความเตือน นอกจากนี้กระบวนการธุรกิจยังสามารถประกอบด้วยบริการ (Service) ต่างๆ ซึ่งบริการแต่ละอันจะรับผิดชอบกิจกรรมในกระบวนการธุรกิจ [15] ในขอบเขตของงานวิจัยนี้ กิจกรรมต่างๆ ในกระบวนการธุรกิจจะเป็นเว็บเซอร์วิส

กระบวนการธุรกิจถูกนำเสนอในรูปแบบของสายงานกิจกรรม (Activity flow) ซึ่งระบุลักษณะการทำงานร่วมกันของกิจกรรม (Orchestration) ที่ต้องทำเพื่อให้สำเร็จตามเป้าหมาย กระบวนการธุรกิจซึ่งถูกนำเสนอด้วยสายงานนี้มีวิวัฒนาการ 2 วิธี คือ ใช้ภาษาโปรแกรมทั่วไป เช่น จาวา เป็นต้น หรือ ใช้ภาษาสำหรับพัฒนากระบวนการธุรกิจโดยเฉพาะ คือ ภาษาบีพีดีแอล (Business Process Definition Language: BPD) เช่น ภาษาบีเพล และภาษาเอกซ์พีดีแอล (XML Process Definition Language: XPDL) เป็นต้น ภาษาประเภทหลังนี้ช่วยเพิ่มความสะดวกในการกำหนดกระบวนการธุรกิจที่ต้องการติดต่อกับเว็บเซอร์วิส



รูปที่ 2.7 กระบวนการธุรกิจตรวจสอบใบสั่งซื้อสินค้า [15]

2.1.8 ภาษาบีเพล (Business Process Execution Language: BPEL) [4]

ภาษาบีเพล เป็นภาษาสำหรับนิยามกระบวนการธุรกิจ ภาษาบีเพลทำหน้าที่ควบคุมการทำงานของเว็บเซอร์วิสต่างๆ ที่ติดต่อกับกระบวนการธุรกิจ พัฒนามาจากภาษาเอกซ์เอ็มแอล ลักษณะของภาษาบีเพล คือ เป็นภาษาเชิงกระบวนการคำสั่ง (Procedural Language) คล้ายกับแผนภูมิสายงาน (Flow Chart) ทำหน้าที่กำหนดว่าจะเรียกเว็บเซอร์วิสไหน เมื่อไหร่ และอาจมีการเก็บตัวแปรด้วย การทำงานจะไปข้างหน้าเรื่อยๆ จนจบ ดังรูปที่ 2.8

```

<process name="FullTravelPackage" ..>
  <sequence>
    <receive partner="client" operation="getTravelPackage" variable="request" .../>
    <sequence>
      <invoke partner="airline" operation="getFlight" outputVariable="flightout" />
      <invoke partner="hotel" operation="getRoom" .../>
    </sequence>
    <reply partner="client" operation="getTravelPackage" variable="proposition" .../>
  </sequence>
</process>

```

รูปที่ 2.8 ตัวอย่างภาษาบีเพล

จากตัวอย่างด้านบน จะมีการรับค่าพารามิเตอร์จากผู้เรียกใช้เก็บในตัวแปรชื่อ request จากนั้นก็เรียกใช้เว็บเซอร์วิส getFlight และ getRoom ตามลำดับ สุดท้ายคืนค่าตัวแปรชื่อ proposition กลับไปยังผู้เรียกใช้ จะเห็นได้ว่าจากมุมมองของผู้เรียกใช้บีเพลจะเป็นเว็บเซอร์วิสตัวหนึ่ง แต่เป็นเว็บเซอร์วิสที่เรียกเว็บเซอร์วิสอื่นในการทำงาน ภาษาบีเพลกำลังเป็นภาษามาตรฐานสำหรับนิยามกระบวนการธุรกิจเพราะได้รับการสนับสนุนจากองค์กรใหญ่ [14]

2.1.9 ภาษาโดเมนเฉพาะ (Domain-specific language) [16]

ภาษาโดเมนเฉพาะ คือ ภาษาโปรแกรม หรือข้อกำหนดภาษาที่กระทำกรทำได้ (Executable specification language) ที่พัฒนามาสำหรับงานหรือโดเมน (Domain) เฉพาะอย่าง เช่น ภาษาเอสคิวแอล เป็นต้น ซึ่งใช้สำหรับดึงข้อมูลจากฐานข้อมูล หรือภาษาแยค (YACC) ซึ่งใช้สร้างตัวแจง (Parser) เป็นต้น ซึ่งตรงข้ามกับภาษาโปรแกรมทั่วไป (General-purpose language) เช่น ซีหรือจาวา เป็นต้น ภาษาโดเมนเฉพาะอาจเป็นภาษาข้อความหรือภาษารูปภาพได้ [17]

นอกจากนี้ภาษาโดเมนเฉพาะยังถือว่าเป็นภาษาโปรแกรมสำหรับผู้ใช้นั้นปลาย (End-user programming) ซึ่งช่วยให้ผู้ใช้ทั่วไปสามารถเขียนโปรแกรมง่ายขึ้น เพราะภาษาโดเมนเฉพาะซ่อนความซับซ้อนของปัญหา และแสดงคำศัพท์เฉพาะ (Terminology) ที่ผู้ใช้คุ้นเคย [16] ตัวอย่างเช่นการใช้แมโครหรือภาษาบทคำสั่ง (Scripting language) เช่น การโปรแกรมแผ่นตารางทำการ (Spread sheet programming) โดยใช้ภาษาแมโครในเอกซ์เซล เป็นต้น

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัย “Hybrid Web Service Composition: Business Processes Meet Business Rules” โดย Anis Charfi et al., 2004 [6]

งานวิจัยนี้นำเสนอการแยกกฎธุรกิจ ออกจากตัวกระบวนการธุรกิจ โดยใช้ภาษาพีเพิล กำหนดกระบวนการธุรกิจ แล้วใช้หลักการของ aspect-oriented programming แยกกฎธุรกิจ ออกมาเป็นแอสเพ็ค (Aspect) งานวิจัยนี้เน้นที่การจัดการ และแยกกฎธุรกิจออกเป็นส่วนๆ (Module) หรือแอสเพ็ค และสร้างเครื่องมือเพื่อรวมแอสเพ็คเหล่านี้เข้ากับกระบวนการธุรกิจเมื่อถึง เวลาทำงานจริง งานวิจัยนี้เป็นการต่อยอดภาษาพีเพิล แต่ว่าในส่วนของการกฎธุรกิจผู้พัฒนาจะต้อง กำหนดด้วยภาษาพีเพิลโดยตรง ผู้กำหนดกฎจึงต้องมีความรู้เรื่องภาษาพีเพิลพอสมควร นอกจากนี้กฎที่ได้ต้องรวมกับพีเพิลที่มีอยู่แล้ว ต่างจากงานวิจัยที่นำเสนอที่กฎแต่ละกฎสามารถ ทำงานในรูปแบบเว็บเซอร์วิสที่ทำงานอิสระได้ด้วยตัวเอง

2.2.2 งานวิจัย “Orchestrating Supply Chain Interactions Using Emerging Process Description Language and Business Rules” โดย Costello et al., 2004 [19]

งานวิจัยนี้นำเสนอซอฟต์แวร์ตัวกลางที่เป็นเครื่องประมวลกฎ (Rule Engine) ทำหน้าที่ ตีความกฎ ซึ่งเขียนในรูปแบบภาษาอาร์เอ็มแอล (RML) [18] โดยซอฟต์แวร์ตัวกลางนี้จะรองรับข้อความ เอกซ์เอ็มแอลจากผู้เรียกใช้ จากนั้นจะค้นหากฎที่เหมาะสม ตรวจสอบ ประเมินเงื่อนไข และทำงาน ตามที่กำหนดไว้ เช่น การเรียกเว็บเซอร์วิส เป็นต้น นอกจากนี้งานวิจัยยังนำเสนอเครื่องมือสำหรับ แก้ไขกฎ (Rule Editor) แต่เครื่องมือที่ว่าแก้ไขกฎในรูปแบบ IF-THEN ซึ่งเมื่อกฎมีความซับซ้อนจะแก้ไข กฎได้ลำบาก นอกจากนี้ผู้ใช้ต้องมีความรู้เกี่ยวกับเอ็มเอ็มแอลสคีมาด้วย (XML Schema) กฎที่ได้ จากงานวิจัยจะอยู่ในรูปแบบภาษาอาร์เอ็มแอล และต้องทำงานร่วมกับเครื่องประมวลกฎ ไม่สามารถ ทำงานในรูปแบบเว็บเซอร์วิสได้

2.2.3 งานวิจัย “Web Services Based Execution of Business Rules” โดย R. Schmidt, 2002 [20]

งานวิจัยนี้นำเสนอการประมวลผลกระบวนการธุรกิจ โดยใช้กฎธุรกิจ ร่วมกับเว็บเซอร์วิส งานวิจัยนี้ใช้วิธีการฝังกฎที่เขียนในรูปแบบภาษาอาร์เอ็มแอลไว้ในส่วนเฮดเดอร์ (Header) ของ ข้อความโซพ (SOAP) โดยกฎจะถูกตีความที่เครื่องแม่ข่ายกลางหลายๆ เครื่อง ทำให้ไม่ต้องมี เครื่องแม่ข่ายศูนย์กลางที่รับภาระประมวลกฎเพียงเครื่องเดียว อย่างไรก็ตามกฎไม่สามารถทำงาน โดยตัวเองอิสระได้ ต้องผูกติดกับเว็บเซอร์วิสอื่นๆ

2.2.4 งานวิจัย “Integration of the Decision Table Formalism with a Relational Database Environment” โดย J. Vanthienen et al., 1995 [9]

งานวิจัยนี้นำเสนอวิธีการเก็บตารางการตัดสินใจ และกฎลงในฐานข้อมูล โดยเสนอสคีมา (Schema) ของตาราง และความสัมพันธ์ต่างๆ ในฐานข้อมูล นอกจากนี้ยังนำเสนอการสร้าง และการตรวจสอบ (Verification) ตารางการตัดสินใจในฐานข้อมูลด้วยคำสั่งเอสควิแอล (SQL) สุดท้ายงานวิจัยนี้เสนอเครื่องมือโปรโลกา (Prolog) ซึ่งเป็นเครื่องมือสำหรับสร้าง แก้ไข และจัดการ ตารางการตัดสินใจด้วย งานวิจัยที่เรานำเสนอนี้จะนำแนวคิดจากงานวิจัยนี้ไปใช้ในการเก็บกฎธุรกิจที่ได้จากภาษาโดเมนเฉพาะ ก่อนที่จะนำไปสร้างบีเพลต่อไป

2.2.5 งานวิจัย “A Domain Specific Language for Securing Distributed Systems” โดย H. Hamdi et al., 2007 [21]

งานวิจัยนี้เสนอภาษาโดเมนเฉพาะเพื่อกำหนดนโยบายความมั่นคง (Security policy) ในระบบแบบกระจาย (Distributed system) เช่น การให้อำนาจ (Authorization) การควบคุมการเข้าถึง (Access control) คุณภาพบริการ (Quality of Service) เป็นต้น นโยบายที่กำหนดในภาษานี้เขียนในรูปของกฎงานวิจัยนี้ยังพัฒนาคอมไพเลอร์ที่สามารถตรวจสอบความถูกต้อง และความขัดแย้ง (Conflict) ของนโยบายได้ด้วย งานวิจัยนี้อ้างว่าภาษานี้สามารถแสดงนโยบายที่ซับซ้อนให้เข้าใจง่ายขึ้นแก่ผู้ใช้ อย่างไรก็ตามภาษาโดเมนเฉพาะของงานวิจัยนี้ออกแบบมาเพื่องานด้านความมั่นคง (Security) โดยเฉพาะ โครงสร้างและคำที่ใช้ในภาษาจึงเหมาะกับงานนั้นเท่านั้น

2.2.6 งานวิจัย “MTrans, a DSL for Model Transformation” โดย M. Peltier, 2002 [22]

งานวิจัยนี้เสนอภาษาโดเมนเฉพาะสำหรับแบบแปลงตัวแบบ (Model) ของยูเอ็มแอล จากรูปแบบหนึ่งไปเป็นรูปแบบหนึ่ง เช่น แปลงคลาสไดอะแกรมจากแบบ (Pattern) หนึ่งไปเป็นอีกแบบหนึ่ง เป็นต้น งานวิจัยนี้ใช้ภาษาโอซีแอล (Object Constraint Language) [23] มาใช้ในภาษาโดเมนเฉพาะ งานวิจัยนี้อ้างว่าภาษาโดเมนเฉพาะที่ได้ใช้เวลาเรียนรู้สั้น และแสดงในรูปแบบที่เข้าใจง่าย อย่างไรก็ตามผู้ใช้ต้องเป็นผู้เขียนภาษาโดเมนเฉพาะในรูปข้อความเอง ดังนั้นผู้ใช้ต้องมีพื้นฐานของภาษาโอซีแอลบ้าง เพื่อเข้าใจภาษานี้ได้ง่ายขึ้น

บทที่ 3

การออกแบบระบบ

งานวิจัยนำเสนอระบบสำหรับพัฒนากฎธุรกิจ ที่ใช้ในกระบวนการธุรกิจด้วยตารางการตัดสินใจและภาษานิยามกฎธุรกิจหรือภาษาปียาร์ดีแอล (BRDL: Business Rules Definition Language) ซึ่งเป็นภาษาโดเมนเฉพาะที่งานวิจัยนี้ออกแบบขึ้น งานวิจัยนี้จะขออ้างถึงภาษานี้ อย่างย่อๆ ว่าภาษาปียาร์ดีแอล ด้วยแนวทางที่งานนี้วิเคราะห์ซึ่งมีความรู้ด้านกฎธุรกิจ แต่อาจไม่มีทักษะด้านการเขียนโปรแกรม สามารถสร้างธุรกิจได้ด้วยตัวเอง จากนั้นกฎธุรกิจจะได้รับการแปลงให้เป็นบทคำสั่งบีเพล (BPEL script) โดยบทคำสั่งบีเพลที่ได้จะทำงานในรูปของเว็บเซอร์วิส ซึ่งสามารถทำงานร่วมกันกระบวนการธุรกิจและเว็บเซอร์วิสอื่นๆได้

3.1 ภาษานิยามกฎธุรกิจ

งานวิจัยนี้ได้พัฒนาภาษานิยามกฎธุรกิจ หรือภาษาปียาร์ดีแอลขึ้นเพื่อกำหนดกฎธุรกิจ และตารางการตัดสินใจ ภาษาปียาร์ดีแอลเป็นภาษาโดเมนเฉพาะสำหรับกำหนดกฎธุรกิจ อยู่บนพื้นฐานของภาษาเอกซ์เอ็มแอลซึ่งประกอบด้วยแท็กต่างๆ ภาษาปียาร์ดีแอลแบ่งแท็กเป็น 3 กลุ่ม คือ กลุ่มแท็กกำหนดข้อเท็จจริง กลุ่มแท็กกำหนดเซตกฎธุรกิจและกลุ่มแท็กกำหนดกฎธุรกิจ โดยมีรายละเอียดดังนี้

3.1.1 กลุ่มแท็กกำหนดข้อเท็จจริง

กลุ่มแท็กกำหนดข้อเท็จจริง คือ กลุ่มแท็กที่ทำหน้าที่กำหนดข้อเท็จจริงได้แก่ เงื่อนไขและการกระทำที่สามารถใช้ในกฎธุรกิจได้ ข้อเท็จจริงเป็นเหมือนแผ่นแบบ (Template) ของเงื่อนไขและการกระทำ โดยก่อนที่จะสร้างกฎธุรกิจต้องมีการกำหนดเงื่อนไข และการกระทำไว้ก่อน แผ่นแบบเป็นเหมือนวัตถุดิบสำหรับสร้างกฎธุรกิจ ซึ่งโดยปกติทำโดยโปรแกรมเมอร์หรือผู้มีความรู้ด้านเว็บเซอร์วิสและการเขียนโปรแกรม เอกซ์เอ็มแอลของกลุ่มแท็กกำหนดข้อเท็จจริงแสดงในภาคผนวก ค กลุ่มแท็กกำหนดข้อเท็จจริงแบ่งเป็น 4 ประเภท ได้แก่ แท็กกำหนดแผ่นแบบเงื่อนไข (Condition template) แท็กกำหนดแผ่นแบบการกระทำทั่วไป (Regular action template) แท็กกำหนดกิจกรรมในการกระทำ และแท็กกำหนดการกระทำที่เรียกตารางย่อย มีรายละเอียดดังนี้

3.1.1.1 แท็กกำหนดแผ่นแบบเงื่อนไข

แผ่นแบบเงื่อนไข เป็นเงื่อนไขที่สามารถใช้ในกฎต่างๆ ได้ แต่ละแผ่นแบบเงื่อนไขจะประกอบด้วยตัวแปรปรับแต่ง (Custom variable) ดังตัวอย่างรูปที่ 3.1 แผ่นแบบเงื่อนไข provinceCondition มีตัวแปรชื่อ province เมื่อแผ่นแบบเงื่อนไขนี้ถูกเพิ่มเข้าสู่กฎธุรกิจ ผู้ใช้ต้องกำหนดตัวดำเนินการเปรียบเทียบ (Compare operator) และค่าให้แก่ตัวแปร เช่น province = "Phuket" เป็นต้น

ในแต่ละแผ่นแบบเงื่อนไขซึ่งประกอบด้วยแท็กดังนี้

- 1) แท็ก conditionTemplate เป็นแท็กที่เก็บข้อมูลเกี่ยวกับแผ่นแบบเงื่อนไข ได้แก่ รหัสชื่อและคำอธิบาย ซึ่งระบุในแอตทริบิวต์ id name และ description ตามลำดับ
- 2) แท็ก customVariables เก็บข้อมูลของตัวแปรปรับแต่งทั้งหมด ซึ่งเป็นตัวแปรที่ผู้ใช้ต้องกำหนดตัวดำเนินการเปรียบเทียบและค่าให้แก่เงื่อนไข เมื่อเลือกแผ่นแบบเงื่อนไขนี้ไปใช้กับกฎธุรกิจ ในหนึ่งแผ่นแบบเงื่อนไขสามารถมีได้ตัวแปรได้หลายตัว โดยตัวแปรทั้งหมดระบุภายในแท็กนี้และตัวแปรแต่ละตัวระบุด้วยแท็ก customVariable
- 3) แท็ก customVariable ระบุข้อมูลเกี่ยวกับตัวแปรปรับแต่งที่จะกำหนดตัวดำเนินการเปรียบเทียบและค่าโดยผู้ใช้ประกอบด้วยข้อมูล 3 อย่าง ได้แก่ รหัส ชื่อและประเภทข้อมูล ซึ่งระบุในแอตทริบิวต์ id name และ type ตามลำดับ ประเภทข้อมูลของตัวแปรสามารถมีได้ 4 ประเภทคือ สายอักขระ (String) จำนวนเต็ม (Integer) จำนวนทศนิยม (Double) และบูลีน (Boolean) ในกรณีที่ค่าของตัวแปรเป็นค่าที่แน่นอนตายตัว สามารถใช้แท็ก enum เพื่อกำหนดค่าเหล่านั้นได้
- 4) แท็ก enum ระบุค่าที่เป็นไปได้ของตัวแปรปรับแต่ง ถ้ามีแท็ก enum ในแท็ก customVariable ผู้ใช้ต้องใช้ค่าจากแท็ก enum เท่านั้นเพื่อกำหนดค่าให้แก่ตัวแปรปรับแต่ง แท็ก

```
<conditionTemplate id="2" name="provinceCondition" description="Province in
thailand that the employee works in">
  <naturalLanguage>The employee works in province ?</naturalLanguage>
  <customVariables>
    <customVariable id="1" name="province" type="string" >
      <enum value="10" text="Krung Thep Maha Nakhon" />
      <enum value="83" text="Phuket" />
    </customVariable>
  </customVariables>
</ conditionTemplate>
```

รูปที่ 3.1 ตัวอย่างแผ่นแบบเงื่อนไข

enum มีแอดทริบิวต์ value เพื่อระบุค่าและแอดทริบิวต์ text เพื่อระบุการแสดงผลของค่านั้น เช่น ดังตัวอย่างในหน้าที่แล้วตัวแปรปรับเป็นจังหวัดในประเทศไทยซึ่งมีค่าแน่นอน จึงมีแท็ก enum ซึ่งเป็นชื่อจังหวัดเพื่อให้ผู้ใช้เลือก แอดทริบิวต์ text จึงเป็นชื่อจังหวัด แต่แอดทริบิวต์ value เป็นรหัสจังหวัด ซึ่งเป็นค่าจริงที่ใช้ประเมินเงื่อนไขในกฎธุรกิจ

5) แท็ก naturalLanguage ระบุประโยคในรูปภาษาธรรมชาติ เป็นประโยคที่ใช้แสดงเงื่อนไขในภาษาโดเมนเฉพาะและตารางการตัดสินใจ ประโยคโดยทั่วไปจะประกอบด้วยข้อเท็จจริง ตัวดำเนินการ และเครื่องหมายคำถาม เช่น “The height is [operator] ?” เป็นต้น ตัวดำเนินการ (ระบุด้วย “[operator]”) และ เครื่องหมายคำถามเกี่ยวข้องกับตัวแปรปรับแต่งที่กล่าวในข้อ 3 เมื่อกฎถูกแสดงแก่ผู้ใช้ “[operator]” และเครื่องหมายคำถามจะถูกแทนที่ด้วยตัวดำเนินการเปรียบเทียบและค่าของตัวแปรซึ่งกำหนดโดยผู้ใช้ตามลำดับ

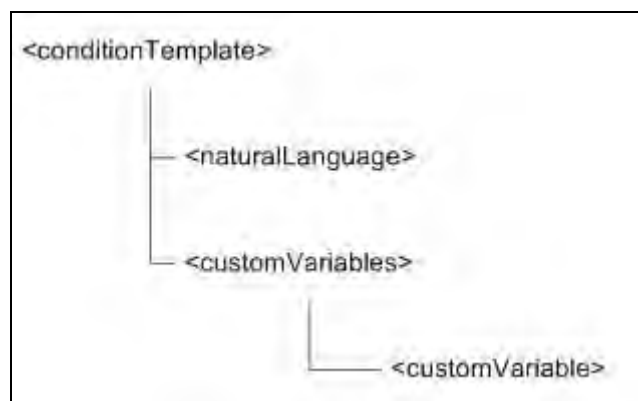
ในกรณีที่ในหนึ่งเงื่อนไขมีตัวแปรหลายตัว ลำดับของตัวดำเนินการและเครื่องหมายคำถามจะสัมพันธ์กับลำดับตัวแปรในแท็ก customVariables นั่นคือ เครื่องหมายคำถามแรกจะแทนตัวแปรแรกในแท็ก customVariables เครื่องหมายคำถามที่สองจะแทนตัวแปรที่สอง ดังรูปที่

3.2

```
<conditionTemplate id="4" name="Body Condition" description="Checking the height and weight of a customer.">
  <naturalLanguage>
    The height is [operator] ? c.m. and weight is [operator] ? k.g.
  </naturalLanguage>
  <customVariables>
    <customVariable id="1" name="Height" type="double"/>
    <customVariable id="2" name="Weight" type="double"/>
  </customVariables>
</conditionTemplate>
```

รูปที่ 3.2 ความสัมพันธ์ของตัวแปรที่กำหนดค่าโดยผู้ใช้กับประโยคในรูปภาษาธรรมชาติ

โครงสร้างภายในแท็ก conditionTemplate แสดงดังรูป 3.3



รูปที่ 3.3 โครงสร้างภายในแท็ก conditionTemplate

3.1.1.2 แท็กกำหนดแผ่นแบบการกระทำทั่วไป (Regular action template)

แผ่นแบบการกระทำทั่วไป คือ แผ่นแบบที่ระบุการกระทำที่ไม่ใช่การเรียกตารางย่อย รูปแบบการกระทำหรือกิจกรรมมีการระบุชัดเจนในไฟล์ข้อเท็จจริง เมื่อกล่าวถึงแผ่นแบบการกระทำในหัวข้อนี้ จะหมายถึงแผ่นแบบการกระทำทั่วไปดังตัวอย่างในรูปที่ 3.4

```
<actionTemplate id="2" name="RecordLoanForReview" description="Record loan to
database to wait manager">
  <naturalLanguage>Wait for manager approval ?</naturalLanguage>
  <customVariables>
    <customVariable id="1" name="catId" type="string"/>
  </customVariables>
  <activities>
    <variable id="1" name="item_cost" type="int"/>
    <invoke service="InventoryService" operation="getCostItem"
      wsdl="InventoryService.wsdl" partnerLinkType="InventoryLinkType">
      <parameter customVariableId="1" toElement="parameters/catID"/>
      <result element="parameters/return" variableId = "1"/>
    </invoke>
    <return>
      <parameter variableId="1"/>
    </return>
  </activities>
</actionTemplate>
```

รูปที่ 3.4 ตัวอย่างแผ่นแบบการกระทำ

แผ่นแบบการกระทำ เป็นการกระทำที่สามารถใช้ในกฎต่างๆ ได้ แต่ละแผ่นแบบการกระทำกำหนดด้วยแท็ก actionTemplate

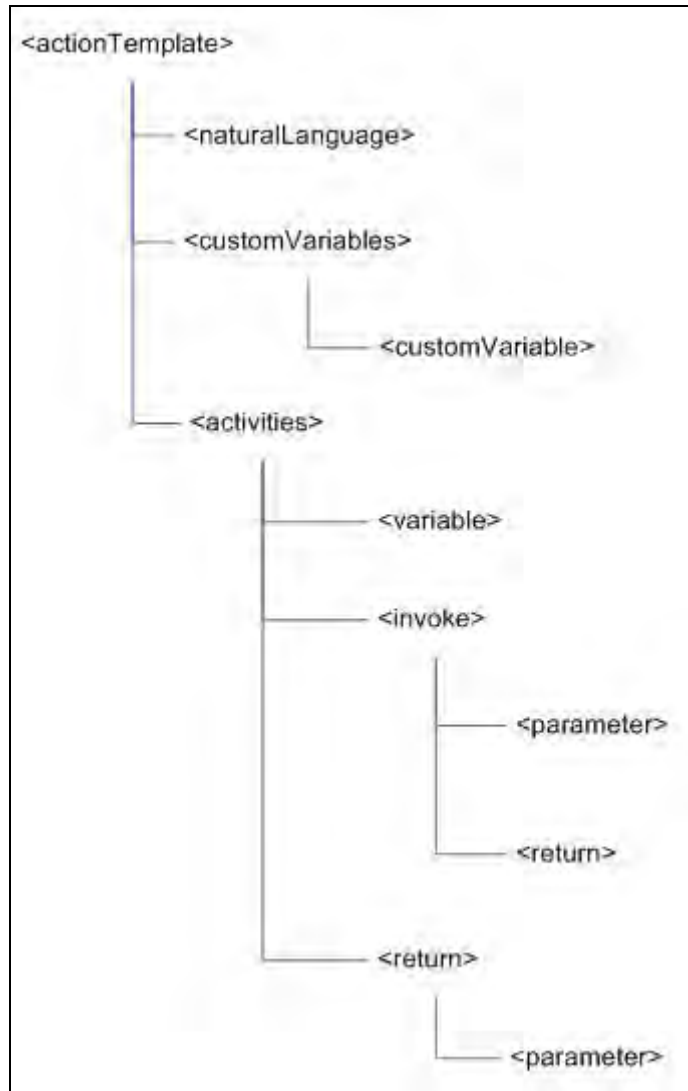
แผ่นแบบการกระทำประกอบด้วยแท็กดังนี้

1) แท็ก actionTemplate ระบุข้อมูลเกี่ยวกับแผ่นแบบการกระทำ ได้แก่ รหัส ชื่อและคำอธิบาย โดยระบุในแอตทริบิวต์ id name และ description ตามลำดับ

2) แท็ก customVariables เก็บข้อมูลของตัวแปรปรับแต่งทั้งหมด ซึ่งเป็นตัวแปรที่ผู้ใช้ต้องกำหนดค่าให้แก่การกระทำเมื่อเลือกแผ่นแบบการกระทำนี้ไปใช้กับกฎธุรกิจ ในหนึ่งแผ่นแบบ

การกระทำสามารถมีได้ตัวแปรได้หลายตัว โดยตัวแปรทั้งหมดระบุภายในแท็กนี้และตัวแปรแต่ละตัวระบุด้วยแท็ก customVariable

โครงสร้างภายในแท็ก actionTemplate ของแผ่นแบบการกระทำทั่วไปแสดงดังรูป 3.5



รูปที่ 3.5 โครงสร้างภายในแท็ก actionTemplate

แผ่นแบบเงื่อนไขและแผ่นแบบการกระทำทั้งหมดกำหนดภายในแท็ก conditionTemplates และ actionTemplates ตามลำดับดังตัวอย่างในรูป 3.6

```

<facts>
  <conditionTemplates>
    <conditionTemplate>...</conditionTemplate>
    <conditionTemplate>...</conditionTemplate>
  </conditionTemplates>
  <actionTemplates>
    <actionTemplate>...</actionTemplate>
    <actionTemplate>...</actionTemplate>
  </actionTemplates>
</facts>

```

รูปที่ 3.6 ตัวอย่างแผ่นแบบเงื่อนไขและแผ่นแบบการกระทำ

3.1.1.3 แท็กกำหนดกิจกรรมในการกระทำ

จากที่กล่าวหัวข้อที่แล้วว่าในการกระทำประกอบด้วยกิจกรรมต่างๆ ซึ่งทำงานเป็นลำดับจากบนลงล่างจนครบ กิจกรรมทั้งหมดอยู่ภายในแท็ก activities กิจกรรมในงานวิจัยนี้มี 2 ประเภท ได้แก่ การคืนค่ากลับไปยังตารางการตัดสินใจซึ่งกำหนดด้วยแท็ก return และการเรียกใช้เว็บเซอร์วิสซึ่งกำหนดด้วยแท็ก invoke นอกจากนี้ภายในแท็ก activities สามารถมีแท็ก variable ซึ่งทำหน้าที่เป็นตัวแปรเฉพาะที่ (Local variable) ภายในแท็ก activities กิจกรรมในการกระทำประกอบด้วยแท็กดังนี้

1) แท็ก activities เป็นแท็กที่บรรจุกิจกรรมทั้งหมดของการกระทำไว้ แท็กที่จะกล่าวต่อไปล้วนเป็นแท็กที่อยู่ภายในแท็กนี้

2) แท็ก variable ทำหน้าที่เป็นตัวแปรเฉพาะที่ภายในแท็ก activities แท็ก invoke สามารถอ้างถึงเพื่อนำค่าของตัวแปรเป็นพารามิเตอร์ของเว็บเซอร์วิส หรือเป็นตัวแปรเก็บผลลัพธ์จากเว็บเซอร์วิส แท็ก variable ประกอบด้วยแอตทริบิวต์ id name type ระบุไอดี ชื่อและประเภทตัวแปร (String Integer Double หรือ Boolean) ตามลำดับ ถ้าต้องการระบุค่าเริ่มต้นให้กับตัวแปรทำโดยระบุค่าในแอตทริบิวต์ value ดังตัวอย่างในรูปที่ 3.7

```

<variable id="1" name="item_id" type="string" value="ABC123"/>
<variable id="2" name="item_cost" type="double"/>

```

รูปที่ 3.7 ตัวอย่างแท็ก variable

3) แท็ก invoke ทำหน้าที่เรียกเว็บเซอร์วิส ประกอบด้วยแอตทริบิวต์ service wsdl operation และ partnerLinkType ซึ่งเป็นข้อมูลที่จำเป็นต่อการเรียกใช้เว็บเซอร์วิส แท็ก invoke มีแท็ก parameter และแท็ก result ทำหน้าที่ระบุพารามิเตอร์และผลลัพธ์ของเว็บเซอร์วิส ตามลำดับ ดังตัวอย่างในรูปที่ 3.8

```
<invoke service="InventoryService" operation="getCostItem"
  wsdl="InventoryService_on_Sun.wsdl"
  partnerLinkType="InventoryLinkType">
  <parameter customVariableId="1" toElement="parameters/catID"/>
  <result element="parameters/return" variableId = "2"/>
</invoke>
```

รูปที่ 3.8 ตัวอย่างแท็ก invoke

4) แท็ก parameter กำหนดพารามิเตอร์ของเว็บเซอร์วิสที่เรียกโดยแท็ก invoke พารามิเตอร์ต้องระบุตัวแปรที่จะส่งเป็นพารามิเตอร์ โดยอาจเป็นตัวแปรปรับแต่งของการกระทำ (ระบุด้วยแอตทริบิวต์ customVariableId) หรือเป็นตัวแปรเฉพาะที่ของแท็ก activities (ระบุด้วยแอตทริบิวต์ variableId) นอกจากนี้พารามิเตอร์ต้องระบุแอตทริบิวต์ toElement เพื่อกำหนดว่าค่าของตัวแปรจะกำหนดให้กับ element ไດของพารามิเตอร์ของเว็บเซอร์วิส การระบุค่าในแอตทริบิวต์นี้ใช้นิพจน์เอกซ์พาท (XPath expression)

5) แท็ก result กำหนดว่าผลลัพธ์ที่ได้จากเว็บเซอร์วิสจะนำมาเก็บในตัวแปรเฉพาะที่ตัวใด ซึ่งผลลัพธ์ที่เก็บสามารถนำไปใช้ในกิจกรรมต่อไปได้ แท็กนี้มีแอตทริบิวต์ variableId ระบุตัวแปรเฉพาะที่ที่ต้องการเก็บผลลัพธ์ และแอตทริบิวต์ element เพื่อระบุว่าจะเลือก element ไດของผลลัพธ์โดยใช้นิพจน์เอกซ์พาท

6) แท็ก return ทำหน้าที่กำหนดว่าจะคืนค่าของตัวแปรใดกลับไปยังกฎธุรกิจที่เรียกการกระทำ ภายในแท็กประกอบด้วยแท็ก parameter ซึ่งระบุตัวแปรที่ต้องการคืน ถ้าต้องการคืนค่าตัวแปรเฉพาะที่ทำโดยระบุแอตทริบิวต์ variableId และถ้าต้องการคืนค่าตัวแปรปรับแต่งทำโดยระบุแอตทริบิวต์ customVariableId ดังตัวอย่างในรูปที่ 3.9

```
<return>
  <parameter variableId="3"/>
  <parameter customVariableId="1"/>
</return>
```

รูปที่ 3.9 ตัวอย่างแท็ก return

3.1.1.4 แท็กกำหนดแผนแบบการกระทำสำหรับเรียกตารางย่อย

ตามที่กล่าวไว้ในส่วนทฤษฎีที่เกี่ยวข้องว่าตารางการตัดสินใจสามารถเรียกใช้ตารางย่อย ซึ่งเป็นตารางการตัดสินใจอื่นได้ ในงานวิจัยนี้ส่วนการกระทำของกฎธุรกิจสามารถเรียกตารางย่อย โดยเลือกแผนแบบการกระทำสำหรับเรียกตารางย่อย ดังรูปที่ 3.10

```
<action id="3" name="Subruleset" subruleset="true" description="Call other ruleset">
  <naturalLanguage>Call sub table</naturalLanguage>
</action>
```

รูปที่ 3.10 แผนแบบการกระทำสำหรับเรียกตารางย่อย

แผนแบบการกระทำชนิดนี้กำหนดด้วยแท็ก action เช่นเดียวกับแผนแบบการกระทำทั่วไป แต่ต้องกำหนดแอตทริบิวต์ subruleset เป็น true และแท็กต้องมีเพียงแท็ก naturalLanguage เท่านั้น ไม่มีตัวแปรปรับแต่งแต่อย่างใด เมื่อแผนแบบนี้ถูกเลือกเข้าสู่กฎธุรกิจ ผู้ใช้ต้องระบุตารางการตัดสินใจที่ต้องการเรียก และพารามิเตอร์ของตารางการตัดสินใจนั้น

3.1.2 กลุ่มแท็กกำหนดเซตกฎธุรกิจ

กลุ่มแท็กกำหนดเซตกฎธุรกิจทำหน้าที่กำหนดเซตกฎธุรกิจ ประกอบด้วยแท็กที่ใช้กำหนดเซตกฎธุรกิจ กฎธุรกิจ เงื่อนไขและการกระทำในกฎนั้น ดังตัวอย่างดังตัวอย่างในรูปที่ 3.11 เอกซ์เอ็มเอสซีมาของกลุ่มแท็กกำหนดเซตกฎธุรกิจแสดงในภาคผนวก ค กลุ่มแท็กกำหนดเซตกฎธุรกิจ ประกอบด้วยแท็กดังนี้

3.1.2.1 แท็ก ruleset คือ เซตกฎธุรกิจซึ่งประกอบด้วยกฎธุรกิจที่เกี่ยวข้องกัน แต่ละกฎกำหนดด้วยแท็ก rule แท็ก ruleset มีแอตทริบิวต์ name ระบุชื่อเซตกฎธุรกิจหรือตารางการตัดสินใจ แอตทริบิวต์ loop กำหนดว่าให้เซตกฎธุรกิจทำงานแบบวนซ้ำหรือไม่ โดยมีแอตทริบิวต์ loopFix กำหนดว่าจำนวนรอบของการวนซ้ำว่าเป็นแบบจำนวนรอบตายตัว (แอตทริบิวต์ loopCount กำหนดจำนวนรอบ) หรือเป็นแบบกำหนดด้วยเงื่อนไขซึ่งระบุในแท็ก loopConditions

3.1.2.2 แท็ก loopConditions ทำหน้าที่เก็บเงื่อนไขทั้งหมดของการวนซ้ำ ซึ่งแต่ละเงื่อนไขกำหนดด้วยแท็กตัวดำเนินการ

3.1.2.3 แท็กตัวดำเนินการ ระบุค่าและตัวดำเนินการเปรียบเทียบ ที่ผู้ใช้กำหนดแก่ตัวแปรหนึ่งตัวของเงื่อนไขซึ่งมี 7 ประเภท ได้แก่ lessThan lessThanEqualTo moreThan moreThanEqualTo equals notEquals และ between ประเภทของแท็กตัวดำเนินการขึ้นอยู่กับตัวดำเนินการเปรียบเทียบที่ผู้ใช้กำหนด เช่น ถ้าผู้ใช้กำหนดตัวดำเนินการเปรียบเทียบเป็น “more

```

<ruleset name="discount_book_rules" loop="True" loopCount="7"
loopFix="False" >
  <loopConditions operation="and">
    <lessThanOrEqualTo id="5" order="6" paramId="4" leftId="4" rightId="7"
excludeLeft="False" excludeRight="False" />
  </loopConditions>

  <group id="1" name="A" description="for customer A" />
  <group ... />

  <rule id="1" name="rule_1" groupId="1" default="False" order="1" >
    <conditions>...</conditions>
    <actions>...</actions>
  </rule>
  <rule>...</rule>
  <returnVariables>
    <returnVariable id="4" name="discount" type="double" />
    <returnVariable ... />
  </returnVariables>
  <customDtParamters>
    <customDtParamter id="1" name="product_category" type="string" />
    <customDtParamter .../>
  </customDtParamters>
</ruleset>

```

รูปที่ 3.11 ตัวอย่างไฟล์เซตกฎธุรกิจ

than or equal to” แท็กตัวดำเนินการจะเป็นประเภท moreThanOrEqualTo เป็นต้น แท็กตัวดำเนินการประกอบด้วยแอตทริบิวต์ดังนี้

1) แอตทริบิวต์ paramId เป็นแอตทริบิวต์ที่อ้างถึงพารามิเตอร์ของตารางการตัดสินใจ ในกรณีที่พารามิเตอร์ถูกใช้เป็นเงื่อนไข

2) แอตทริบิวต์ `returnVariableId` เป็นแอตทริบิวต์ที่อ้างถึงตัวแปรผลลัพธ์ของตารางการตัดสินใจ ในกรณีที่ตัวแปรผลลัพธ์ถูกใช้เป็นเงื่อนไข ถ้ามีแอตทริบิวต์นี้แล้วจะไม่มีแอตทริบิวต์ `paramId` ในทางตรงข้ามก็เช่นเดียวกัน

3) แอตทริบิวต์ `rightId` คือ ค่าที่ระบุโดยผู้ใช้ เช่น 781 2.5 “bookstore” true เป็นต้น ประเภทของค่าขึ้นอยู่กับประเภทข้อมูลของตัวแปร

4) แอตทริบิวต์ `leftId` โดยทั่วไปแอตทริบิวต์นี้จะมีค่าเดียวกับแอตทริบิวต์ `paramId` หรือ `returnVariableId` แต่ถ้าตัวแปรเป็นข้อมูลประเภทตัวเลขและแท็กตัวดำเนินการเป็น `between` แอตทริบิวต์ `leftId` จะเป็นค่าด้านซ้ายของตัวดำเนินการเปรียบเทียบของ `between` เช่น ถ้าผู้ใช้กำหนดเงื่อนไขด้วย `param_1` เป็น `between 10 and 25` ค่า `leftId` จะเป็น 10 และค่า `rightId` จะเป็น 25

5) แอตทริบิวต์ `excludeLeft` และ `excludeRight` ใช้ในกรณีที่แท็กตัวดำเนินการเป็น `between` ระบุว่าค่าที่ขอบด้านซ้ายและด้านขวาของ `between` จะถูกรวมในเงื่อนไขด้วยหรือไม่ ตามลำดับ ถ้าแอตทริบิวต์ มีค่าเป็น True หมายถึง ไม่รวม แต่ถ้าเป็น False หมายถึง รวม เช่น ถ้ากำหนดเงื่อนไขด้วย `param_1` เป็น `between 10 and 25` และแอตทริบิวต์ `excludeLeft` เป็น True และแอตทริบิวต์ `excludeRight` เป็น False เมื่อแปลงแท็กตัวดำเนินการออกมาเป็นเงื่อนไขจะหมายถึง $10 < OrderSize \text{ And } OrderSize \leq 25$

3.1.2.4 แท็ก `group` แทนกลุ่มของกฎธุรกิจ มีขึ้นจัดกลุ่มกฎธุรกิจภายในเซตกฎธุรกิจ

3.1.2.5 แท็ก `returnVariables` เก็บพารามิเตอร์ที่ผู้ใช้สร้างขึ้นเองทั้งหมดของตารางการตัดสินใจ แต่ละพารามิเตอร์กำหนดด้วยแท็ก `returnVariable`

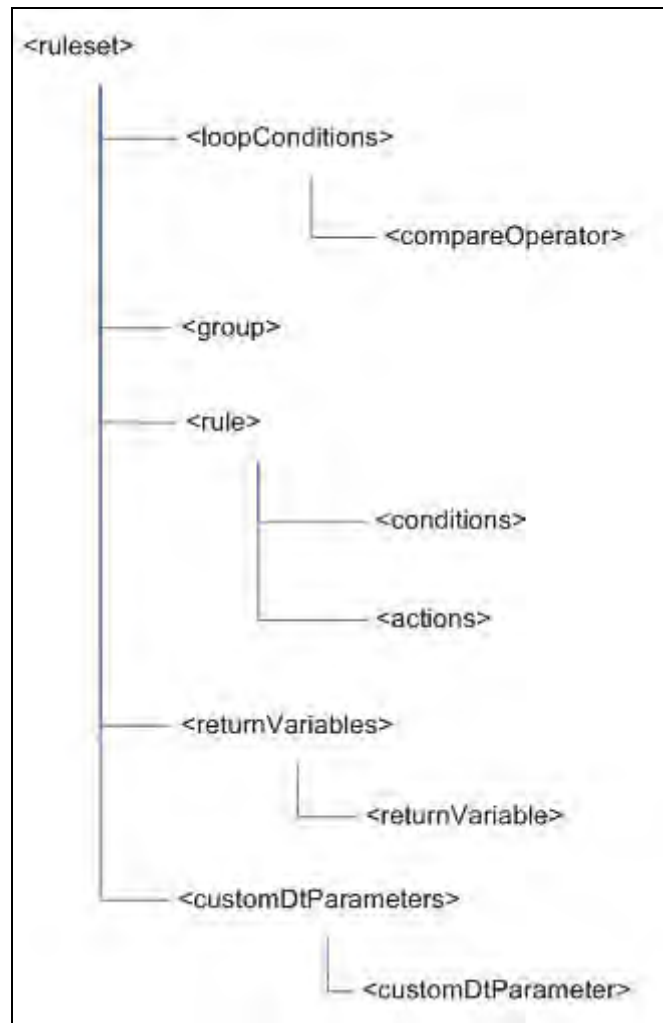
3.1.2.6 แท็ก `returnVariable` ทำหน้าที่เป็นตัวแปรผลลัพธ์ของตารางการตัดสินใจ ซึ่งอาจมีหลายแท็กได้ มีแอตทริบิวต์ `id name type` กำหนดรหัส ชื่อ และประเภทข้อมูล

3.1.2.7 แท็ก `customDtParamters` เก็บพารามิเตอร์ที่ผู้ใช้สร้างขึ้นเองทั้งหมดของตารางการตัดสินใจ แต่ละพารามิเตอร์กำหนดด้วยแท็ก `customDtParamter`

3.1.2.8 แท็ก `customDtParamter` ทำหน้าที่เป็นพารามิเตอร์ที่ผู้ใช้สร้างขึ้นเองของตารางการตัดสินใจ ซึ่งอาจมีหลายแท็กได้ มีแอตทริบิวต์ `id name type` กำหนดรหัส ชื่อ และประเภทข้อมูล

3.1.2.9 แท็ก `rule` คือ กฎธุรกิจแต่ละกฎ ภายในประกอบด้วยเงื่อนไขและการกระทำ ซึ่งทั้งหมดอยู่ภายในแท็ก `conditions` และ `actions` ตามลำดับ ทั้งเงื่อนไขและการกระทำเป็นเหมือนกรณีตัวอย่าง (Instance) ของแผ่นแบบเงื่อนไขและการกระทำที่ระบุในไฟล์ข้อเท็จจริง

โครงสร้างภายในแท็ก `ruleset` ของแผ่นแบบการกระทำทั่วไปแสดงดังรูป 3.12



รูปที่ 3.12 โครงสร้างภายในแท็ก ruleset ของแผ่นแบบการกระทำทั่วไป

3.1.2.10 แท็ก conditions เป็นแท็กที่เก็บเงื่อนไขทั้งหมดของกฎธุรกิจ 1 กฎ และเป็นแท็กลูกของแท็ก rule ภายในแท็ก conditions มีแท็ก condition หลายแท็ก ดังตัวอย่างในรูปที่ 3.13 โดยทั่วไปเงื่อนไขในกฎธุรกิจอาจดำเนินการแอนด์ (And) หรือออร์ (Or) กันอย่างไรก็ได้ แต่ในงานวิจัยนี้เงื่อนไขทั้งหมดในแต่ละกฎธุรกิจต้องแอนด์หรือออร์กันอย่างน้อยอย่างหนึ่งเท่านั้น ซึ่งกำหนดด้วยแท็กตัวดำเนินการตรรกะ (Logical operator)

3.1.2.11 แท็กตัวดำเนินการตรรกะ คือ แท็กที่กำหนดการดำเนินการตรรกะ (แอนด์หรือออร์) ระหว่างเงื่อนไขทั้งหมดในหนึ่งกฎ แท็กนี้มี 2 ประเภท คือ แท็ก and และ แท็ก or ถ้าเงื่อนไขทั้งหมดแอนด์กันแท็กตัวดำเนินการตรรกะจะเป็นแท็ก and ถ้าออร์กันจะเป็นแท็ก or แท็กนี้เป็นแท็กลูกของแท็ก conditions ภายในแท็กประกอบด้วยแท็ก condition ซึ่งแทนเงื่อนไขและเก็บค่าที่เกี่ยวข้องกับเงื่อนไขที่กำหนดโดยผู้ใช้


```

<conditions>
  <and>
    <condition id="1" order="2" conditionFactId="2">
      <lessThan variableId="1" leftId="1" rightId="43" />
    </condition>
    <condition id="3" order="3" conditionFactId="3">
      <between variableId="1" leftId="5.25" rightId="54" excludeLeft="False"
excludeRight="True" />
    </condition>
  </and>
</conditions>

```

รูปที่ 3.13 ตัวอย่างแท็ก conditions

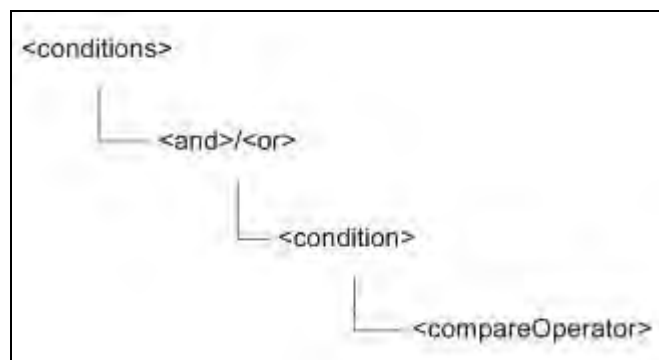
3.1.2.12 แท็ก condition เป็นแท็กที่เก็บค่าของตัวแปรปรับแต่งและตัวดำเนินการเปรียบเทียบ (Comparison operator) เช่น “มากกว่าเท่ากับ 25” เป็นต้น ซึ่งผู้ใช้ต้องกำหนด (Assign) ค่าของตัวแปร (เช่น “25”) และตัวดำเนินการเปรียบเทียบ (เช่น “มากกว่าเท่ากับ”) ลักษณะของค่าตัวแปรขึ้นอยู่กับแผนแบบเงื่อนไขที่เงื่อนไขอ้างอิงในแอตทริบิวต์ conditionFactId ทั้งค่าและตัวดำเนินการกำหนดด้วยแท็กตัวดำเนินการ (Operator tag)

3.1.2.13 แท็กตัวดำเนินการ ระบุค่าและตัวดำเนินการเปรียบเทียบ ที่ผู้ใช้กำหนดแก่ตัวแปรหนึ่งตัวของเงื่อนไขซึ่งมี 7 ประเภท ได้แก่ lessThan lessThanEqualTo moreThan moreThanEqualTo equals notEquals และ between ประเภทของแท็กตัวดำเนินการขึ้นอยู่กับตัวดำเนินการเปรียบเทียบที่ผู้ใช้กำหนด เช่น ถ้าผู้ใช้กำหนดตัวดำเนินการเปรียบเทียบเป็น “more than or equal to” แท็กตัวดำเนินการจะเป็นประเภท moreThanEqualTo เป็นต้น แท็กตัวดำเนินการประกอบด้วยแอตทริบิวต์ดังนี้

- 1) แอตทริบิวต์ variableId เป็นแอตทริบิวต์ที่อ้างอิง id ของตัวแปรที่แท็กตัวดำเนินการระบุค่า
- 2) แอตทริบิวต์ rightId คือ ค่าที่ระบุโดยผู้ใช้ เช่น 781 2.5 “bookstore” true เป็นต้น ประเภทของค่าขึ้นอยู่กับประเภทข้อมูลของตัวแปร
- 3) แอตทริบิวต์ leftId โดยทั่วไปแอตทริบิวต์นี้จะมีค่าเดียวกับแอตทริบิวต์ variableId แต่ถ้าตัวแปรเป็นข้อมูลตัวเลขและแท็กตัวดำเนินการเป็น between แอตทริบิวต์ leftId จะเป็นค่าด้านซ้ายของตัวดำเนินการเปรียบเทียบ between เช่น ถ้าผู้ใช้กำหนดค่าของตัวแปร OrderSize เป็น between 10 and 25 ค่า leftId จะเป็น 10 และค่า rightId จะเป็น 25

4) แอตทริบิวต์ `excludeLeft` และ `excludeRight` ใช้ในกรณีที่แท็กตัวดำเนินการเป็น `between` ระบุว่าค่าที่ขอบด้านซ้ายและด้านขวาของ `between` จะถูกรวมในเงื่อนไขด้วยหรือไม่ ตามลำดับ ถ้าแอตทริบิวต์ มีค่าเป็น `True` หมายถึง ไม่รวม แต่ถ้าเป็น `False` หมายถึง รวม เช่น ถ้ากำหนดค่าของตัวแปร `OrderSize` เป็น `between 10 and 25` และแอตทริบิวต์ `excludeLeft` เป็น `True` และแอตทริบิวต์ `excludeRight` เป็น `False` เมื่อแปลแท็กตัวดำเนินการออกมาเป็นเงื่อนไขจะหมายถึง $10 < OrderSize \text{ And } OrderSize \leq 25$

โครงสร้างภายในแท็ก `conditions` แสดงดังรูป 3.14



รูปที่ 3.14 โครงสร้างภายในแท็ก `conditions`

3.1.2.14 แท็ก `actions` เป็นแท็กที่เก็บการกระทำทั้งหมดของกฎธุรกิจ 1 กฎ และเป็นแท็กลูกของแท็ก `rule` ภายในแท็ก `actions` มีแท็ก `action` หลายแท็ก ซึ่งแต่ละแท็กเก็บค่าที่เกี่ยวข้องกับการกระทำซึ่งกำหนดโดยผู้ใช้ ดังตัวอย่างด้านล่าง

3.1.2.15 แท็ก `action` เป็นแท็กที่เก็บค่าของตัวแปรปรับแต่งของการกระทำ ดังได้กล่าวในส่วนไฟล์ข้อเท็จจริงว่าแผ่นแบบการกระทำจะมีตัวแปรปรับแต่งที่ผู้ใช้ต้องกำหนดค่า แอตทริบิวต์ `actionFactId` ระบุแผ่นแบบการกระทำที่เป็นแบบของการกระทำนี้ แต่ละค่าที่วันกำหนดด้วยแท็ก `variableValue`

3.1.2.16 แท็ก `variableValue` ทำหน้าที่กำหนดค่าของตัวแปรปรับแต่งของการกระทำ มีแอตทริบิวต์ `id` เพื่อระบุว่ากำหนดค่าของตัวแปรใดของแผ่นแบบการกระทำในไฟล์ข้อเท็จจริง ดังตัวอย่างรูปที่ 3.15 แท็กนี้สามารถกำหนดค่าได้ 3 แบบ คือ

```

<actions>
  <action id="6" actionFactId="6" order="6">
    <variableValue id="1" value="77" />
    <variableValue id="2" conditionFactId="2"
customVariableId="1" />
    <assign actionResultId="3" returnVariableId="4" />
  </action>
</actions>

```

รูปที่ 3.15 ตัวอย่างแท็ก variableValue ภายในแท็ก action

1) กำหนดเป็นค่าคงที่ เช่น 25 “Bookstore” true เป็นต้น การกำหนดทำโดยระบุค่าในแอตทริบิวต์ value

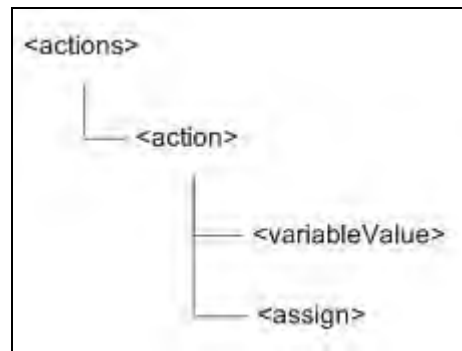
2) กำหนดโดยอ้างอิงถึงค่าของพารามิเตอร์ของตารางการตัดสินใจ ถ้าเป็นพารามิเตอร์จากตัวแปรปรับแต่งของเงื่อนไข กำหนดโดยระบุ id ของเงื่อนไขและตัวแปรด้วยแอตทริบิวต์ conditionFactId และ customVariableId ตามลำดับ ถ้าพารามิเตอร์เป็นแบบกำหนดโดยผู้ใช้กำหนดด้วยแอตทริบิวต์ dtParameterId

3) กำหนดโดยอ้างอิงถึงตัวแปรผลลัพธ์ของตารางการตัดสินใจ กำหนดโดยระบุ id ของด้วยแอตทริบิวต์ dtParameterId

3.1.2.17 แท็ก assign ทำหน้าที่กำหนดว่าจะนำผลลัพธ์ที่ได้จากการกระทำมากำหนดให้กับผลลัพธ์ตัวใดของตารางการตัดสินใจ ซึ่งแท็ก assign อาจไม่มีได้ ขึ้นอยู่กับผู้ใช้ต้องการนำผลลัพธ์ของการกระทำมาใช้หรือไม่ แอตทริบิวต์ actionResultId คือ id ของผลลัพธ์จากการกระทำ แอตทริบิวต์ returnVariableId คือ id ผลลัพธ์ตารางการตัดสินใจซึ่งส่งค่ากลับไปยังลูกข่ายที่เรียกใช้ตารางการตัดสินใจในรูปแบบเว็บเซอวิซ

ไฟล์เซตกฎธุรกิจประกอบด้วยกฎธุรกิจหลายๆ กฎธุรกิจที่สัมพันธ์กัน แต่ละกฎธุรกิจประกอบด้วยเงื่อนไขและการกระทำ ซึ่งมาจากแผ่นแบบเงื่อนไขและแผ่นแบบการกระทำในไฟล์ข้อเท็จจริง

โครงสร้างภายในแท็ก actions แสดงดังรูป 3.16



รูปที่ 3.16 โครงสร้างภายในแท็ก actions

3.1.3 กลุ่มแท็กกำหนดกฎธุรกิจ

กลุ่มแท็กกำหนดกฎธุรกิจทำหน้าที่เก็บข้อมูลเซตกฎธุรกิจทั้งหมด รวมทั้งข้อมูลเกี่ยวกับไฟล์ข้อเท็จจริงที่อ้างถึงโดยเซตกฎธุรกิจ ไฟล์เซตกฎธุรกิจและการแบ่งหมวดหมู่เซตกฎธุรกิจ ดังตัวอย่างในรูปที่ 3.17 เอกซ์เอ็มเอสซีมาของกลุ่มแท็กนี้แสดงในภาคผนวก ค กลุ่มแท็กกำหนดกฎธุรกิจประกอบด้วยแท็กดังนี้

- 1) แท็ก businessRules เป็นแท็กรากของไฟล์ เก็บข้อมูลทั้งหมดของระบบกฎธุรกิจ เช่น เซตกฎธุรกิจ หมวดหมู่ และไฟล์ข้อเท็จจริง เป็นต้น
- 2) แท็ก category แทนหมวดหมู่ของเซตกฎธุรกิจโดยหนึ่งแท็กแทนหนึ่งหมวดหมู่ มีแอตทริบิวต์ name และ description ระบุชื่อและคำบรรยายตามลำดับ
- 3) แท็ก facts แทนไฟล์ข้อเท็จจริงหนึ่งไฟล์ ซึ่งในระบบอาจมีได้หลายไฟล์ อย่างไรก็ตามหนึ่งเซตกฎธุรกิจสามารถอ้างถึงไฟล์ข้อเท็จจริงได้เพียงหนึ่งไฟล์นั้น มีแอตทริบิวต์ name และ location ระบุชื่อและชื่อไฟล์ตามลำดับ

```

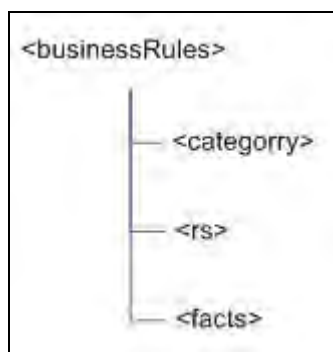
<businessRules>
  <category id="1" name="Discount Calculation"
description="Discount calculation of goods" />
  <facts id="1" name="factForDiscountCalcution"
location="predefined_facts_discount_subactions.xml" />
  <rs id="1" name="discount_book_rs" location="discount_rs.xml"
description="Book dicount policy" categoryId="1" factsId="1"
bpel="discount_book_rules_1.bpel" />
</businessRules>

```

รูปที่ 3.17 ตัวอย่างไฟล์กฎธุรกิจ

4) แท็ก rs แทนหนึ่งเซตกฎธุรกิจ แอตทริบิวต์ location ระบุชื่อไฟล์ที่เก็บเซตกฎธุรกิจ แอตทริบิวต์ bpel ระบุชื่อไฟล์บีเพลซึ่งแปลงมาจากเซตกฎธุรกิจนี้ นอกจากนี้ยังระบุหมวดหมู่และไฟล์ข้อเท็จจริงที่ใช้ในเซตกฎธุรกิจด้วยแอตทริบิวต์ categoryId และ factsId ตามลำดับ

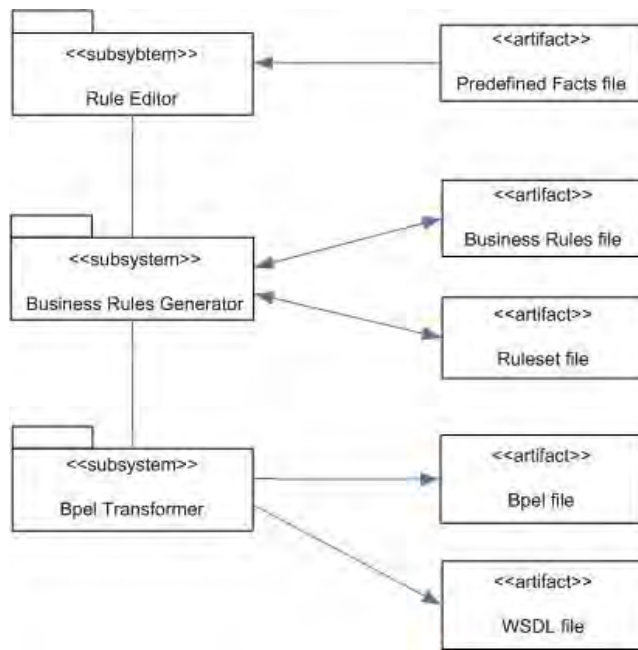
โครงสร้างภายในแท็ก businessRules แสดงดังรูป 3.18



รูปที่ 3.18 โครงสร้างภายในแท็ก businessRules

3.2 สถาปัตยกรรมระบบ

เพื่อให้ผู้ใช้สามารถเข้าใจกฎธุรกิจที่แสดงได้ งานวิจัยนี้ออกแบบภาษาบีอาร์ดีแอลเพื่อแสดงกฎธุรกิจในรูปของภาษารวมชาติ (Natural Language) ซึ่งใกล้เคียงกับภาษามนุษย์ทั่วไป โดยผู้ใช้สร้างกฎธุรกิจผ่านเครื่องมือบรรณาธิกรณกฎธุรกิจ (Rule Editor) ซึ่งสร้างกฎบนพื้นฐานของข้อเท็จจริงในไฟล์ข้อเท็จจริง (Predefined Facts file) ไฟล์นี้กำหนดเงื่อนไขและการกระทำที่ใช้ในกฎได้ จากนั้นกฎจะถูกเก็บลงไฟล์กฎธุรกิจ (Business Rules file) และไฟล์เซตกฎธุรกิจ (Ruleset file) และถูกแปลงเป็นไฟล์บีเพลในท้ายที่สุด ทั้งไฟล์ข้อเท็จจริง ไฟล์กฎธุรกิจ และไฟล์เซตกฎธุรกิจกำหนดด้วยภาษาบีอาร์ดีแอล ระบบการพัฒนามาตรฐานกฎธุรกิจประกอบด้วย 8 ส่วนสำคัญ ดังรูปที่ 3.19 มีรายละเอียดดังนี้



รูปที่ 3.19 องค์ประกอบของระบบสำหรับการพัฒนากฎธุรกิจ

3.2.1 เครื่องมือบรรณาธิกรณกฎธุรกิจ (Rule Editor)

เครื่องมือบรรณาธิกรณกฎธุรกิจเป็นเว็บแอปพลิเคชัน (Web Application) สำหรับให้นักวิเคราะห์สร้าง ลบ แก้ไขและจัดการกฎธุรกิจรวมทั้งเงื่อนไขและการกระทำ เครื่องมือนี้ยังสามารถสร้างตารางการตัดสินใจจากเงื่อนไขที่ผู้ใช้กำหนด ดังรูปที่ 3.20 นอกจากนี้ยังสามารถตรวจสอบความกำกวมของตารางได้ เครื่องมือนี้จะอ่านข้อมูลเกี่ยวกับเงื่อนไขและการกระทำจากไฟล์ข้อเท็จจริงและแสดงกฎธุรกิจในรูปภาพมาตรฐานชาติเพื่อให้ผู้ใช้สามารถแก้ไขกฎธุรกิจแต่ละกฎ ดังรูปที่ 3.21 รายละเอียดการใช้เครื่องมือบรรณาธิกรณกฎธุรกิจอธิบายในภาคผนวก

Condition			Action	
Pay with ?	Customer type is ?	The total is ?	Allow ?% discount	Bonus is ?
1. [Credit Card]	2. [member]	3. [more than or equal to 5000]	1. [10]	2. [voucher]
1. [Wire Transfer]	2. [member]	3. [more than or equal to 5000]	1. [5]	2. [voucher]
1. [Credit Card]	2. [member]	-	1. [10]	2. [alarm clock]
1. [Wire Transfer]	2. [member]	-	1. [5]	2. [alarm clock]
1. [Credit Card]	-	2. [more than or equal to 10000]	1. [5]	2. [alarm clock]
1. [Wire Transfer]	-	2. [more than or equal to 10000]	1. [0.0]	2. [alarm clock]
1. [Credit Card]	-	2. [more than or equal to 5000]	1. [5]	2. [pen]
1. [Wire Transfer]	-	2. [more than or equal to 5000]	1. [0.0]	2. [pen]
1. [Credit Card]	-	2. [more than or equal to 1000]	1. [5]	2. [key holder]
1. [Wire Transfer]	-	2. [more than or equal to 1000]	1. [0.0]	2. [key holder]
-	-	-	1. [0.0]	2. [No bonus]

รูปที่ 3.20 เครื่องมือบรรณาธิกรณกฎธุรกิจแสดงตารางการตัดสินใจ

Rule: rule 9		Group: Member group		Delete rule	
If Operation: And Change operation					
Condition			Edit	Delete	Change Order
1. Pay with Credit Card			Edit	Delete	Up Down
2. Customer type is member			Edit	Delete	Up Down
The total is <input type="text" value="more than or equal to"/> <input type="text" value="5000"/>			Ok Cancel	Delete	Up Down
<input type="button" value="Add Condition"/>					
Then					
Action			Edit	Delete	Change Order
1. Allow 10% discount			Edit	Delete	Up Down
2. Bonus is voucher			Edit	Delete	Up Down
<input type="button" value="Add Action"/>					
Edit: Name <input type="text" value="rule 9"/>		<input type="checkbox"/> Default rule	Group <input type="text" value="Member group"/>	<input type="button" value="Ok"/>	

รูปที่ 3.21 เครื่องมือบรรณาธิกรณกฎธุรกิจ

3.2.2 ตัวสร้างกฎธุรกิจ (Business Rules Generator)

เมื่อผู้ใช้สร้างกฎธุรกิจจากเครื่องมือบรรณาธิกรณกฎธุรกิจแล้ว ตัวสร้างกฎธุรกิจจะบันทึกกฎธุรกิจทั้งหมดลงในไฟล์เซตกฎธุรกิจและบันทึกข้อมูลเกี่ยวกับเซตกฎธุรกิจนั้น เช่น ชื่อ ชื่อไฟล์ เซตกฎธุรกิจ คำบรรยาย เป็นต้น ลงในไฟล์กฎธุรกิจเพื่อเก็บข้อมูลเกี่ยวกับกฎธุรกิจนั้นไว้ เมื่อผู้ใช้ต้องการดูกฎ หรือแก้ไขกฎในภายหลัง ตัวสร้างกฎธุรกิจจะอ่านข้อมูลกฎธุรกิจจากไฟล์เหล่านี้ มาแสดงแก่ผู้ใช้ในเครื่องมือบรรณาธิกรณกฎธุรกิจ รายละเอียดเกี่ยวกับไฟล์เซตกฎธุรกิจและไฟล์กฎธุรกิจจะกล่าวในภายหลัง

3.2.3 ตัวแปลงบีเพล (Bpel Transformer)

ตัวแปลงบีเพลทำหน้าที่รับข้อมูลกฎธุรกิจจากตัวสร้างกฎธุรกิจ เพื่อนำมาแปลงเป็นไฟล์บีเพล เมื่อผู้ใช้สร้างหรือแก้ไขกฎธุรกิจจากเครื่องมือบรรณาธิกรณกฎธุรกิจ เครื่องมือบรรณาธิกรณกฎธุรกิจจะส่งข้อมูลกฎธุรกิจที่ถูกแก้ไขมายังตัวสร้างกฎธุรกิจ สุดท้ายตัวสร้างกฎธุรกิจจะส่งข้อมูลกฎธุรกิจให้ตัวแปลงบีเพล เพื่อสร้างหรือแก้ไขไฟล์บีเพลของกฎธุรกิจนั้น รายละเอียดการแปลงจะกล่าวถึงในส่วนไฟล์บีเพล

3.2.4 ไฟล์ข้อเท็จจริง (Predefined Facts file)

ไฟล์ข้อเท็จจริง คือ ไฟล์เอกซ์เอ็มแอลที่ระบุข้อเท็จจริงได้แก่ เงื่อนไขและการกระทำที่สามารถใช้ในกฎธุรกิจได้ โดยระบุด้วยภาษาปีอาร์ดีแอลกลุ่มแท็กกำหนดข้อเท็จจริงซึ่งกล่าวในหัวข้อที่แล้ว ไฟล์ข้อเท็จจริงเป็นเหมือนแม่แบบ (Template) ของเงื่อนไขและการกระทำ โดย

```

<conditionTemplate id="16" name="Province Condition"
  description="Province of thailand">
  <naturalLanguage>
    The province is [operator] ?
  </naturalLanguage>
  <customVariables>
    <customVariable id="1" name="province" type="string" >
      <enum value="10" text="Krung Thep Maha Nakhon" />
      <enum value="83" text="Phuket" />
    </customVariable>
  </customVariables>
</conditionTemplate>

```

รูปที่ 3.22 ตัวอย่างแผ่นแบบเงื่อนไขในไฟล์ข้อเท็จจริง

ก่อนที่จะสร้างกฎธุรกิจนั้นต้องมีการกำหนดเงื่อนไข และการกระทำในไฟล์นี้ไว้ก่อน แผ่นแบบเป็นเหมือนวัตถุสำหรับสร้างกฎธุรกิจ ซึ่งโดยปกติทำโดยโปรแกรมเมอร์หรือผู้มีความรู้ด้านเว็บเซอวิสและการเขียนโปรแกรม เมื่อเงื่อนไขแสดงต่อผู้ใช้ จะแสดงในรูปประโยคภาษาธรรมชาติ เพื่อให้ผู้ใช้เข้าใจง่าย ตัวอย่างดังรูปที่ 3.22 เป็นตัวอย่างแผ่นแบบเงื่อนไข ที่กำหนดเงื่อนไขเกี่ยวกับจังหวัดที่ลูกจ้างทำงาน มีตัวแปรที่เกี่ยวข้องคือ รหัสจังหวัด และมีประโยคภาษาธรรมชาติคือ “The province is [operator] ?” ไฟล์ข้อเท็จจริงประกอบด้วย 2 ส่วน ได้แก่

3.2.4.1 แผ่นแบบเงื่อนไข

แผ่นแบบเงื่อนไขเป็นเงื่อนไขที่สามารถใช้ในกฎต่างๆ ได้ โดยในไฟล์ข้อเท็จจริงหนึ่งสามารถมีแผ่นแบบเงื่อนไขได้หลายอัน แต่ละแผ่นแบบเงื่อนไขจะประกอบด้วยตัวแปรปรับแต่ง (Custom variable) เช่น แผ่นแบบเงื่อนไข provinceCondition มีตัวแปรชื่อ province เมื่อแผ่นแบบเงื่อนไขนี้ถูกเพิ่มเข้าสู่กฎธุรกิจ ผู้ใช้ต้องกำหนดตัวดำเนินการเปรียบเทียบ (Compare operator) และค่าให้แก่ตัวแปร เช่น province = “Phuket” เป็นต้น นอกจากนี้แผ่นแบบยังมีภาษาธรรมชาติ (Natural language) ซึ่งแสดงเงื่อนไขแก่ผู้ใช้ในรูปประโยคภาษาอังกฤษ

3.2.4.2 แผ่นแบบการกระทำ

แผ่นแบบการกระทำ เป็นการกระทำที่สามารถใช้ในกฎต่างๆ ได้ โดยในไฟล์ข้อเท็จจริงหนึ่งสามารถมีแผ่นแบบการกระทำได้หลายอัน มีตัวแปรปรับแต่งที่ผู้ใช้ต้องกำหนดและ

ภาษาธรรมชาติเช่นเดียวกับแผ่นแบบเงื่อนไข แผ่นแบบการกระทำมี 2 ประเภท คือ แผ่นแบบการกระทำทั่วไป และแผ่นแบบการกระทำสำหรับเรียกตารางย่อย

1) แผ่นแบบการกระทำทั่วไป คือ แผ่นแบบที่ระบุการกระทำที่ไม่ใช่การเรียกตารางย่อย รูปแบบการกระทำหรือกิจกรรมมีการระบุชัดเจนในไฟล์ข้อเท็จจริง การกระทำประกอบด้วยกิจกรรมต่างๆ ซึ่งทำงานเป็นลำดับจากบนลงล่างจนครบ กิจกรรมในงานวิจัยนี้มี 2 ประเภท ได้แก่ การคืนค่ากลับไปยังตารางการตัดสินใจ และการเรียกใช้เว็บเซอร์วิส โดยกิจกรรมสามารถนำค่าของตัวแปรปรับแต่งที่ผู้ใช้กำหนด มาเป็นพารามิเตอร์ของการเรียกเว็บเซอร์วิส หรือคืนค่ากลับไปยังตารางการตัดสินใจได้

2) แผ่นแบบการกระทำสำหรับเรียกตารางย่อย แผ่นแบบการกระทำชนิดนี้ไม่มีตัวแปรปรับแต่งแต่อย่างใด เมื่อแผ่นแบบนี้ถูกเลือกเข้าสู่กฎธุรกิจ ผู้ใช้ต้องระบุตารางการตัดสินใจที่ต้องการเรียก และพารามิเตอร์ของตารางการตัดสินใจนั้น

3.2.5 ไฟล์เซตกฎธุรกิจ (Ruleset file)

ไฟล์เซตกฎธุรกิจเป็นไฟล์เอกซ์เอ็มแอลที่เก็บกฎธุรกิจในรูปภาษาปียาร์ดีแอล กลุ่มแท็กกำหนดเซตกฎธุรกิจซึ่งกล่าวในหัวข้อที่แล้ว เมื่อผู้ใช้สร้างหรือแก้ไขกฎธุรกิจจากเครื่องมือบรรณาธิกรณกฎธุรกิจ กฎธุรกิจจะถูกบันทึกลงไฟล์นี้ โดยหนึ่งไฟล์จะเก็บหนึ่งเซตกฎธุรกิจจากหนึ่งตารางการตัดสินใจ หรือหนึ่งไฟล์ต่อหนึ่งตารางการตัดสินใจนั่นเอง ดังตัวอย่างดังตัวอย่างในรูปที่ 3.23 ตารางการตัดสินใจชื่อ “discount_book_rules” ประกอบด้วยกฎต่างๆ เช่น กฎ “rule_1” เป็นต้น

```
<ruleset name="discount_book_rules" >
  <rule id="1" name="rule_1" groupId="1" default="False" order="1" >
    <conditions>...</conditions>
    <actions>...</actions>
  </rule>
  <rule>...</rule>
</ruleset>
```

รูปที่ 3.23 ตัวอย่างไฟล์เซตกฎธุรกิจของตารางการตัดสินใจ

3.2.6 ไฟล์กฎธุรกิจ (Business Rules file)

ไฟล์กฎธุรกิจเป็นไฟล์เอกซ์เอ็มแอลที่เขียนด้วยภาษาพีอาร์ดีแอล กลุ่มแท็กกำหนดกฎธุรกิจซึ่งกล่าวในหัวข้อที่แล้ว ทำหน้าที่เก็บข้อมูลเซตกฎธุรกิจทั้งหมด รวมทั้งข้อมูลเกี่ยวกับไฟล์ข้อเท็จจริงที่อ้างอิงโดยเซตกฎธุรกิจ และการแบ่งหมวดหมู่เซตกฎธุรกิจ จากหัวข้อที่แล้วจะเห็นว่าแต่ละไฟล์เซตกฎธุรกิจคือหนึ่งเซตกฎธุรกิจหรือหนึ่งตารางการตัดสินใจ แต่ในระบบการพัฒนาธุรกิจ อาจมีมากกว่าหนึ่งตาราง ดังนั้นจึงต้องมีการเก็บข้อมูลและความสัมพันธ์ของทุกตารางไว้เพื่อการจัดการตารางการตัดสินใจ ซึ่งข้อมูลที่ว่าเก็บในไฟล์กฎธุรกิจนั่นเอง

3.2.7 ไฟล์บีเพลและไฟล์บรรยายเว็บเซอร์วิส

เมื่อกฎธุรกิจจากเครื่องมือบรรณาธิกรณกฎธุรกิจถูกบันทึกลงไฟล์เซตกฎธุรกิจแล้ว ตัวแปลงบีเพลจะนำข้อมูลกฎธุรกิจที่ได้รับจากตัวสร้างกฎธุรกิจมาสร้างไฟล์บีเพล โดยหนึ่งไฟล์บีเพลก็คือหนึ่งเซตกฎธุรกิจหรือหนึ่งตารางการตัดสินใจ ไฟล์บีเพลที่ได้เมื่ออ่านโดยเครื่องประมวลผลบีเพล (BPEL engine) สามารถทำงานในรูปแบบเว็บเซอร์วิสที่ให้บริการกฎธุรกิจ งานวิจัยนี้เรียกเว็บเซอร์วิสจากไฟล์บีเพลที่มาจากตารางการตัดสินใจว่าเว็บเซอร์วิสกฎธุรกิจ

อย่างไรก็ตามการที่ไฟล์บีเพลจะทำงานในรูปแบบเว็บเซอร์วิสได้นั้น ต้องมีไฟล์บรรยายเว็บเซอร์วิส (WSDL) ซึ่งเขียนด้วยภาษาบรรยายเว็บเซอร์วิส ไฟล์นี้จะบรรยายละเอียดและที่อยู่ของเว็บเซอร์วิส ผู้เรียกใช้เว็บเซอร์วิสต้องอ่านข้อมูลจากไฟล์นี้ก่อนที่จะเรียกใช้เว็บเซอร์วิสได้ ดังนั้นเพื่อให้ตารางการตัดสินใจสามารถทำงานในรูปแบบเว็บเซอร์วิสที่สามารถให้บริการได้สมบูรณ์ จึงต้องมี 2 ไฟล์ที่สำคัญ คือ ไฟล์บีเพลและไฟล์บรรยายเว็บเซอร์วิส

3.2.7.1 ไฟล์บรรยายเว็บเซอร์วิส

ไฟล์บรรยายเว็บเซอร์วิสส่วนสำคัญในไฟล์บรรยายเว็บเซอร์วิส คือ ประเภทข้อมูลของพารามิเตอร์และผลลัพธ์ของเว็บเซอร์วิส ซึ่งกำหนดในส่วนเอกซ์เอ็มแอลสคีม่า ดังที่ได้กล่าวในบทที่แล้วว่าตารางการตัดสินใจจะมีพารามิเตอร์และตัวแปรผลลัพธ์ของตารางการตัดสินใจ ทั้งสองส่วนจะถูกแปลงเป็นประเภทข้อมูลสำหรับพารามิเตอร์และผลลัพธ์ของเว็บเซอร์วิสตามลำดับ เช่น ถ้าตารางการตัดสินใจคำนวณค่าขนส่งมีพารามิเตอร์ `timeOfArriVal` และ `product_category` ทั้งสองตัวแปรจะถูกระบุในอิลิเมนต์ในส่วนของประเภทข้อมูลของพารามิเตอร์เว็บเซอร์วิส โดยประเภทข้อมูลของพารามิเตอร์เว็บเซอร์วิส คือ `Standard_Shipment_Rate_ParamType` และประเภทข้อมูลของผลลัพธ์เว็บเซอร์วิส คือ `tandard_Shipment_Rate_ResultType` ซึ่งมีอิลิเมนต์ `price_per_shipment` `price_per_item` และ `price_per_weight` ที่มาจากผลลัพธ์ของตารางการตัดสินใจ ดังตัวอย่างเอกซ์เอ็มแอลสคีม่าในรูปที่ 3.24

ส่วนข้อมูลอื่นในไฟล์ เช่น ชื่อเว็บเซอวิส การดำเนิน เป็นต้น จะถูกสร้างชื่อและรหัสของ ตารางการตัดสินใจ ที่อยู่ของเว็บเซอวิสกำหนดเป็น localhost และกำหนดให้ใช้โปรโตคอล HTTP

```
<schema>
  <element name="Standard_Shipment_Rate_ParamType">
    <complexType><sequence>
      <element name="timeOfArrival" type="integer" />
      <element name="product_category" type="string" />
    </sequence></complexType>
  </element>
  <element name="Standard_Shipment_Rate_ResultType">
    <complexType><sequence>
      <element name="price_per_shipment" type="double" />
      <element name="price_per_item" type="double" />
      <element name="price_per_weight" type="double" />
    </sequence></complexType>
  </element>
</schema>
```

รูปที่ 3.24 ตัวอย่างเอกซ์เอ็มแอลสคีมาในไฟล์บรรยายเว็บเซอวิสจากตารางการตัดสินใจ

ในการรับส่งข้อมูล

3.2.7.2 ไฟล์บีเพล

ไฟล์บีเพลประกอบด้วยข้อความ If-Then ซึ่งแต่ละข้อความแทนกฎธุรกิจหนึ่งกฎ เงื่อนไข อยู่ในส่วนแท็ก condition และการกระทำอยู่ในส่วนแท็ก sequence ดังตัวอย่างในรูปที่ 3.25 การสร้างไฟล์บีเพลมีรายละเอียดดังนี้

1) พารามิเตอร์และผลลัพธ์ของเว็บเซอวิสกฎธุรกิจ

เมื่อไฟล์บีเพลถูกสร้างขึ้นจะประกาศตัวแปรสองตัวเพื่อรองรับค่าพารามิเตอร์จากผู้เรียกใช้ และเก็บผลลัพธ์ที่จะส่งคืนกลับไป โดยประเภทข้อมูลของทั้งคู่เป็นไปตามที่อธิบายในหัวข้อไฟล์บรรยายเว็บเซอวิสก่อนหน้า

2) การทำงานแบบวนซ้ำของเซตกฎธุรกิจ

เมื่อเซตกฎธุรกิจแปลงเป็นไฟล์บีเพลจะมีคำสั่ง while ครอบทั้งเซตกฎธุรกิจนั้นไว้ ในส่วนเงื่อนไขของการวนซ้ำขึ้นอยู่กับลักษณะการกำหนดรอบ ดังนี้

```

<process name="discount_book_rules_1" >
  <if name="rule_1">
    <condition>เงื่อนไขของกฎที่หนึ่ง</condition>
    <sequence>การกระทำของกฎที่หนึ่ง</sequence>
  <elseif>
    <condition>เงื่อนไขของกฎที่สอง</condition>
    <sequence>การกระทำของกฎที่สอง</sequence>
  </elseif>
  <elseif>...</elseif>
  <else>...</else>
</if>
</process>

```

รูปที่ 3.25 ตัวอย่างไฟล์บีเพิลที่ประกอบด้วยข้อความ If-Then

- ระบุจำนวนรอบแบบตายตัว มีการสร้างตัวแปรสำหรับนับรอบ เช่น ตัวแปร countLoop และกำหนดค่าเริ่มต้นของตัวแปรเท่ากับจำนวนรอบ เมื่อเซตกฎธุรกิจประมวลผลครบ 1 รอบ จึงลบค่าตัวแปรลง 1 สำหรับเงื่อนไขของ while กำหนดเป็น countLoop > 0 เพียงเท่านั้น เซตกฎธุรกิจจะทำงานแบบวนซ้ำเท่ากับจำนวนรอบที่กำหนด ดังตัวอย่างหน้าในรูปที่ 3.26

- กำหนดด้วยเงื่อนไขโดยใช้ตัวแปรผลลัพธ์ของตารางการตัดสินใจ การแปลงเป็นภาษาบีเพิลทำได้ สร้างนิพจน์บูลีนจากตัวแปรผลลัพธ์ที่เงื่อนไขอ้างอิง เช่น เงื่อนไขพารามิเตอร์ param_1 ตัวดำเนินการเปรียบเทียบ คือ less than or equal to และค่าคือ 7 ดังนั้นนิพจน์บูลีนคือ param_1 <= 7

3) การแปลงเงื่อนไขในไฟล์เซตกฎธุรกิจเป็นภาษาบีเพิล

เมื่อสร้างนิพจน์บูลีนครบทุกเงื่อนไขแล้ว นำนิพจน์บูลีนเหล่านั้นมาเชื่อมด้วยตัวดำเนินการตรรกะ (แอนด์หรือออร์) จากนั้นนำตัวแปรปรับแต่งจากแผ่นแบบเงื่อนไข และค่าและตัวดำเนินการเปรียบเทียบที่ผู้ใช้กำหนดไว้มาสร้างนิพจน์บูลีน เช่น ตัวแปรปรับแต่งชื่อ OrderSize ตัวดำเนินการเปรียบเทียบ คือ less than และค่าคือ 25 ดังนั้นนิพจน์บูลีนคือ OrderSize < 25 เมื่อสร้างนิพจน์บูลีนครบทุกเงื่อนไขแล้ว นำนิพจน์บูลีนเหล่านั้นมาเชื่อมด้วยตัวดำเนินการตรรกะ ทั้งนี้ถ้ากฎธุรกิจกำหนดให้เงื่อนไขทั้งหมดแอนด์กันก็เชื่อมด้วย and ถ้ากฎธุรกิจกำหนดให้เงื่อนไขทั้งหมด

```

<assign name="Assign_Loop_Count_19">
  <copy>
    <from>7</from>
    <to>$count</to>
  </copy>
</assign>
<while>
  <condition>$count > 0</condition>
  <sequence>
    <if> ...</if>
    <assign name="Decrease_Loop_Count_20">
      <copy>
        <from>$count - 1</from>
        <to>$count</to>
      </copy>
    </assign>
  </sequence>
</while>

```

รูปที่ 3.26 ตัวอย่างไฟล์บีเพลที่เป็นการทำงานแบบวนซ้ำ

ออร์กันก็เชื่อมด้วย or เช่น OrderSize < 25 or CustomerType == "libray" or IsPremiumCustomer== true เป็นต้น

4) การแปลงการกระทำจากไฟล์เซตกฎธุรกิจเป็นภาษาบีเพล

การแปลงการกระทำของกฎธุรกิจหนึ่งในไฟล์เซตกฎธุรกิจเป็นภาษาบีเพล สำหรับการกระทำทั่วไป (ที่ไม่ใช่การเรียกตารางย่อย) ทำโดยการอ่านค่าของตัวแปรปรับแต่งของการกระทำที่กำหนดในไฟล์เซตกฎธุรกิจ และแผนแบบการกระทำของการกระทำนั้น จากนั้นประกาศตัวแปรในไฟล์บีเพลเพื่อเก็บค่าเหล่านั้น การจะนำค่าตัวแปรไปใช้ทำอะไรขึ้นอยู่กับกิจกรรมที่กำหนดในแผนแบบการกระทำ ลำดับของการกระทำในไฟล์บีเพลขึ้นกับลำดับของการกระทำที่กำหนดในไฟล์เซตกฎธุรกิจ ดังที่เคยกล่าวในหัวข้อไฟล์ข้อเท็จจริงว่ากิจกรรมมี 2 ประเภท คือ การคืนค่ากลับไปยังกฎ

ธุรกิจที่เรียกการกระทำนั้น และการเรียกใช้เว็บเซอร์วิส การแปลงกิจกรรมเหล่านั้นเป็นภาษาบีเพล มีรายละเอียดดังนี้

- การคืนค่ากลับไปยังกฎธุรกิจ

การเปลี่ยนกิจกรรมการคืนค่าให้อยู่ในรูปภาษาบีเพล ทำโดยประกาศตัวแปรชั่วคราวและนำค่าที่ต้องการคืนเก็บไว้ในตัวแปรชั่วคราวนั้น ถ้าในกฎธุรกิจกำหนดให้ค่าที่คืนจากการกระทำนั้น ถูกกำหนดให้แก่ผลลัพธ์ของตารางการตัดสินใจ จะทำการสำเนาค่าจากตัวแปรชั่วคราวนั้นสู่อัตโนมัติของผลลัพธ์ของเว็บเซอร์วิสกฎธุรกิจ

- การเรียกใช้เว็บเซอร์วิส

การเปลี่ยนกิจกรรมการเรียกใช้เว็บเซอร์วิสให้อยู่ในรูปภาษาบีเพล ทำโดยใช้คำสั่งเรียกเว็บเซอร์วิสซึ่งอยู่แล้วในภาษาบีเพล จากนั้นกำหนดพารามิเตอร์และการเก็บผลลัพธ์ของเว็บเซอร์วิสนั้นตามที่กำหนดในกิจกรรม ดังตัวอย่างในรูป 3.27

จากตัวอย่างแท็ก assign แรกเป็นการ กำหนดค่าให้พารามิเตอร์ของเว็บเซอร์วิส จากนั้นเรียกเว็บเซอร์วิสด้วยแท็ก invoke โดยระบุแอตทริบิวต์ที่จำเป็น เช่น ชื่อ operation ของเว็บเซอร์วิส ชื่อตัวแปรพารามิเตอร์ (แอตทริบิวต์ inputVariable) และชื่อตัวแปรที่เก็บผลลัพธ์ (แอตทริบิวต์ outputVariable) แท็ก assign สุดท้ายนำค่าผลลัพธ์จากเว็บเซอร์วิสเก็บไว้ในตัวแปร เพื่อนำไปใช้ในกิจกรรมอื่นต่อไป

- การเรียกใช้ตารางย่อย

ตารางการตัดสินใจแต่ละตารางทำงานในรูปเว็บเซอร์วิส ดังนั้นการเรียกใช้ตารางย่อยจึงเป็นการเรียกใช้เว็บเซอร์วิสอันหนึ่ง เมื่อแปลงการกระทำที่เรียกใช้ตารางย่อยเป็นภาษาบีเพล จึงใช้คำสั่งเรียกเว็บเซอร์วิสเช่นเดียวกับที่กล่าวในหัวข้อก่อนหน้า การกำหนดพารามิเตอร์และการจัดการผลลัพธ์ของเว็บเซอร์วิสตารางย่อย เป็นไปตามที่กำหนดในไฟล์เซตกฎธุรกิจ

```

<assign name="Assign_webservice_parameter">
  <copy>
    <from>${DtOperationIn.DtParam/wage-per-month}</from>
    <to>${calWebserviceIn.parameters/wagePerMonth}</to>
  </copy>
  <copy>
    <from>${DtOperationIn.DtParam/ unworkableDays}</from>
    <to>${calWebserviceIn.parameters/unworkableDay}</to>
  </copy>
</assign>
<invoke name="calCompensationFeeForUnworkable2"
operation="calCompensationFeeForUnworkable"
partnerLink="CalWebServiceLinkTypePL" portType="CalWebService"
inputVariable="calWebserviceIn" outputVariable="calWebserviceOut" />
<assign name="Assign_result_from_webservice">
  <copy>
    <from>${calWebserviceOut.parameters/return}</from>
    <to>${temp_result_webservice}</to>
  </copy>
</assign>

```

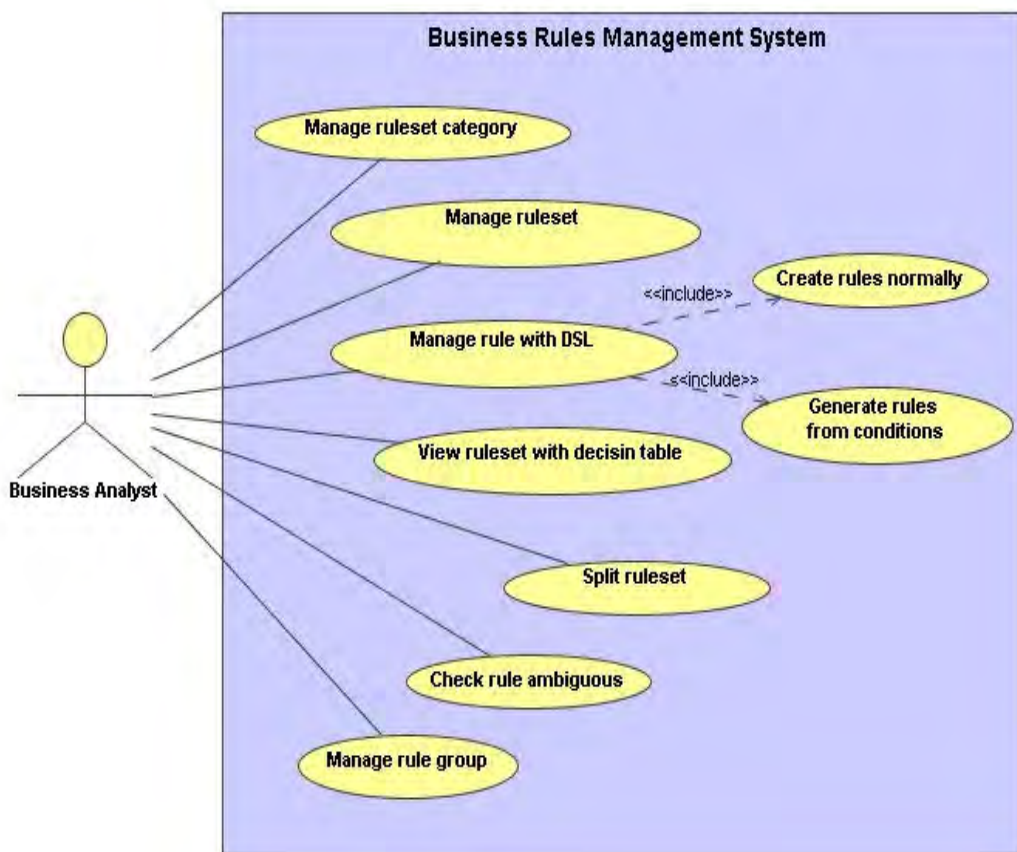
รูปที่ 3.27 ตัวอย่างไฟล์บีเพลที่มีเรียกใช้เว็บเซอร์วิส

3.3 การออกแบบการทำงานของระบบ

ระบบที่พัฒนาประกอบด้วยการทำงานตามแผนภาพยูสเคส ดังรูปที่ 3.28 ระบบมีความสามารถตามยูสเคสดังนี้

3.3.1 การจัดการประเภทเซตกฎธุรกิจ (Manage ruleset category) ผู้ใช้สามารถแบ่งประเภทของเซตกฎธุรกิจเป็นหมวดหมู่ได้ เช่น หมวดนโยบายด้านการขาย หมวดนโยบายด้านทรัพยากรมนุษย์ เป็นต้น

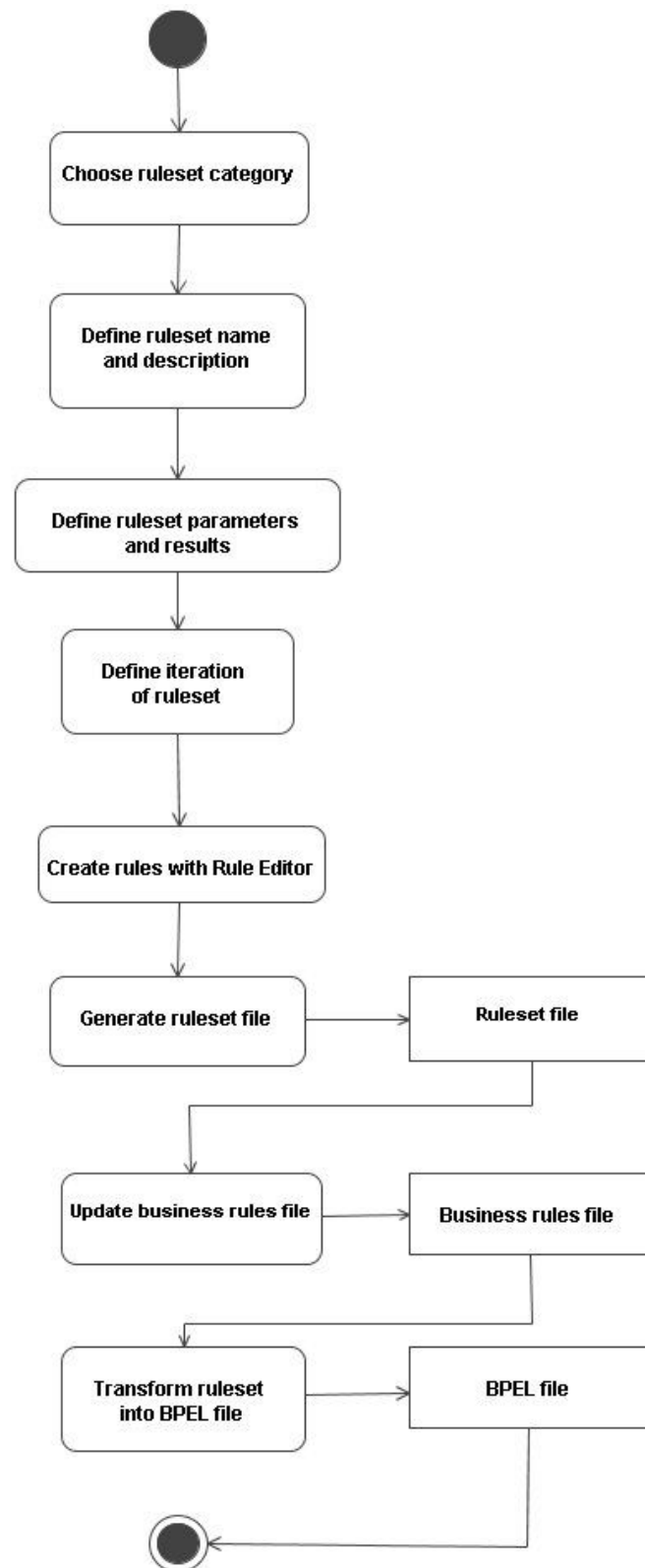
3.3.2 การจัดการเซตกฎธุรกิจ (Manage ruleset) ผู้ใช้สามารถดูสร้าง แก้ไขและลบเซตกฎธุรกิจ การสร้างเซตกฎธุรกิจมีขั้นตอนดังแผนภาพกิจกรรมของรูปที่ 3.29



รูปที่ 3.28 แผนภาพยูสเคสแสดงการทำงานของระบบ

จากรูปที่ 3.29 การทำงานเริ่มจากผู้ใช้เลือกประเภทเซตกฎธุรกิจ (Ruleset category) จากนั้น สร้างเซตกฎธุรกิจ โดยกำหนดชื่อและคำบรรยาย ต่อมากำหนดพารามิเตอร์และผลลัพธ์ของเซตกฎธุรกิจ ถ้าต้องการให้เซตกฎธุรกิจทำงานแบบวนซ้ำ จะต้องกำหนดให้เซตกฎธุรกิจทำงานแบบวนซ้ำพร้อมกับเงื่อนไขของการวนซ้ำ เมื่อกำหนดคุณสมบัติของเซตกฎธุรกิจแล้ว ก็ทำ

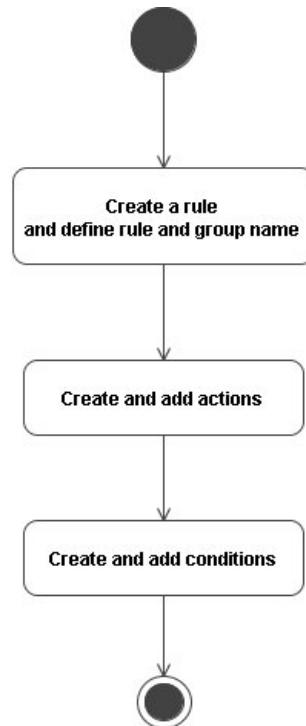
การสร้างกฎธุรกิจ สุดท้ายเซตกฎธุรกิจและกฎธุรกิจจะบันทึกลงไฟล์เซตกฎธุรกิจ และถูกแปลงเป็นไฟล์บีเพล



รูปที่ 3.29 แผนภาพกิจกรรมการสร้างเซตกฎธุรกิจ

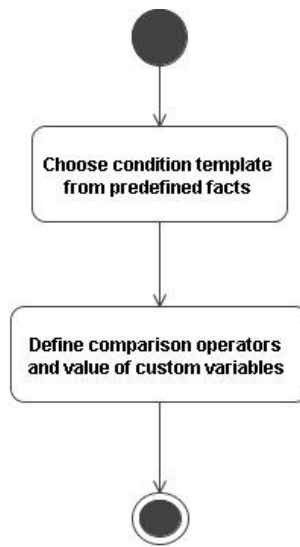
3.3.3 การจัดการกฎธุรกิจด้วยเครื่องมือบรรณาธิกรณกฎธุรกิจ (Manage rule with Rule Editor) ผู้ใช้สามารถสร้าง ดู แก้ไขและลบกฎธุรกิจด้วยเครื่องมือบรรณาธิกรณกฎธุรกิจ การสร้างกฎธุรกิจมี 2 วิธี คือ การสร้างกฎธุรกิจแบบปกติ และการสร้างกฎธุรกิจจากเงื่อนไขที่กำหนด

3.3.3.1 การสร้างกฎธุรกิจแบบปกติ (Create rules normally) เป็นการสร้างกฎธุรกิจที่ละกฎ โดยผู้ใช้จะสร้างเงื่อนไขและการกระทำในแต่ละกฎด้วยตัวเอง มีขั้นตอนดังแผนภาพกิจกรรมของรูปที่ 3.30

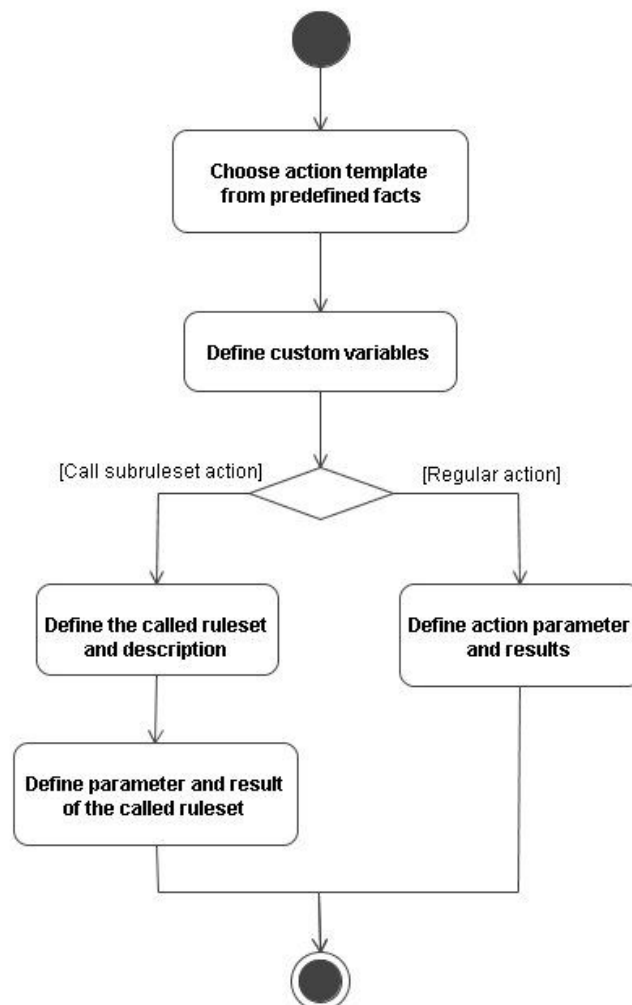


รูปที่ 3.30 แผนภาพกิจกรรมการสร้างกฎธุรกิจ

จากรูปที่ 3.30 การสร้างกฎธุรกิจทำโดยระบุชื่อกฎและกลุ่ม จากนั้นสร้างเงื่อนไขและการกระทำ การสร้างเงื่อนไขและการกระทำเป็นดังแผนภาพกิจกรรมของรูปที่ 3.31 และ 3.32 ตามลำดับ



รูปที่ 3.31 แผนภาพกิจกรรมการสร้างเงื่อนไขในกฎธุรกิจ



รูปที่ 3.32 แผนภาพกิจกรรมการสร้างการกระทำในกฎธุรกิจ

จากรูปที่ 3.31 เมื่อผู้ใช้เพิ่มเงื่อนไขในกฎธุรกิจจะต้องเลือกแผ่นแบบเงื่อนไข (Condition template) จากไฟล์ข้อเท็จจริง จากนั้นกำหนดตัวดำเนินการเปรียบเทียบ เช่น มากกว่า น้อยกว่า เป็นต้น และตัวแปรที่เกี่ยวข้องกับเงื่อนไขนั้น รายละเอียดเกี่ยวกับการกำหนดเงื่อนไขจะกล่าวในบทต่อไป

จากรูปที่ 3.32 เมื่อผู้ใช้เพิ่มการกระทำในกฎธุรกิจจะต้องเลือกแผ่นแบบการกระทำ (Action template) จากไฟล์ข้อเท็จจริง ถ้าการกระทำนั้นเป็นการเรียกใช้เซตกฎธุรกิจอื่นหรือตารางการตัดสินใจอื่น จะต้องระบุชื่อเซตกฎธุรกิจนั้นและคำบรรยาย จากนั้นกำหนดค่าของพารามิเตอร์และผลลัพธ์ของเซตกฎธุรกิจที่เรียก ถ้าการกระทำเป็นการกระทำทั่วไป กำหนดค่าของพารามิเตอร์และผลลัพธ์ของการกระทำนั้น รายละเอียดเกี่ยวกับการกำหนดการกระทำจะกล่าวในบทต่อไป

3.3.3.2 การสร้างกฎธุรกิจจากเงื่อนไขที่กำหนด (Generate rules from conditions) ผู้ใช้สามารถสร้างกฎธุรกิจจากเงื่อนไขที่กำหนดได้ โดยผู้ใช้เลือกเงื่อนไขจากแผ่นแบบเงื่อนไข และกำหนดตัวดำเนินการเปรียบเทียบและค่าเริ่มต้นของตัวแปรปรับแต่ง เมื่อผู้ใช้เลือกเงื่อนไขครบแล้ว เครื่องมือบรรณาธิกรณกฎธุรกิจจะสร้างกฎธุรกิจทั้งหมดจากทุกกรณีที่เป็นไปได้ของค่าเริ่มต้นของตัวแปรปรับแต่ง เช่น เงื่อนไขประเภทลูกค้ามีค่าเริ่มต้นที่ผู้ใช้กำหนด คือ เป็นสมาชิกหรือไม่เป็นสมาชิก และเงื่อนไขเวลาในการส่งสินค้ามีค่าเริ่มต้น คือ 2 วันหรือมากกว่า 2 วัน

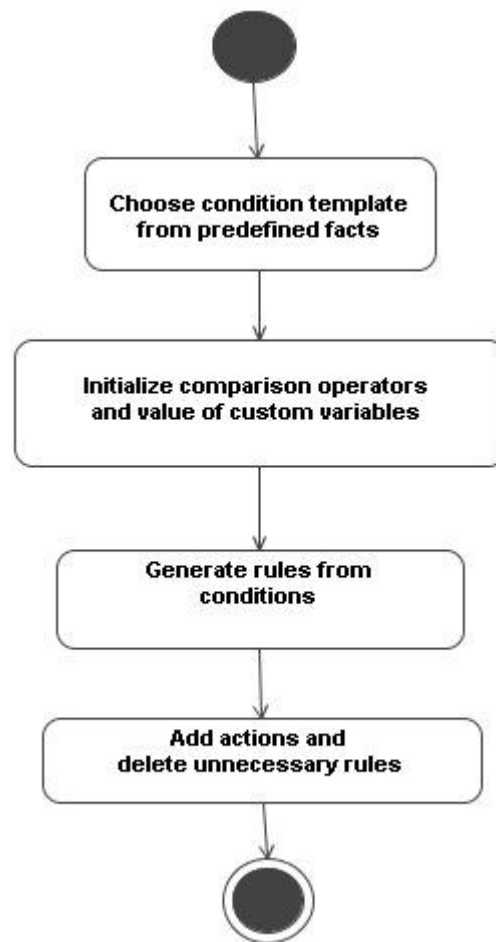
ผู้ใช้สามารถเลือกที่จะให้นำเงื่อนไขที่ยกตัวอย่าง มาสร้างกฎธุรกิจจากทุกกรณีที่เป็นไปได้ของเงื่อนไขเหล่านั้น เมื่อสร้างกฎธุรกิจจากเงื่อนไขที่กล่าวจะได้กฎธุรกิจทั้งหมด 16 กฎ หรือ 2^N โดย N จำนวนค่าเริ่มต้นของเงื่อนไขทั้งหมด ในที่นี้ค่าเริ่มต้นของเงื่อนไขแรก มี 2 ค่า คือ เป็นสมาชิกหรือไม่เป็นสมาชิก และค่าเริ่มต้นของเงื่อนไขที่สองมี 2 ค่าเช่นกัน คือ 2 วันหรือมากกว่า 2 วัน ดังนั้นจึงมีจำนวนค่าเริ่มต้นของเงื่อนไขทั้งหมดเท่ากับ $2+2 = 4$ ดังนั้นจะมีกฎธุรกิจทั้งหมดที่เป็นไปได้คือ $2^4 = 16$ กฎ ซึ่งแต่ละกฎจะมีเงื่อนไขแบบต่างๆ แตกต่างกัน

ตารางที่ 3.1 แสดงทุกกฎที่เป็นไปได้ เครื่องหมาย “X” หมายถึง กฎมีเงื่อนไขนั้น เครื่องหมาย “-” หมายถึง กฎไม่มีเงื่อนไขนั้น เช่น กฎที่ 7 มีเงื่อนไข คือ ลูกค้าเป็นสมาชิกและเวลาในการส่งสินค้า 2 วัน กฎที่ 10 มีเงื่อนไขคือ ลูกค้าไม่เป็นสมาชิก และเวลาในการส่งสินค้ามากกว่า 2 วัน กฎที่ 12 มีเงื่อนไขลูกค้าเป็นสมาชิก เป็นต้น รายละเอียดการใช้เครื่องมือเพื่อสร้างกฎธุรกิจจากเงื่อนไขที่กำหนดแสดงในภาคผนวก ง หน้า 116

กฎธุรกิจทั้งหมดที่สร้างขึ้นมีแต่เงื่อนไขเท่านั้น ผู้ใช้สามารถเพิ่มการกระทำ หรือลบกฎธุรกิจที่ไม่จำเป็นภายหลังได้ ดังแผนภาพกิจกรรมรูป 3.33

ตารางที่ 3.1 กฎธุรกิจที่สร้างจากเงื่อนไขที่กำหนดโดยผู้ใช้

กฎ	เงื่อนไข			
	ลูกค้าเป็นสมาชิก	ลูกค้าไม่เป็นสมาชิก	ส่งสินค้า 2 วัน	ส่งสินค้ามากกว่า 2 วัน
1	X	X	X	X
2	-	X	X	X
3	X	-	X	X
4	X	X	-	X
5	X	X	X	-
6	X	X	-	-
7	X	-	X	-
8	X	-	-	X
9	-	X	X	-
10	-	X	-	X
11	-	-	X	X
12	X	-	-	-
13	-	X	-	-
14	-	-	X	-
15	-	-	-	X
16	-	-	-	-



รูปที่ 3.33 แผนภาพกิจกรรมการสร้างกฎธุรกิจจากเงื่อนไขที่กำหนด

3.3.4 การดูเซตกฎธุรกิจในรูปตารางการตัดสินใจ (View ruleset with decision table) ผู้ใช้สามารถดูและตรวจสอบกฎธุรกิจ โดยแสดงเซตกฎธุรกิจในรูปตารางการตัดสินใจ

3.3.5 การแบ่งเซตกฎธุรกิจ (Split ruleset) ในกรณีที่กฎธุรกิจในเซตกฎธุรกิจเพิ่มมากขึ้น และอาจมีกฎธุรกิจที่ไม่สัมพันธ์กับกฎธุรกิจอื่น ผู้ใช้สามารถแยกกฎธุรกิจเหล่านั้นไปเป็นอีกเซตกฎธุรกิจหนึ่งได้

3.3.6 การตรวจสอบความกำกวม (Check rule ambiguous) ผู้ใช้สามารถตรวจสอบความกำกวมของตารางการตัดสินใจได้ว่า มีกฎธุรกิจใดที่มีเงื่อนไขเหมือนกันหรือไม่

3.3.7 การจัดกลุ่มกฎธุรกิจ (Manage rule group) ผู้ใช้สามารถแบ่งกลุ่มกฎธุรกิจภายในตารางการตัดสินใจได้ และสามารถเลือกดูกฎธุรกิจเฉพาะบางกลุ่มได้ เช่น ตารางการตัดสินใจที่กำหนดกฎธุรกิจในขึ้นเงินเดือน อาจแบ่งกลุ่มเป็น กลุ่มผู้บริหาร กลุ่มผู้จัดการ กลุ่มพนักงานทั่วไป เป็นต้น

3.4 แผนภาพคลาส

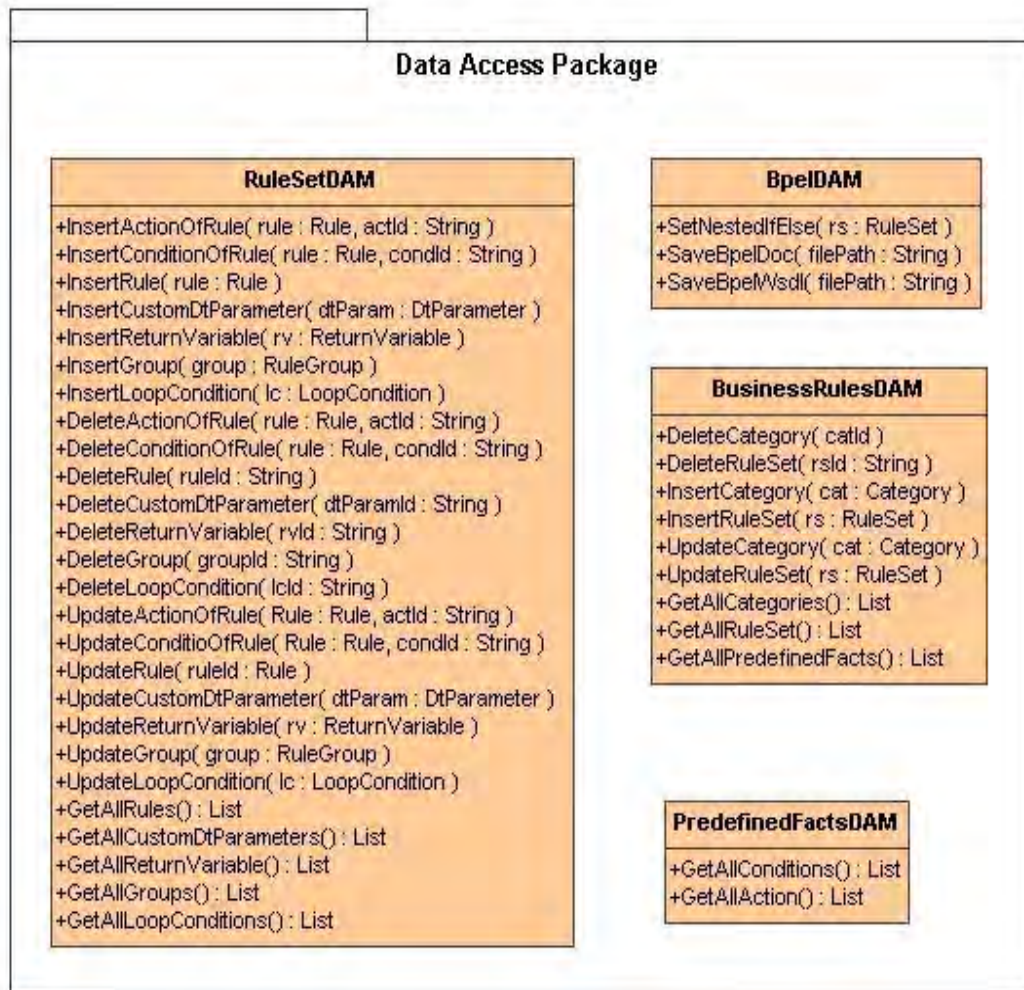
ระบบที่ออกแบบแบ่งคลาสออกเป็น 3 แพคเกจ ได้แก่

- แพคเกจ Presentation ประกอบด้วยคลาสสำหรับแสดงส่วนติดต่อผู้ใช้ (User Interface) และเว็บเพจต่างๆ
 - แพคเกจ Business Logic ประกอบด้วยคลาสที่เก็บข้อมูลเซตกฎธุรกิจและข้อมูลอื่นๆที่เกี่ยวข้องในระบบการพัฒนากฎธุรกิจ
 - แพคเกจ Data Access ประกอบด้วยคลาสที่ทำหน้าที่สร้าง แก้ไขและบันทึกข้อมูลจากคลาสในแพคเกจ Business Logic ลงไฟล์เอกซ์เอ็มแอล
- ความสัมพันธ์ของทั้ง 3 แพคเกจ แสดงดังรูปที่ 3.34

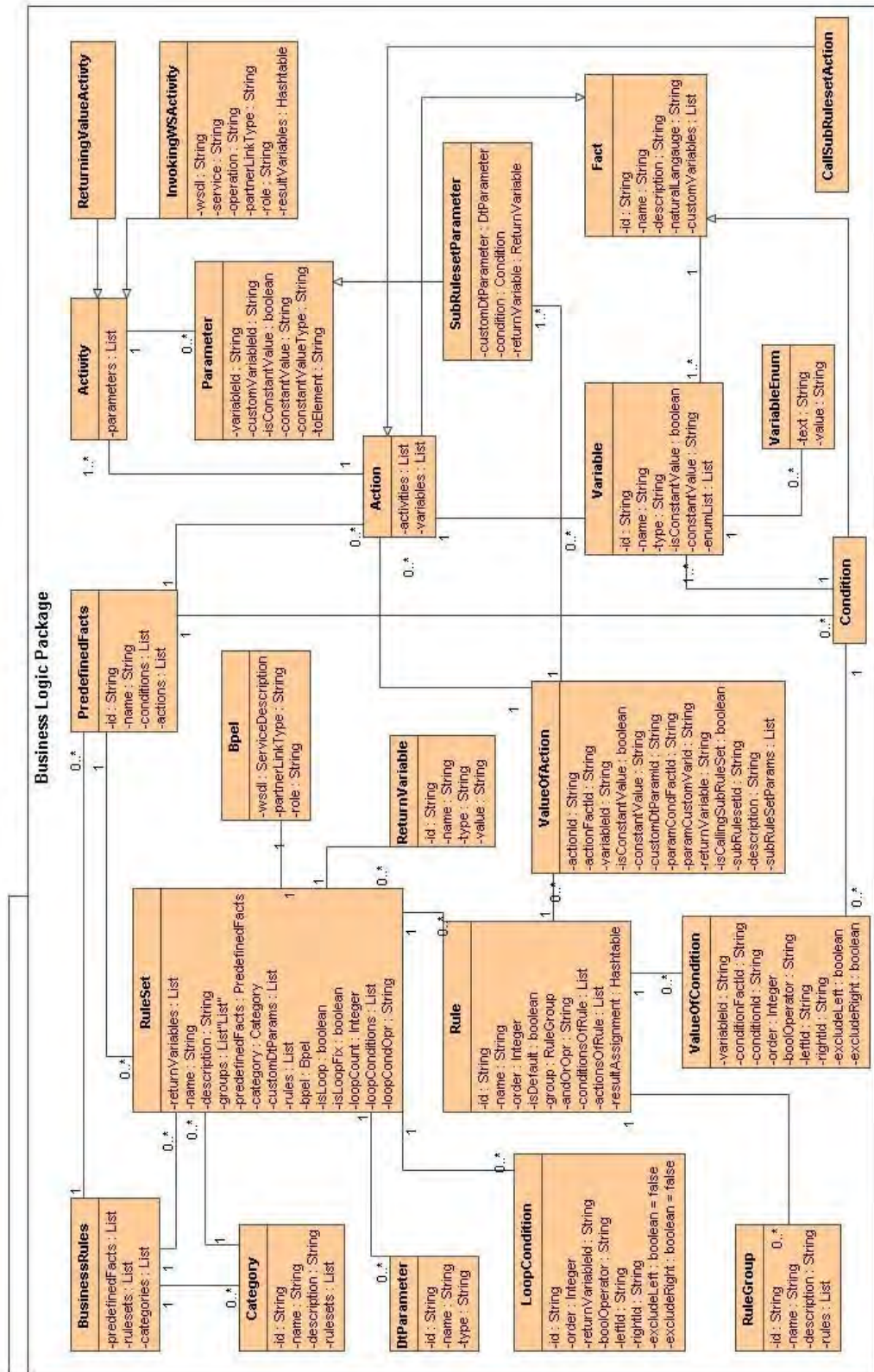


รูปที่ 3.34 ความสัมพันธ์ของแพคเกจของคลาส

ความสัมพันธ์ของคลาสมภายในแพคเกจ Business Logic สามารถแสดงได้ดังรูปที่ 3.36 ซึ่งรายละเอียดแสดงในภาคผนวก ก และรายละเอียดของคลาสมภายในแพคเกจ Data Access สามารถแสดงได้ดังรูปที่ 3.35 ซึ่งรายละเอียดแสดงในภาคผนวก ข



รูปที่ 3.35 ความสัมพันธ์และคุณสมบัติของคลาสในแพ็คเกจ Data Access



รูปที่ 3.36 ความสัมพันธ์และคุณสมบัติของคลาสในแพ็คเกจ Business Logic

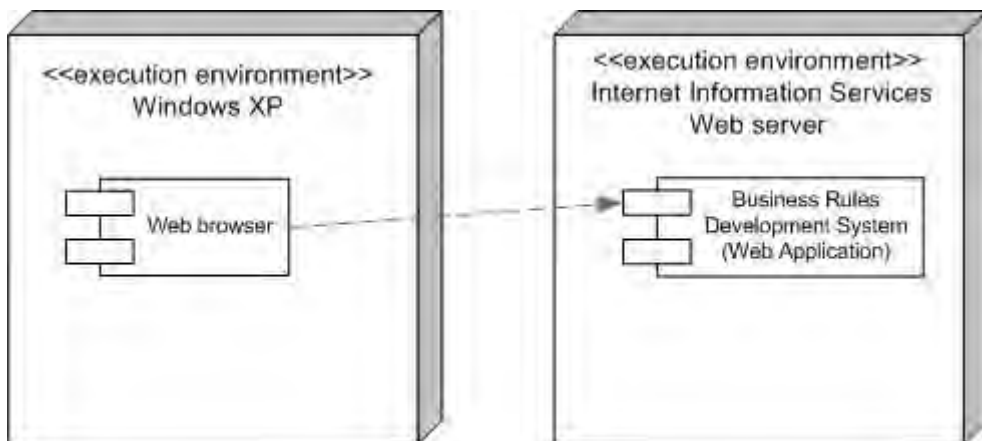
บทที่ 4 การพัฒนาและการทดสอบ

บทนี้จะกล่าวถึงการพัฒนาและทดสอบ งานวิจัยนี้ยกกรณีศึกษาที่เกี่ยวข้องกับกระบวนการธุรกิจและกฎธุรกิจมา 2 กรณี เพื่อทดสอบการทำงานของระบบการพัฒนากฎธุรกิจที่สร้างขึ้น รวมทั้งทดสอบไฟล์บีเพลผลลัพท์ที่ได้จากตารางการตัดสินใจว่าทำงานถูกต้องและสามารถทำงานร่วมกับกระบวนการธุรกิจอื่นๆ ได้

4.1 การพัฒนา

4.1.1 เครื่องมือและสภาพแวดล้อมในการพัฒนา

ระบบที่พัฒนาในงานวิจัยนี้ใช้โปรแกรมมิงลิ่งสตูดิโออินเทอร์เน็ต 2005 เป็นเครื่องมือในการพัฒนา โดยใช้ภาษาซีชาร์ป (C#) ระบบที่สร้างขึ้นทำงานเป็นเว็บแอปพลิเคชัน ที่ทำงานบนเครื่องบริการเว็บอินเทอร์เน็ตอินฟอร์เมชันเซอร์วิส (Internet Information Services) ดังแผนภาพดีพลอยเมนต์ในรูปที่ 4.1



รูปที่ 4.1 แผนภาพดีพลอยเมนต์

4.2 กรณีทดสอบที่ 1 : การคำนวณเงินทดแทน

กรณีศึกษาแรกเกี่ยวข้องกับกระบวนการธุรกิจ และกฎธุรกิจของจ่ายค่าเงินทดแทน ในกรณีศึกษานี้มีกระบวนการธุรกิจที่คำนวณหาจำนวนเงินทดแทนมากที่สุดที่เป็นไปได้ที่ต้องจ่ายให้ลูกจ้าง 1 คน และมีตารางการตัดสินใจที่เกี่ยวข้องกับกระบวนการธุรกิจ 3 ตารางประกอบด้วยตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้ และตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด

4.2.1 ลักษณะเงินทดแทน

เงินทดแทนคือ เงินที่ลูกจ้างได้รับจากกองทุนเงินทดแทนหรือนายจ้าง ในกรณีที่ลูกจ้างประสบอันตรายหรือเจ็บป่วย หรือตายเนื่องจากการทำงาน [24] ประกอบด้วย

- 1) ค่ารักษาพยาบาล ไม่เกิน 30,000 บาทต่อการเจ็บป่วยหนึ่งครั้ง
- 2) ค่าฟื้นฟูสมรรถภาพในการทำงาน ในกรณีที่ลูกจ้าง

- สูญเสียอวัยวะบางส่วน
- สูญเสียสมรรถภาพในการทำงานของอวัยวะ
- ทุพพลภาพ

จ่ายไม่เกิน 20,000 บาท

- 3) ค่าทำศพ จ่ายเป็นจำนวนเงิน 100 เท่าของอัตราสูงสุดของค่าจ้างขั้นต่ำรายวัน

- 4) ค่าทดแทน เป็นการจ่ายทดแทนเป็นรายเดือนให้แก่ลูกจ้าง ซึ่งขึ้นอยู่กับ

ลักษณะของการประสพภัย เงินทดแทนที่ได้รับจะแตกต่างกันไปตามอัตราของแต่ละลักษณะของการประสพภัย เป็นกรณีๆ ไปดังนี้

- กรณีลูกจ้างไม่สามารถทำงานได้ ถ้าทำงานไม่ได้เกิน 3 วัน จ่ายร้อยละ 60 ของค่าจ้างรายเดือน

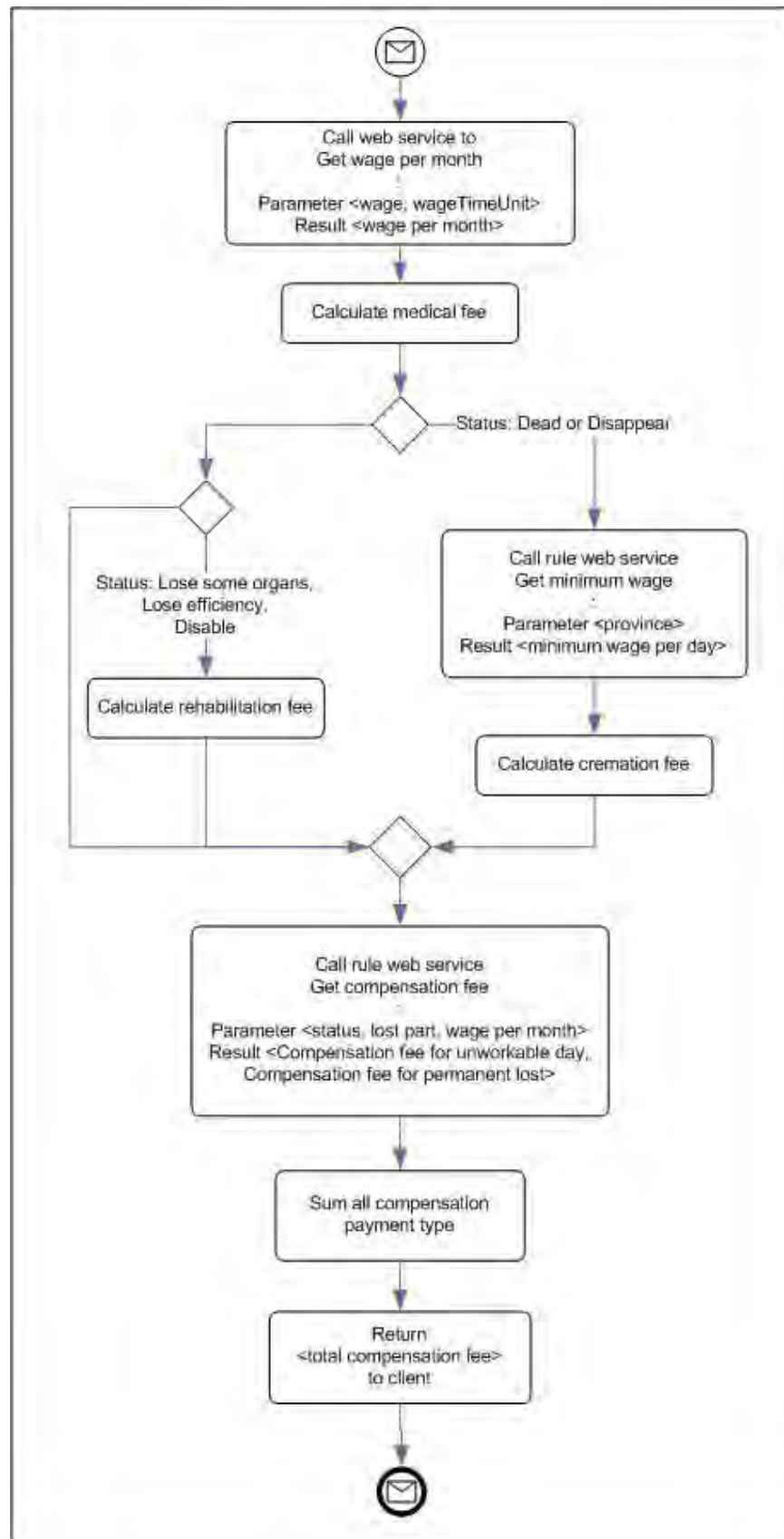
- กรณีสูญเสียอวัยวะบางส่วนของร่างกาย จ่ายตามประเภทของอวัยวะที่สูญเสีย ในอัตราร้อยละ 60 ของค่าจ้างรายเดือน ไม่เกิน 10 ปี เช่น สูญเสียแขนหนึ่งข้างจ่าย 10 ปี สูญเสียขาหนึ่งข้างจ่าย 6 ปี 8 เดือน เป็นต้น

- กรณีทุพพลภาพ จ่ายในอัตราร้อยละ 60 ของค่าจ้างรายเดือน 15 ปี

- กรณีตายหรือสูญหาย จ่ายในอัตราร้อยละ 60 ของค่าจ้างรายเดือน 8 ปี

4.2.2 กระบวนการธุรกิจ

ในกรณีที่ทดสอบนี้ได้สร้างกระบวนการธุรกิจที่คำนวณหาจำนวนเงินทดแทน ที่มากที่สุดที่เป็นไปได้ที่ต้องจ่ายให้กับลูกจ้าง ในกรณีที่ลูกจ้างประสบอุบัติเหตุในการทำงาน 1 ครั้ง โดยคิดจากเงินทดแทนทุกประเภทที่กล่าวในหัวข้อลักษณะเงินทดแทน ได้แก่ ค่ารักษาพยาบาล ค่าฟื้นฟู ค่าทำศพและค่าทดแทน กระบวนการธุรกิจเป็นดังรูปที่ 4.2 ในหน้าถัดไป กระบวนการธุรกิจที่สร้างขึ้นเขียนด้วยภาษาบีเพิลและทำงานในรูปแบบเว็บเซอวิสเช่นเดียวกับตารางการตัดสินใจ ประกอบด้วยขั้นตอน ดังนี้



รูปที่ 4.2 กระบวนการธุรกิจคำนวณหาจำนวนเงินทดแทน

- 1) กระบวนการธุรกิจรับพารามิเตอร์จากผู้เรียกซึ่งประกอบด้วย สภาพของลูกจ้าง (เสียชีวิต หรือทุพพลภาพ เป็นต้น) จังหวัดที่ทำงาน ค่าจ้าง หน่วยงานที่ได้รับค่าจ้าง (เช่น ได้ค่าจ้าง เป็นรายวัน หรือรายชั่วโมง เป็นต้น) อวัยวะที่สูญเสีย จำนวนวันที่ทำงานไม่ได้
- 2) เรียกใช้เว็บเซอร์วิสเพื่อหาค่าจ้างรายเดือน บางครั้งลูกจ้างอาจได้รับค่าจ้างเป็น รายวันหรือรายชั่วโมง แต่ในการคำนวณค่าทดแทนนั้นคิดบนฐานของค่าจ้างรายเดือน ดังนั้นจึง ต้องหาค่าจ้างรายเดือน โดยเรียกใช้เว็บเซอร์วิสเพื่อคำนวณหาค่านั้น
- 3) คำนวณหาค่ารักษาพยาบาลมากที่สุดที่อาจต้องจ่าย คือ 30,000 บาท
- 4) ตรวจสอบว่าลูกจ้างเสียชีวิตหรือสูญหายหรือไม่
- 5) ถ้าลูกจ้างเสียชีวิตหรือสูญหาย จะทำการหาค่าแรงขั้นต่ำ ซึ่งขึ้นอยู่กับจังหวัดที่ลูกจ้าง ทำงาน การคำนวณส่วนนี้มีลักษณะเป็นกฎธุรกิจ ซึ่งสามารถแยกออกมาเป็นเว็บเซอร์วิสกฎธุรกิจ หรือตารางการตัดสินใจได้ โดยจะกล่าวถึงหัวข้อตารางการตัดสินใจถัดไป
- 6) คำนวณค่าทำศพ คือ 100 เท่าของค่าแรงขั้นต่ำรายวัน
- 7) ถ้าลูกจ้างไม่เสียชีวิตหรือสูญหาย แต่สูญเสียอวัยวะบางส่วน สูญเสียสมรรถภาพใน การทำงานของอวัยวะหรือทุพพลภาพ คำนวณค่าฟื้นฟูสมรรถภาพ ซึ่งเท่ากับ 20,000 บาท
- 8) เมื่อคำนวณค่าทำศพหรือค่าฟื้นฟูสมรรถภาพแล้ว คำนวณค่าทดแทนรายเดือน และค่าทดแทนจากความสูญเสียถาวร ซึ่งขึ้นอยู่กับจำนวนวันที่ลูกจ้างไม่สามารถทำงานได้ และ สภาพของลูกจ้าง เช่น พิการ หรือสูญเสียอวัยวะบางส่วน เป็นต้น การคำนวณส่วนนี้มีลักษณะเป็น กฎธุรกิจ สามารถแยกออกมาเป็นเว็บเซอร์วิสกฎธุรกิจ หรือตารางการตัดสินใจได้
- 9) คำนวณหาผลรวมจำนวนเงินทดแทนจากค่ารักษาพยาบาล ค่าทำศพ ค่าฟื้นฟู สมรรถภาพและค่าทดแทนรายเดือน
- 10) คืนค่าผลรวมนั้นไปยังผู้เรียกใช้กระบวนการธุรกิจ

4.2.3 ตารางการตัดสินใจ

จากกระบวนการธุรกิจที่กล่าวข้างต้นสามารถแยกกฎธุรกิจเพื่อนำมาสร้างตารางการตัดสินใจที่เกี่ยวข้องกับการคำนวณเงินทดแทน 3 ตาราง ดังนี้

- 1) ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน
ตารางการตัดสินใจนี้เป็นตารางหาค่าจ้างขั้นต่ำรายวัน มีเงื่อนไขคือจังหวัดที่ลูกจ้างทำงาน อยู่ ดังตัวอย่างในรูปที่ 4.3

Condition	Action
The province is ?	Minimum wage per day is ? baht
1. [Rayong]	1. [173]
1. [Trat]	1. [156]
1. [Chon Buri]	1. [180]
1. [Chanthaburi]	1. [163]

รูปที่ 4.3 ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน

การกำหนดเงื่อนไขในไฟล์ข้อเท็จจริงเป็นดั่งโค้ดเอกซ์เอ็มแอลในรูปที่ 4.4 กำหนดภาษารวมชาติคือ “The province is ?” มีตัวแปรปรับแต่ง คือ “province” เป็นประเภทสายอักขระ

จากรูปที่ 4.4 จะเห็นว่ามีแท็ก enum ซึ่งระบุค่าที่เป็นไปได้ของตัวแปรปรับแต่ง province ซึ่งก็คือจังหวัดในประเทศไทย ผู้ใช้จะค่าเหล่านี้เป็นชื่อจังหวัด แต่ค่าจริงๆ ที่เก็บและใช้ประเมินเงื่อนไขในไฟล์เซตกฎธุรกิจเป็นรหัสจังหวัด

```

<conditionTemplate id="16" name="Province Condition"
  description="Province of thailand">
  <naturalLanguage>
    The province is [operator] ?
  </naturalLanguage>
  <customVariables>
    <customVariable id="1" name="province" type="string" >
      <enum value="10" text="Krung Thep Maha Nakhon" />
      <enum value="83" text="Phuket" />
      ...
    </customVariable>
  </customVariables>
</conditionTemplate>

```

รูปที่ 4.4 แผ่นแบบเงื่อนไขจังหวัดที่ถูกจ้างทำงาน

สำหรับการกระทำนั้นเป็นการคืนค่าจ้างที่กำหนดในตารางนั้นกลับไปยังผู้เรียกใช้ ดังรูปที่ 4.5 มีตัวแปรปรับแต่ง คือ “mininumWage” เป็นประเภทตัวเลข

ถ้าเครื่องลูกข่ายต้องการเรียกใช้ตารางการตัดสินใจนี้ ต้องส่งพารามิเตอร์ คือ จังหวัด และจะได้ผลลัพธ์เป็นค่าแรงขั้นต่ำของจังหวัดนั้น

```

<actionTemplate id="8" name="Allow Miminum Wage" description="Allow the
miminum wage per day">
  <naturalLanguage>Miminum wage per day is ? baht</naturalLanguage>
  <customVariables>
    <customVariable id="1" name="miminumWage" type="double" />
  </customVariables>
  <activities>
    <return>
      <parameter customVariableId="1"/>
    </return>
  </activities>
</actionTemplate>

```

รูปที่ 4.5 แผ่นแบบการกระทำการกำหนดค่าแรงขั้นต่ำ

2) ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้ ตารางการตัดสินใจนี้คำนวณหา เงินทดแทนกรณีลูกจ้างไม่สามารถทำงานได้ ซึ่งคิดเป็นวัน และจ่ายค่าชดเชยต่อเมื่อลูกจ้างทำงานไม่ได้เกิน 3 วัน ดังรูปที่ 4.6

Condition	Action	
The employee can't work ? days.	The employee has wage ? bath per month and can't work ? days.	Give compensation fee ? baht
1. [more than 3]	1. [Parameter.wage-per-month, Unworkable Days Condition.unworkableDays]	-
-	-	1. [0]

รูปที่ 4.6 ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้

เงื่อนไขในตาราง คือ จำนวนวันที่ทำงานไม่ได้ ซึ่งเงื่อนไขนี้มีตัวแปรปรับแต่งหนึ่งตัว คือ unworkableDays เป็นข้อมูลประเภทจำนวนเต็ม

จากตาราง ในกฎแรกถ้าลูกจ้างทำงานไม่ได้เกิน 3 วัน จะเรียกการกระทำ “Calculating unworkable-days compenstation fee” ซึ่งเป็นการกระทำแรกในตาราง มีภาษาธรรมชาติ คือ “The employee has wage ? bath per month and can't work ? days.” การกระทำนี้มีตัวแปรปรับแต่ง 2 ตัว คือ ค่าจ้างรายเดือน และจำนวนวันที่ทำงานไม่ได้ ตัวแปรแรกผู้ใช้กำหนดเป็นค่าจากพารามิเตอร์ของตารางการตัดสินใจ คือ wage-per-month ขณะที่ตัวแปรที่สอง เป็นค่าจากตัวแปร unworkableDays ซึ่งมาจากเงื่อนไขนั่นเอง

การกระทำนี้ซึ่งกำหนดในไฟล์ข้อเท็จจริง จะทำการเรียกใช้เว็บเซอวิสคำนวณค่าทดแทน สำหรับการทำงานไม่ได้โดยส่งพารามิเตอร์คือ ค่าจ้างรายเดือน และจำนวนวันที่ทำงานไม่ได้ ซึ่ง กำหนดรับจากมาตัวแปรปรับแต่งที่กล่าวไปแล้ว เว็บเซอวิสจะคืนค่าทดแทนกลับมา และการกระทำจะนำค่าคืนกลับไปยังตารางการตัดสินใจ ตามเอกซ์เอ็มแอลในรูปที่ 4.8

สุดท้ายตารางการตัดสินใจจะกำหนดค่านี้ ให้กับผลลัพธ์ของตารางเพื่อคืนค่ากลับไปยังผู้เรียกใช้ตารางการตัดสินใจ

จากตารางการตัดสินใจข้างต้นจะเห็นอีกกฎ ที่กำหนดเงื่อนไขจำนวนวันที่ทำงานไม่ได้ ไม่มากกว่า 3 วัน ในกรณีนี้จะไม่จ่ายค่าทดแทนให้โดยเรียกใช้การกระทำ ที่คืนค่าทดแทนกลับไป จากตารางเป็นคืนค่า 0 ซึ่งกำหนดโดยผู้สร้างตารางการตัดสินใจ

ถ้าเครื่องลูกข่ายต้องการเรียกใช้ตารางการตัดสินใจนี้ ต้องส่งพารามิเตอร์ คือ ค่าจ้างราย เดือน และจำนวนวันที่ทำงานไม่ได้ และได้ผลลัพธ์เป็นค่าทดแทนทั้งหมด

3) ตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด

ตารางการตัดสินใจนี้คำนวณค่าทดแทน ที่ต้องจ่ายเป็นรายเดือนให้แก่ลูกจ้าง กรณีที่ ลูกจ้างไม่สามารถทำงานได้ เสียชีวิต สูญหาย ทุพพลภาพ หรือสูญเสียอวัยวะบางส่วน ดังบางส่วน ของตารางในรูปที่ 4.7

Condition		Action	
The employee status is ?	The lost organ is ?	The employee has wage ? bath per month and get compensation ? years and ? months.	Call sub table
1. [dead]	-	1. [Parameter.wage_per_month, 8, 0]	2. [Calculating compensation for unworkable day]
2. [disappear]	-	1. [Parameter.wage_per_month, 8, 0]	2. [Calculating compensation for unworkable day]
1. [disabled]	-	1. [Parameter.wage_per_month, 15, 0]	2. [Calculating compensation for unworkable day]
1. [lost some organs]	2. [one arm]	1. [Parameter.wage_per_month, 10, 0]	2. [Calculating compensation for unworkable day]
1. [lost some organs]	2. [one leg]	1. [Parameter.wage_per_month, 6, 8]	2. [Calculating compensation for unworkable day]
1. [lost some organs]	2. [one hand]	1. [Parameter.wage_per_month, 9, 0]	2. [Calculating compensation for unworkable day]

รูปที่ 4.7 ตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด


```

<actionTemplate id="14" name="Calculating unworkable-days compenstation fee"
  description="Calculating compenstation fee for unworkable days.">
  <naturalLanguage>The employee has wage ? bath per month and can't work ?
days.</naturalLanguage>
  <customVariables>
    <customVariable id="1" name="wagePerMonth" type="double"/>
    <customVariable id="2" name="unworkableDays" type="int"/>
  </customVariables>
  <activities>
    <variable id="3" name="compensationFee" type="double"/>
    <invoke service="CalWebServiceService"
operation="calCompensationFeeForUnworkable"
  wsdl="CalWebServiceService.wsdl"
partnerLinkType="CalWebServiceLinkType">
      <parameter customVariableId="1" toElement="parameters/wagePerMonth"/>
      <parameter customVariableId="2" toElement="parameters/unworkableDay"/>
      <result element="parameters/return" variableId = "3"/>
    </invoke>
    <return>
      <parameter variableId="3"/>
    </return>
  </activities>
</ actionTemplate >

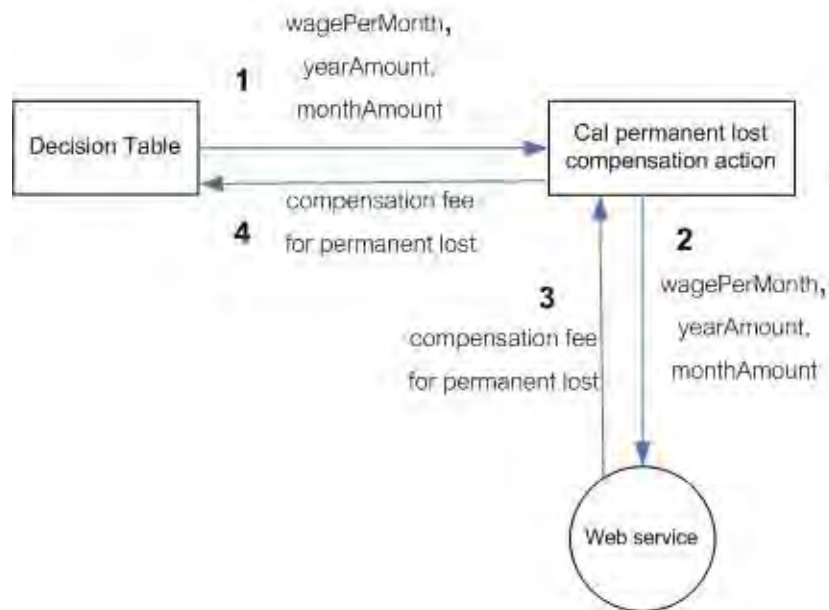
```

รูปที่ 4.8 แผ่นแบบการกระทำการคำนวณค่าทดแทนจากจำนวนวันที่ไม่สามารถทำงานได้

จากตารางการตัดสินใจในรูปมี 2 เงื่อนไข คือ เงื่อนไขสภาพของลูกจ้างที่เป็นเหตุให้ต้องจ่ายค่าชดเชยเช่น เสียชีวิต ทูพพลภาพ เป็นต้น และเงื่อนไขวัยวุฒิที่สูญเสีย

สำหรับการกระทำนั้น มี 2 การกระทำ คือ

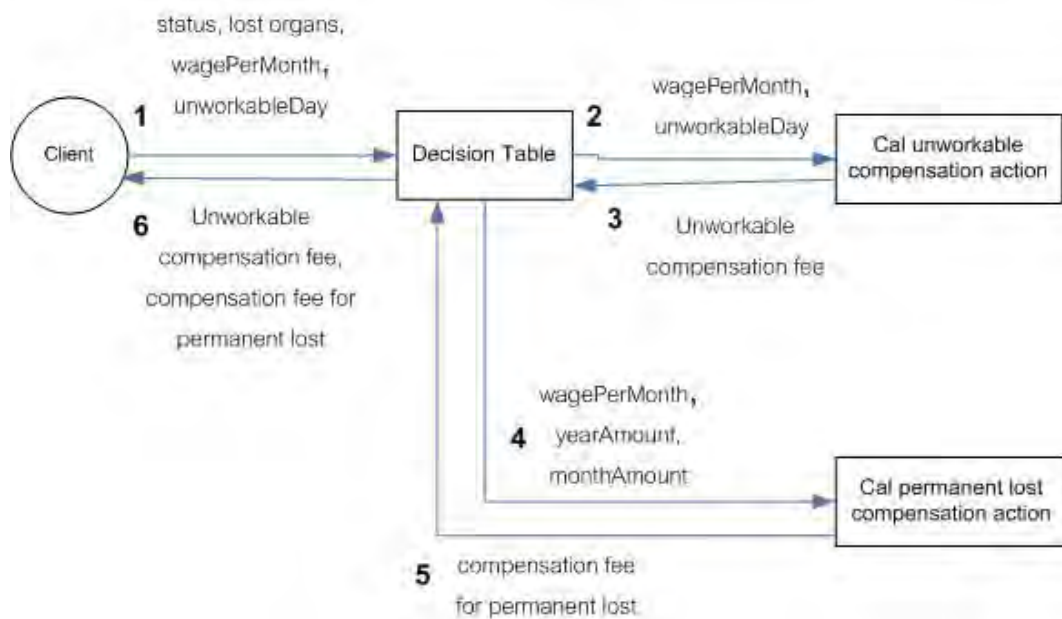
- การกระทำคำนวณค่าทดแทนจากความสูญเสียถาวร ได้แก่ เสียชีวิต สูญหาย ทูพพลภาพ และสูญเสียอวัยวะบางส่วน ตัวแปรปรับแต่ง คือ ค่าจ้างรายเดือน จำนวนปีและเดือนที่ต้องจ่ายให้ลูกจ้าง การกระทำเรียกใช้เว็บเซอร์วิสเพื่อคำนวณค่าทดแทนโดย ส่งค่าของตัวแปรปรับแต่งเป็นพารามิเตอร์ จากนั้นเว็บเซอร์วิสคืนค่าทดแทนกลับมายังการกระทำ และตารางการตัดสินใจ ดังรูปที่ 4.9



รูปที่ 4.9 การส่งพารามิเตอร์และรับข้อมูลระหว่างตารางการตัดสินใจ การกระทำและเว็บเซอร์วิสของการกระทำคำนวณค่าทดแทนจากความสูญเสียถาวร

- การกระทำที่เรียกตารางย่อยเพื่อคำนวณค่าทดแทนจากจำนวนวันที่ไม่สามารถทำงานได้ ซึ่งก็คือ ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้ ที่ได้กล่าวในหัวข้อที่แล้วนั่นเอง

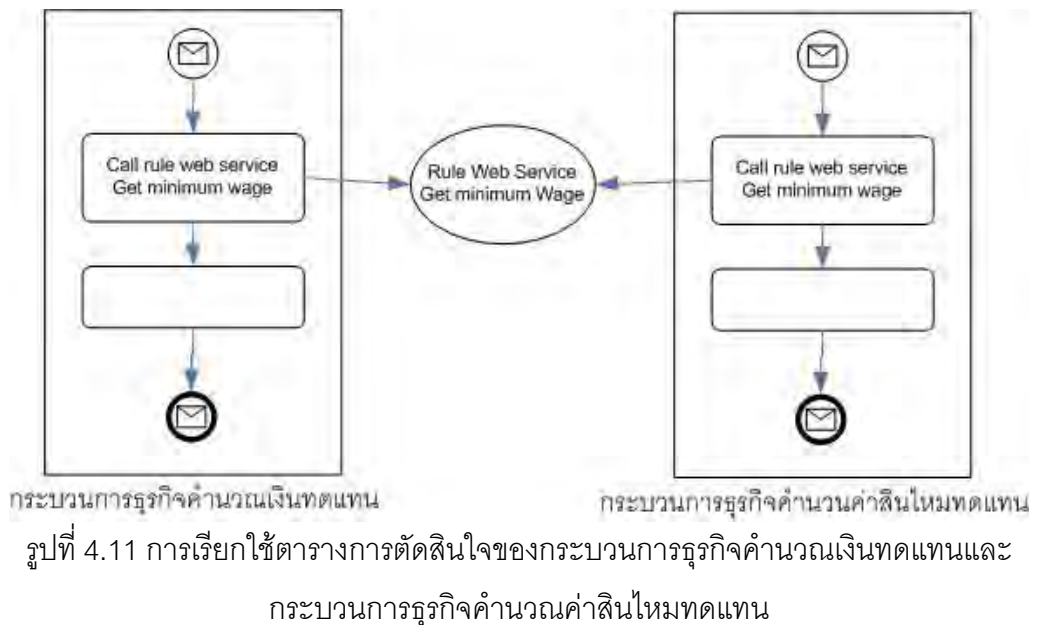
ดังนั้นตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมดมีพารามิเตอร์ คือ สภาพของลูกจ้าง อวัยวะที่สูญเสีย ค่าจ้างรายเดือน จำนวนวันที่ไม่สามารถทำงานได้ ส่วนผลลัพธ์ของตารางคือ ค่าทดแทนจากจำนวนวันที่ไม่สามารถทำงานได้ และค่าทดแทนจากความสูญเสียถาวร ดังรูปที่ 4.10



รูปที่ 4.10 การส่งพารามิเตอร์และรับข้อมูลระหว่างผู้เรียกใช้ตารางการตัดสินใจ ตารางการตัดสินใจ และการกระทำของการกระทำคำนวณค่าทดแทนทั้งหมด

4.2.4 การนำตารางการตัดสินใจมาใช้ซ้ำ

หัวข้อที่กล่าวถึงตารางการตัดสินใจที่ถูกเรียกใช้โดย กระบวนการธุรกิจคำนวณหาเงินทดแทนที่จ่ายให้ลูกจ้างกรณีประสบอุบัติเหตุ ตารางการตัดสินใจที่สามารถให้บริการกระบวนการธุรกิจอื่นๆ ได้ เช่น ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน สามารถนำไปใช้ในการคำนวณค่าสินไหมทดแทนกรณีเจ้าหน้าที่ของรัฐกระทำผิดละเมิดต่อบุคคลภายนอก โดยมีหลักเกณฑ์ข้อหนึ่งระบุว่า “ถ้าผู้เสียหายที่มีรายได้ไม่แน่นอนหรือไม่มีรายได้ประจำให้ชดใช้ความเสียหายได้ตามช่วงเวลาที่ยุ่เสียหายไม่สามารถไปทำงานได้ ให้คำนวณจากอัตราค่าจ้างขั้นต่ำรายวันตามกฎหมายว่าด้วยคุ้มครองแรงงาน [25]” ด้วยหลักเกณฑ์ที่ว่านี้ กระบวนการธุรกิจคำนวณค่าสินไหมทดแทน สามารถเรียกใช้ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน เพื่อนำข้อมูลมาคำนวณได้ ดังรูป 4.11 กระบวนการธุรกิจคำนวณเงินทดแทนและกระบวนการธุรกิจคำนวณค่าสินไหมทดแทน เรียกใช้ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน



4.2.5 การทดสอบความถูกต้อง

การทดสอบความถูกต้องเป็นการทดสอบว่าเว็บเซอร์วิสจากตารางการตัดสินใจ และกระบวนการธุรกิจทำงานได้ถูกต้องหรือไม่ การทดสอบใช้โปรแกรม NetBeans IDE 6.1 การทดสอบทำโดยสร้างข้อความโซพ (SOAP message) เพื่อเรียกใช้และส่งพารามิเตอร์ให้เว็บเซอร์วิส จากนั้นตรวจสอบผลลัพธ์ที่ได้จากเว็บเซอร์วิส ว่าถูกต้องตามที่กำหนดในตารางการตัดสินใจและกระบวนการธุรกิจหรือไม่

1) ตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน

ตารางการตัดสินใจนี้มีพารามิเตอร์ คือ รหัสจังหวัดที่ต้องการทราบค่าจ้างขั้นต่ำ ดังตัวอย่างข้อความโซพดังรูปที่ 4.12

```
<Envelope>
  <Body>
    <Mininum_wage_policy_30ParamType>
      <Province_Condition_province>21</Province_Condition_province>
    </Mininum_wage_policy_30ParamType>
  </Body>
</Envelope>
```

รูปที่ 4.12 ตัวอย่างข้อความโซพของพารามิเตอร์ของตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน

จากรูปที่ 4.12 ตัวอย่างส่งรหัสจังหวัด 21 (จังหวัดระนอง) เป็นพารามิเตอร์ ผลลัพธ์ที่ได้คือ 173 ดังข้อความไซพในรูปที่ 4.13 ซึ่งเป็นตามที่กำหนดในตารางการตัดสินใจ

```
<Envelope>
  <Body>
    <Mininum_wage_policy_30ResultType>
      <Mininum_wage_per_day>173.0</Mininum_wage_per_day>
    </Mininum_wage_policy_30ResultType>
  </Body>
</Envelope>
```

รูปที่ 4.13 ตัวอย่างข้อความไซพของผลลัพธ์ของตารางการตัดสินใจหาค่าจ้างขั้นต่ำรายวัน

2) ตารางการตัดสินใจคำนวณค่าทดแทนกรณีถูกจ้างไม่สามารถทำงานได้ ตารางการตัดสินใจนี้มีพารามิเตอร์ได้แก่ จำนวนวันที่ไม่สามารถทำงานได้และค่าจ้างต่อเดือน ดังข้อความไซพในรูปที่ 4.14

```
<Envelope>
  <Body>
    <Unworkable-day_compensation_fee_calculation_32ParamType>
      <Unworkable_days_Condition_unworkableDays>10
    </Unworkable_days_Condition_unworkableDays>
      <wage-per-month>20000</wage-per-month>
    </Unworkable-day_compensation_fee_calculation_32ParamType>
  </Body>
</Envelope>
```

รูปที่ 4.14 ตัวอย่างข้อความไซพของพารามิเตอร์ของตารางการตัดสินใจคำนวณค่าทดแทนกรณีถูกจ้างไม่สามารถทำงานได้

จากตัวอย่างในรูป 4.14 จำนวนวันคือ 10 และค่าจ้างต่อเดือนคือ 20000 ผลลัพธ์คือค่าทดแทนจากวันที่ไม่สามารถทำงานได้ การหาค่าทดแทนมีสูตร คือ (ค่าจ้างรายเดือน / 26)*จำนวน

วัน*0.6 ดังนั้นจากพารามิเตอร์ที่ส่งไปต้องจ่ายค่าทดแทนเท่ากับ 4615.38 ตามผลลัพธ์ในรูปแบบที่ 4.15

```
<Envelope>
  <Body>
    <Unworkable-day_compensation_fee_calculation_32ResultType>
      <compensation>4615.38</compensation>
    </Unworkable-day_compensation_fee_calculation_32ResultType>
  </Body>
</Envelope>
```

รูปที่ 4.15 ตัวอย่างข้อความไซฟของผลลัพธ์ของตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกจ้างไม่สามารถทำงานได้

3) ตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด

ตารางการตัดสินใจนี้มีพารามิเตอร์ได้แก่ สภาพของลูกจ้าง อวัยวะที่สูญเสีย จำนวนวันที่ไม่สามารถทำงานได้และค่าจ้างต่อเดือน ดังข้อความไซฟในรูปแบบที่ 4.16

```
<Envelope>
  <Body>
    <Compensation_fee_calculation_rules_31ParamType>
      <Status_of_Employee_Condition_status>Lost some organs
    </Status_of_Employee_Condition_status>
    <Lost_some_organs_lostOrgans>One arm
    </Lost_some_organs_lostOrgans>
    <wage_per_month>20000</wage_per_month>
    <Unworkable_day_amount>10</Unworkable_day_amount>
    </Compensation_fee_calculation_rules_31ParamType>
  </Body>
</Envelope>
```

รูปที่ 4.16 ตัวอย่างข้อความไซฟของพารามิเตอร์ของตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด

จากตัวอย่างสภาพของลูกจ้างคือ สูญเสียอวัยวะ อวัยวะที่สูญเสียคือ แขนหนึ่งข้าง จำนวนวันคือ 10 และค่าจ้างรายเดือนคือ 20000

ผลลัพธ์คือ ค่าทดแทนจากการสูญเสียอวัยวะและค่าทดแทนที่ทำงานไม่ได้ ค่าทดแทนจากแทนสูญเสียแขนหนึ่งข้างคือ การจ่ายเงินทดแทนรายเดือนเป็นเวลา 10 ปี โดยแต่ละเดือนจะจ่ายเป็นเงิน 60% ของค่าจ้างรายเดือน นั่นคือต้องจ่ายเงินทดแทนในส่วนนี้รวมเท่ากับ $10 \times 12(\text{เดือน}) \times 20000 \times 0.6 = 1440000.0$ ได้ผลลัพธ์ดังข้อความไซไฟในรูปที่ 4.17 ค่าทดแทนที่ทำงานไม่ได้

```
<Envelope>
  <Body>
    <Compensation_fee_calculation_rules_31ResultType>
      <permanent-lost_compensation_fee>1440000.0
    </permanent-lost_compensation_fee>
    <unworkable_compensation_fee>4615.38
    </unworkable_compensation_fee>
    </Compensation_fee_calculation_rules_31ResultType>
  </Body>
</Envelope>
```

รูปที่ 4.17 ตัวอย่างข้อความไซไฟของผลลัพธ์ของตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด

4) กระบวนการธุรกิจ

กระบวนการธุรกิจนี้คำนวณหาจำนวนเงินทดแทน ที่มากที่สุดที่เป็นไปได้ที่ต้องจ่ายให้กับลูกจ้าง พาราเมเตอร์ได้แก่ สภาพของลูกจ้าง รหัสจังหวัดที่ลูกจ้างทำงาน ค่าจ้าง หน่วยเวลาที่ได้รับค่าจ้าง อวัยวะที่สูญเสีย จำนวนวันที่ไม่สามารถทำงานได้ ดังรูปที่ 4.18

จากตัวอย่างสภาพของลูกจ้างคือ สูญเสียอวัยวะ รหัสจังหวัดคือ 21 ค่าจ้างคือ 20000 หน่วยเวลาที่ได้รับค่าจ้างคือ เดือน (หมายถึงได้ค่าจ้าง 20000 บาทต่อเดือน) อวัยวะที่สูญเสียคือ แขนหนึ่งข้าง จำนวนวันคือ 10

ผลลัพธ์คือ ผลรวมของค่าทดแทนจากการสูญเสียอวัยวะ ค่าทดแทนจากการทำงานไม่ได้ ค่ารักษาพยาบาลและค่าฟื้นฟูสมรรถภาพในการทำงาน ซึ่งเท่ากับ $1440000 + 4615.38 + 30000 + 20000 = 1494615.38$ ดังผลลัพธ์ข้อความไซไฟในรูปที่ 4.19

```

<Envelope>
  <Body>
    <CalculatingPossibleMaxCompensationOperation>
      <employeeStatus>Lost some organs</employeeStatus>
      <province>21</province>
      <wage>20000</wage>
      <wageTimeUnit>month</wageTimeUnit>
      <lostOrgan>One arm</lostOrgan>
      <unworkable_days>10</unworkable_days>
    </CalculatingPossibleMaxCompensationOperation>
  </Body>
</Envelope>

```

รูปที่ 4.18 ตัวอย่างข้อความไซพของพารามิเตอร์ของกระบวนการธุรกิจคำนวณหาจำนวนเงิน

ทดแทน

```

<Envelope>
  <Body>
    <CalculatingPossibleMaxCompensationOperationResponse>
      <compensationFee>1494615.38</compensationFee>
    </CalculatingPossibleMaxCompensationOperationResponse>
  </Body>
</Envelope>

```

รูปที่ 4.19 ตัวอย่างข้อความไซพของผลลัพธ์ของกระบวนการธุรกิจคำนวณหาจำนวนเงินทดแทน

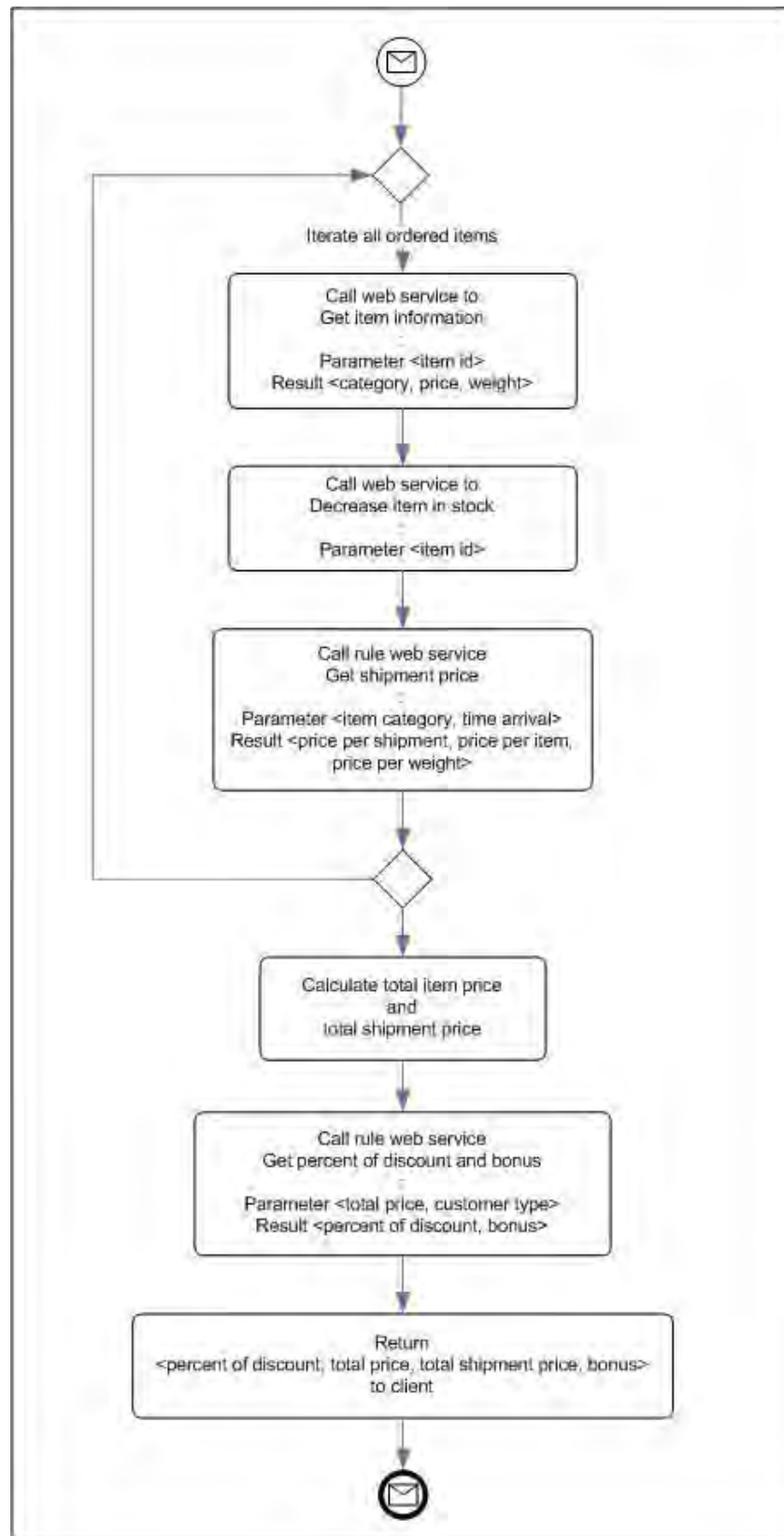
4.3 กรณีทดสอบที่ 2 : การสั่งซื้อสินค้าผ่านเว็บไซต์

กรณีทดสอบที่ 2 เกี่ยวข้องกับกระบวนการธุรกิจ และกฎธุรกิจของการสั่งซื้อสินค้าผ่านเว็บไซต์ มีกระบวนการธุรกิจสำหรับการสั่งซื้อสินค้า และมีตารางการตัดสินใจที่เกี่ยวข้องหลัก 2 ตารางประกอบด้วย ตารางการตัดสินใจกำหนดค่าส่งสินค้า และตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม

4.3.1 กระบวนการธุรกิจ

กระบวนการธุรกิจในกรณีทดสอบนี้เกี่ยวข้องกับการสั่งซื้อสินค้าผ่านเว็บไซต์ เพื่อหาราคารวม ค่าขนส่ง เปอร์เซ็นต์ส่วนลดและของแถม ดังรูปที่ 4.20 กระบวนการธุรกิจที่สร้างขึ้นเขียนด้วยภาษาพีเพิลและทำงานในรูปแบบเว็บเซอวิส ประกอบด้วยขั้นตอนดังนี้

- 1) กระบวนการธุรกิจรับข้อมูลสินค้าที่ซื้อ ประเภทลูกค้า จำนวนวันส่งสินค้าและวิธีการชำระเงิน
- 2) ทำการวนซ้ำเพื่อประมวลผลสินค้าแต่ละชิ้นที่ซื้อ
- 3) เรียกใช้เว็บเซอวิสเพื่อขอรายละเอียดสินค้า โดยส่งรหัสสินค้าเป็นพารามิเตอร์ ผลลัพธ์ คือ ชื่อสินค้า ประเภท ราคาและน้ำหนัก
- 4) เรียกใช้เว็บเซอวิสตัดยอดสินค้าในคลังสินค้า
- 5) เรียกใช้เว็บเซอวิสกฎธุรกิจเพื่อหาค่าส่งสินค้าทุกประเภท ค่าส่งสินค้าขึ้นอยู่กับประเภทสินค้า การกำหนดค่าส่งสินค้ามีลักษณะการทำงานเป็นกฎธุรกิจ จึงได้แยกการทำงานส่วนนี้เป็นเว็บเซอวิสกฎธุรกิจ หรือตารางการตัดสินใจ ซึ่งจะกล่าวในหัวข้อถัดไป
- 6) กำหนดเปอร์เซ็นต์ส่วนลดและของแถม โดยขึ้นอยู่กับราคารวมและประเภทของลูกค้า การทำงานส่วนนี้มีลักษณะการทำงานเป็นกฎธุรกิจ จึงได้แยกการทำงานส่วนนี้เป็นเว็บเซอวิสกฎธุรกิจ หรือตารางการตัดสินใจ
- 7) เมื่อประมวลผลสินค้าครบทุกชิ้นแล้ว คำนวณราคารวมและค่าส่งรวม
- 8) คืนผลลัพธ์ คือ เปอร์เซ็นต์ส่วนลด ของแถม ค่าส่งรวมและราคารวม กลับไปยังผู้เรียกใช้กระบวนการธุรกิจนี้ ซึ่งเป็นเว็บแอปพลิเคชันที่ติดต่อกับผู้ซื้อสินค้า



รูปที่ 4.20 กระบวนการธุรกิจการสั่งซื้อสินค้าผ่านเว็บไซต์

4.3.2 ตารางการตัดสินใจ

จากกระบวนการธุรกิจข้างต้นสามารถแยกส่วนที่เป็นกฎธุรกิจออกมาเป็นเว็บเซอริวิสกฏธุรกิจ หรือตารางการตัดสินใจได้ดังนี้

1) ตารางการตัดสินใจกำหนดค่าส่งสินค้า เป็นตารางการตัดสินใจที่กำหนดค่าส่งสินค้า โดยค่าส่งสินค้ามี 3 ประเภท คือ

- ค่าส่งต่อครั้ง
- ค่าส่งสินค้าต่อชิ้น
- ค่าส่งสินค้าต่อน้ำหนักหนึ่งกิโลกรัม

สินค้าทุกประเภทต้องคิดค่าส่งต่อครั้งและคิดเพิ่มด้วยค่าส่งสินค้าต่อชิ้นหรือค่าส่งสินค้าต่อน้ำหนักขึ้นอยู่กับประเภทสินค้า ค่าส่งขึ้นอยู่กับประเภทสินค้าและจำนวนวันส่ง ถ้าส่งสินค้าหลายชิ้นให้คิดค่าส่งต่อครั้งจากสินค้าที่มีค่าส่งสูงที่สุด เช่น ลูกค้าซื้อของเล่นกับหนังสือ ซึ่งค่าส่งต่อครั้งเป็น 50 และ 100 บาทตามลำดับ แต่เนื่องจากสินค้าส่งพร้อมกันหรือครั้งเดียวกัน จึงคิดค่าส่งต่อครั้งเป็น 100 บาท เป็นต้น ส่วนค่าส่งประเภทที่เหลือต้องคิดรวมทุกชิ้น ตารางการตัดสินใจกำหนดค่าส่งสินค้า ประกอบด้วย 4 ตาราง ดังนี้

- ตารางการตัดสินใจกำหนดค่าส่งหลัก เป็นตารางการตัดสินใจหลักที่รับพารามิเตอร์ทั้งหมดจากเครื่องลูกข่ายที่เรียกใช้บริการหาค่าขนส่ง ตารางนี้ประกอบกฎธุรกิจ 3 กฎ ดังรูปที่ 4.21 มีเงื่อนไขคือจำนวนวันส่ง คือ ส่งสินค้าภายใน 1 วัน ส่งสินค้าภายใน 2 วันและส่งสินค้าภายใน 3 - 5 วัน การกระทำคือการเรียกตารางย่อยเพื่อหาค่าส่งตามประเภทของวันส่ง ซึ่งกล่าวหัวข้อถัดไป

Condition	Action
Delivery within ? business days.	Call sub table
1. [\leq between \leq 3 and 5]	1. [Get shipment price for 3-5 days]
1. [equal to 2]	1. [Get shipment price for 2 days]
1. [equal to 1]	1. [Get shipment price for 1 day]

รูปที่ 4.21 ตารางการตัดสินใจกำหนดค่าส่งหลัก

ตารางการตัดสินใจกำหนดค่าส่งหลักมีพารามิเตอร์ คือ จำนวนวันส่งและประเภทสินค้า ผลลัพธ์ของตาราง คือ ค่าส่งต่อครั้ง ค่าส่งสินค้าต่อชิ้น ค่าส่งสินค้าต่อน้ำหนักหนึ่งกิโลกรัม

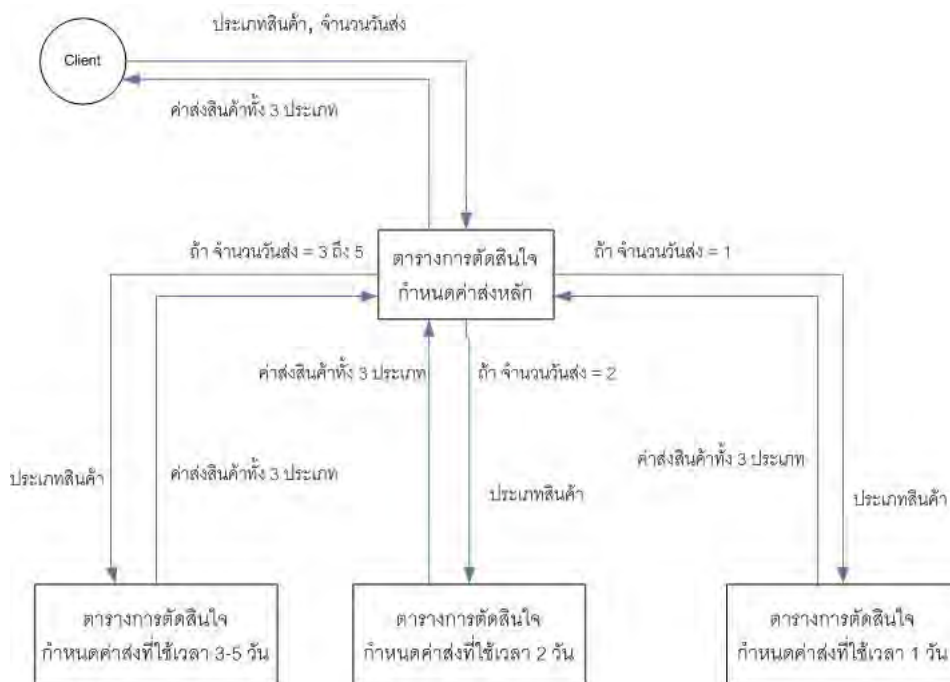
- ตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 1 วัน เป็นตารางการตัดสินใจที่กำหนดค่าส่งสินค้าทั้ง 3 ประเภทที่ได้กล่าวข้างต้นสำหรับส่งภายใน 1 วัน ค่าส่งสินค้าต่างกันไปตามประเภทของสินค้า ดังรูปที่ 4.22 การกระทำในตารางเป็นการคืนค่าส่งที่ผู้สร้างตารางกำหนดไว้ทั้ง 3 ประเภทกลับไปยังผู้เรียกใช้ตารางการตัดสินใจ และมีพารามิเตอร์ คือ ประเภทสินค้า

Condition	Action			
	The product is in category ?	Price per shipment is ? baht	Price per item is ? baht	Price per one kilogram is ? baht
1. [Jewelry]	1. [390]	2. [150]	-	-
2. [Watches]	1. [390]	2. [150]	-	-
3. [Books]	1. [390]	2. [150]	-	-
1. [CDs]	1. [210]	2. [150]	-	-
2. [DVDs]	1. [210]	2. [150]	-	-
1. [Video Game Consoles]	1. [435]	2. [150]	-	-
1. [Video Games]	1. [330]	2. [150]	-	-
1. [Toys]	1. [510]	-	2. [60]	-
-	1. [480]	-	2. [60]	-

รูปที่ 4.22 ตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 1 วัน

- ตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 2 วัน เป็นตารางการตัดสินใจที่กำหนดค่าส่งสินค้าทั้ง 3 ประเภทที่ได้กล่าวข้างต้นสำหรับส่งภายใน 2 วัน มีโครงสร้างเหมือนตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 1 วัน
- ตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 3-5 วัน เป็นตารางการตัดสินใจที่กำหนดค่าส่งสินค้าทั้ง 3 ประเภทที่ได้กล่าวข้างต้นสำหรับส่งภายใน 3-5 วัน มีโครงสร้างเหมือนตารางการตัดสินใจกำหนดค่าส่งที่ใช้เวลา 1 วัน

ความสัมพันธ์ของทั้งตารางการตัดสินใจกำหนดค่าส่งสินค้าทั้ง 4 ตารางเป็นดังรูปที่ 4.23



รูปที่ 4.23 ความสัมพันธ์ของทั้งตารางการตัดสินใจกำหนดค่าส่งสินค้าทั้ง 4 ตาราง

2) ตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม เป็นตารางการตัดสินใจที่กำหนดเปอร์เซ็นต์ส่วนลดและของแถมที่ลูกค้าได้รับในการซื้อครั้งหนึ่งๆ ดังรูปที่ 4.24

Condition			Action	
Pay with ?	Customer type is ?	The total is ?	Allow ?% discount	Bonus is ?
1. [Credit Card]	2. [member]	3. [more than or equal to 5000]	1. [10]	2. [voucher]
1. [Wire Transfer]	2. [member]	3. [more than or equal to 5000]	1. [5]	2. [voucher]
1. [Credit Card]	2. [member]	-	1. [10]	2. [alarm clock]
1. [Wire Transfer]	2. [member]	-	1. [5]	2. [alarm clock]
1. [Credit Card]	-	2. [more than or equal to 10000]	1. [5]	2. [alarm clock]
1. [Wire Transfer]	-	2. [more than or equal to 10000]	1. [0.0]	2. [alarm clock]
1. [Credit Card]	-	2. [more than or equal to 5000]	1. [5]	2. [pen]
1. [Wire Transfer]	-	2. [more than or equal to 5000]	1. [0.0]	2. [pen]
1. [Credit Card]	-	2. [more than or equal to 1000]	1. [5]	2. [key holder]
1. [Wire Transfer]	-	2. [more than or equal to 1000]	1. [0.0]	2. [key holder]
-	-	-	1. [0.0]	2. [No bonus]

รูปที่ 4.24 ตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม

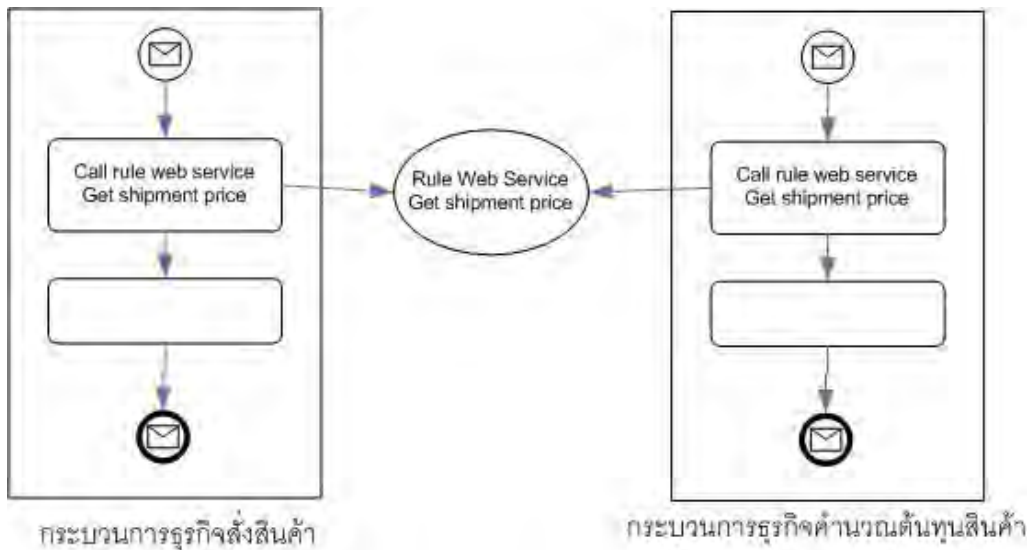
เงื่อนไข คือ วิธีการชำระเงิน (บัตรเครดิตหรือโอนเงิน) ประเภทลูกค้า (สมาชิกหรือไม่ได้เป็นสมาชิก) และราคารวม จากตารางจะเห็นว่ากฎธุรกิจ 4 กฎแรกระบุเงื่อนไขประเภทลูกค้าว่าเป็นสมาชิก แต่กฎที่เหลือระบุเป็นไม่สนใจ (เป็นหรือไม่เป็นลูกค้าก็ได้) อย่างไรก็ตามเมื่อตารางการตัดสินใจประมวลผลจะประเมินกฎเป็นลำดับจากบนลงล่าง ดังนั้นเงื่อนไขของ 4 กฎแรกจึงครอบคลุมทุกกรณีที่ลูกค้าเป็นสมาชิกแล้ว กฎธุรกิจที่เหลือจึงเป็นกฎสำหรับลูกค้าที่ไม่เป็นสมาชิกเท่านั้น โดยที่ผู้สร้างตารางการตัดสินใจไม่จำเป็นต้องระบุประเภทลูกค้าชัดเจนทุกกฎ

การกระทำเป็นการคืนค่าเปอร์เซ็นต์ส่วนลดและของแถมกลับไปยังผู้เรียกใช้ตารางการตัดสินใจนี้

4.3.3 การนำตารางการตัดสินใจมาใช้อย่าง

หัวข้อที่แล้วกล่าวถึงตารางการตัดสินใจที่ถูกเรียกใช้โดยกระบวนการธุรกิจสั่งสินค้า ตารางการตัดสินใจที่ว่าสามารถให้บริการกระบวนการธุรกิจอื่นๆ ได้ เช่น ตารางการกำหนดค่าส่งสินค้า ตารางนี้สามารถให้บริการกระบวนการธุรกิจคำนวณต้นทุนสินค้า เช่น ในกรณีที่ลูกค้าสั่งซื้อสินค้าเป็นจำนวนมากผ่านเว็บไซต์และต้องการนำสินค้านั้นไปขายต่อ ลูกค้าสามารถสร้างกระบวนการธุรกิจเพื่อคำนวณต้นทุนสินค้า โดยคิดต้นทุนจากราคาสินค้าและค่าขนส่ง กระบวนการธุรกิจนี้สามารถขอข้อมูลค่าขนส่งจากตารางการตัดสินใจ โดยที่ลูกค้าไม่จำเป็นต้องสั่งสินค้าก่อน ดังรูป

4.25 กระบวนการธุรกิจสั่งสินค้าและกระบวนการธุรกิจคำนวณต้นทุนสินค้าเรียกใช้ตารางการตัดสินใจกำหนดค่าส่งสินค้า



รูปที่ 4.25 การใช้ตารางการตัดสินใจร่วมกันของกระบวนการธุรกิจส่งสินค้า และกระบวนการธุรกิจคำนวณต้นทุนสินค้า

4.3.4 การทดสอบความถูกต้อง

1) ตารางการตัดสินใจกำหนดค่าส่งหลัก

เนื่องจากตารางการตัดสินใจกำหนดค่าส่งหลักมีการกระทำเป็นการเรียกใช้ตารางย่อย ซึ่งผลลัพธ์จากตารางย่อยเหมือนกับผลลัพธ์ของตารางหลัก ดังนั้นการทดสอบตารางการตัดสินใจกำหนดค่าส่งหลักเพียงตารางเดียวก็สามารถตรวจสอบผลลัพธ์การทำงานของตารางย่อย หรือตารางคิดค่าส่งอื่นๆ ได้

พารามิเตอร์ของตารางการตัดสินใจนี้ได้แก่ จำนวนวันที่ใช้ในการส่งและประเภทสินค้า ดังตัวอย่างข้อความไซพในรูปที่ 4.26

```

<Envelope>
  <Body>
    <Standard_Shipment_Rate_33ParamType>
      <TimeOfArrival_timeOfArrival>1</TimeOfArrival_timeOfArrival>
      <product_category>Toys</product_category>
    </Standard_Shipment_Rate_33ParamType>
  </Body>
</Envelope>

```

รูปที่ 4.26 ตัวอย่างข้อความไซพของพารามิเตอร์ของตารางการตัดสินใจกำหนดค่าส่งหลัก

จากตัวอย่างในรูปจำนวนวันที่ใช้ในการส่งคือ 1 วันและประเภทสินค้าคือ ของเล่นผลลัพธ์คือ ค่าส่งต่อครั้งเท่ากับ 510.0 บาท ค่าส่งต่อน้ำหนักหนึ่งกิโลกรัม 60.0 บาท ค่าส่งต่อชิ้นไม่คิด ดังข้อความไซพในรูปที่ 4.27

```
<Envelope>
  <Body>
    <Standard_Shipment_Rate_33ResultType>
      <price_per_shipment>510.0</price_per_shipment>
      <price_per_item>0.0</price_per_item>
      <price_per_weight>60.0</price_per_weight>
    </Standard_Shipment_Rate_33ResultType>
  </Body>
</Envelope>
```

รูปที่ 4.27 ตัวอย่างข้อความไซพของผลลัพธ์ของตารางการตัดสินใจกำหนดค่าส่งหลัก

2) ตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม

พารามิเตอร์ของตารางการตัดสินใจนี้ได้แก่ วิธีการชำระเงิน ประเภทลูกค้าและราคาสินค้ารวม ดังข้อความไซพในรูปที่ 4.28

```
<Body>
  <Discount_policy_37ParamType>
    <Payment_Option_paymentType>Credit Card</Payment_Option_paymentType>
    <customerTypeCondition_CustomerType>member</customerTypeCondition_CustomerType>
    <TotalPrice_totalPrice>2000</TotalPrice_totalPrice>
  </Discount_policy_37ParamType>
</Body>
```

รูปที่ 4.28 ตัวอย่างข้อความไซพของพารามิเตอร์ของตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม

จากตัวอย่างในรูปจำนวนวันที่ใช้ในการส่งคือ วิธีการชำระเงินคือ บัตรเครดิต ประเภท
ลูกค้าคือ สมาชิกและราคาสินค้ารวมคือ 2000 บาท

ผลลัพธ์คือส่วนลด 10% และของแถมคือนาฬิกาปลุก ดังข้อความไซฟในรูปที่ 4.29

```
<Envelope>
  <Body>
    <Discount_policy_37ResultType>
      <percent_of_discount>10.0</percent_of_discount>
      <bonus>alarm clock</bonus>
    </Discount_policy_37ResultType>
  </Body>
</Envelope>
```

รูปที่ 4.29 ตัวอย่างข้อความไซฟของผลลัพธ์ของตารางการตัดสินใจกำหนดเปอร์เซ็นต์
ส่วนลดและของแถม

3) กระบวนการธุรกิจ

กระบวนการธุรกิจในกรณีทดสอบนี้เกี่ยวข้องกับการสั่งซื้อสินค้าผ่านเว็บไซต์ เพื่อหาราคา
รวม ค่าขนส่ง เปอร์เซนต์ส่วนลดและของแถม พารามิเตอร์ได้แก่ ประเภทลูกค้า (เป็นสมาชิก
หรือไม่เป็น) จำนวนวันที่ส่งสินค้า วิธีการชำระเงินและรายการสินค้าที่สั่งซื้อ ดังข้อความไซฟในรูป
ที่ 4.30

จากตัวอย่างข้อความไซฟในรูปประเภทลูกค้าคือ สมาชิก จำนวนวันที่ส่งสินค้าคือ 1 วัน
วิธีการชำระเงินคือ บัตรเครดิตและรายการสินค้าที่สั่งซื้อ คือ สินค้ารหัส 1 2 และ 3

ผลลัพธ์ คือ ราคารวมเท่ากับ 1600 บาท ค่าขนส่งรวม 435 บาท เปอร์เซนต์ส่วนลด 5.0 %
และของแถมคือ พวงกุญแจ ดังข้อความไซฟในรูปที่ 4.31


```

<Envelope>
  <Body>
    <placeOrderOperation>
      <customerType>member</customerType>
      <delieveryETA>1</delieveryETA>
      <paymentMethod>Credit Card</paymentMethod>
      <items><itemsType>
        <itemId>1</itemId>
        <itemId>2</itemId>
        <itemId>3</itemId>
      </itemsType></items>
    </placeOrderOperation>
  </Body>
</Envelope>

```

รูปที่ 4.30 ตัวอย่างข้อความโซพของพารามิเตอร์ของกระบวนการธุรกิจสั่งซื้อสินค้าผ่านเว็บไซต์

```

<Envelope>
  <Body>
    <placeOrderOperationResponse>
      <totalPrice>1600</totalPrice>
      <percent_discount>5.0</percent_discount>
      <totalShipmentPrice>435</totalShipmentPrice>
      <bonus>key holder</bonus>
    </placeOrderOperationResponse>
  </Body>
</Envelope>

```

รูปที่ 4.31 ตัวอย่างข้อความโซพของผลลัพธ์ของกระบวนการธุรกิจสั่งซื้อสินค้าผ่านเว็บไซต์

4.4 การทดสอบประสิทธิภาพการให้บริการของเว็บเซอร์วิสธุรกิจ

งานวิจัยนี้ทดสอบประสิทธิภาพการให้บริการของเว็บเซอร์วิสธุรกิจด้วยโปรแกรมโซฟุไอ (SoapUI) เวอร์ชัน 2.0 สำหรับเว็บเซอร์วิสธุรกิจทำงานบน Sun Java System Application Server เวอร์ชัน 9.1 ซึ่งมีเครื่องประมวลผลบีเพิลจึงสามารถอ่าน และประมวลผลไฟล์บีเพิลเพื่อให้งานเป็นเว็บเซอร์วิสได้

4.4.1 สภาพแวดล้อมในการทดสอบ

เครื่องคอมพิวเตอร์ที่ใช้ทดสอบมีดังนี้

1) เครื่องผู้ให้บริการเว็บเซอร์วิส ติดตั้งโปรแกรม SoapUI ทำหน้าที่เรียกใช้เว็บเซอร์วิสธุรกิจ ระบบปฏิบัติการ Windows XP Service Pack 2 ซีพียู Pentium Xeon 3.06 GHz แรม 1 กิกะไบต์ ฮาร์ดดิสก์ 34 กิกะไบต์ พื้นที่ว่าง 14 กิกะไบต์

2) เครื่องผู้ให้บริการเว็บเซอร์วิส ติดตั้ง Sun Java System Application Server ระบบปฏิบัติการ Windows 2003 ซีพียู Pentium 4 3.00 GHz แรม 2 กิกะไบต์ ฮาร์ดดิสก์ 10 กิกะไบต์ พื้นที่ว่าง 6 กิกะไบต์

4.4.2 การทดสอบเวลาที่ใช้ในการทำงานของเว็บเซอร์วิส

การทดสอบนี้ใช้โปรแกรมโซฟุไอซึ่งติดตั้งที่เครื่องผู้ให้บริการเว็บเซอร์วิส 1 เครื่อง ทำการจำลองการเรียกใช้บริการจากผู้ใช้พร้อมๆ กัน เป็นจำนวน 1 10 20 30 50 60 70 80 100 150 และ 200 โดยเรียกใช้ติดต่อกันเป็นเวลา 60 วินาที

จากนั้นทำการจับเวลาเฉลี่ยที่ใช้ในการเรียกใช้เว็บเซอร์วิส ซึ่งนับตั้งแต่เริ่มเรียกใช้จนได้รับผลลัพธ์ เว็บเซอร์วิสธุรกิจที่จะทดสอบมาจากตารางการตัดสินใจดังนี้

1) ตารางการตัดสินใจกำหนดเปอร์เซ็นต์ส่วนลดและของแถม ตารางการตัดสินใจนี้มีการกระทำเป็นเพียงการคืนค่ากลับไปยังผู้เรียกใช้เท่านั้น และไม่มีการเรียกตารางย่อยแต่อย่างใด

2) ตารางการตัดสินใจกำหนดค่าส่งหลัก ตารางการตัดสินใจนี้มีการกระทำเรียกใช้ตารางย่อย ส่วนการกระทำในตารางย่อยเป็นเพียงการคืนค่ากลับไปยังตารางหลักเท่านั้น

3) ตารางการตัดสินใจคำนวณค่าทดแทนกรณีลูกค้าไม่สามารถทำงานได้ ตารางการตัดสินใจนี้มีการกระทำเป็นการเรียกใช้เว็บเซอร์วิส 1 ครั้ง และไม่มีการเรียกตารางย่อย

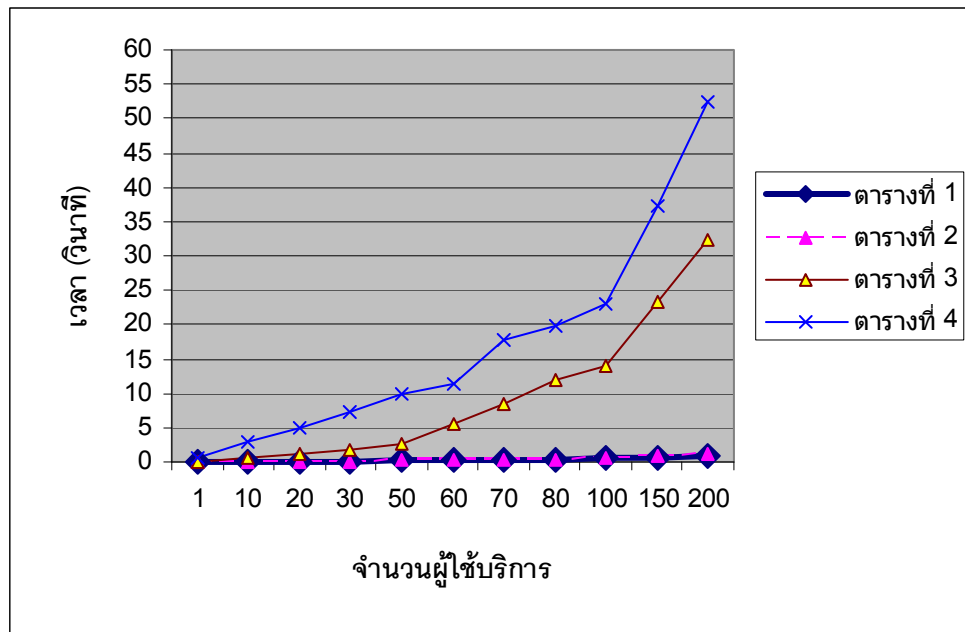
4) ตารางการตัดสินใจคำนวณค่าทดแทนทั้งหมด ตารางการตัดสินใจนี้มีการกระทำ 2 กระทำ การกระทำแรกเป็นการเรียกใช้เว็บเซอร์วิส การกระทำที่สองเป็นการเรียกตารางย่อยซึ่งมีการกระทำเป็นการเรียกใช้เว็บเซอร์วิสเช่นกัน นั่นคือใน 1 ธุรกรรมมีการเรียกใช้เว็บเซอร์วิส 2 ครั้ง

4.4.3 ผลการทดสอบ

จากการจับเวลาเฉลี่ยที่ใช้ในการเรียกใช้เว็บเซอร์วิส ซึ่งนับตั้งแต่เริ่มเรียกใช้จนได้รับผลลัพธ์ ได้ผลการทดสอบได้ผลดังตารางที่ 4.1 และเมื่อนำข้อมูลที่ได้จากการทดสอบมาเขียนกราฟเพื่อเปรียบเทียบข้อมูล ได้ผลดังรูปที่ 4.32

ตารางที่ 4.1 ผลการทดสอบเวลาที่ใช้ในการเรียกใช้เว็บเซอร์วิสกฎธุรกิจ

จำนวนผู้ใช้งาน	เวลาที่ใช้เฉลี่ย (วินาที)			
	ตารางที่ 1	ตารางที่ 2	ตารางที่ 3	ตารางที่ 4
1	0.016	0.016	0.053	0.550
10	0.050	0.047	0.520	2.768
20	0.094	0.097	1.140	5.075
30	0.133	0.143	1.755	7.401
50	0.257	0.233	2.647	9.882
60	0.304	0.295	5.471	11.250
70	0.355	0.376	8.447	17.800
80	0.395	0.365	11.833	19.742
100	0.496	0.475	13.967	23.091
150	0.705	0.779	23.220	37.276
200	1.008	1.040	32.430	52.431



รูปที่ 4.32 กราฟแสดงการเปรียบเทียบระหว่างจำนวนผู้ใช้บริการพร้อมกัน และเวลาเฉลี่ยต่อการเรียกใช้เว็บเซอวิส 1 ครั้ง

4.5 สรุปผลการทดสอบ

ในการทดสอบความสามารถในการให้บริการ ปรากฏว่าเว็บเซอวิสกฎธุรกิจสามารถให้บริการได้อย่างถูกต้อง ทั้งการทดสอบเว็บเซอวิสแบบเดี่ยวๆ และการทดสอบกระบวนการธุรกิจที่ใช้เว็บเซอวิสเหล่านั้น

ในส่วนการทดสอบประสิทธิภาพการให้บริการของเว็บเซอวิสกฎธุรกิจ จากการทดสอบพบว่า จำนวนผู้ใช้บริการเว็บเซอวิสพร้อมๆ กัน จะมีผลต่อเวลาที่ใช้ในการบริการของเว็บเซอวิส นั่นคือ ถ้ามีผู้ใช้บริการเว็บเซอวิสพร้อมๆ มากขึ้น เวลาที่ใช้ในการบริการของเว็บเซอวิสก็จะมากขึ้น เช่นกัน

นอกจากนี้ลักษณะของตารางการตัดสินใจก็มีผลต่อเวลาในการให้บริการเช่นเดียวกัน ถ้าตารางการตัดสินใจที่การกระทำไม่มีการเรียกใช้เว็บเซอวิสอื่น (เว็บเซอวิสที่ไม่ใช่เว็บเซอวิสกฎธุรกิจ) จะใช้เวลาให้บริการน้อยกว่าตารางการตัดสินใจที่การกระทำมีการเรียกใช้เว็บเซอวิสอื่น และยิ่งการกระทำมีการเรียกใช้เว็บเซอวิสอื่นมากขึ้นเท่าใด ก็จะใช้เวลามากขึ้น

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอระบบสนับสนุนการกำหนดกฎธุรกิจด้วยภาษานิยามกฎธุรกิจหรือภาษาปีอาร์ดีแอลและตารางการตัดสินใจ ผู้ใช้สร้างกฎธุรกิจด้วยเครื่องมือบรรณาธิกรณกฎธุรกิจผ่านเว็บแอปพลิเคชันซึ่งงานวิจัยนี้พัฒนาขึ้น โดยมีความสามารถในการสร้าง แก้ไข จัดการกฎธุรกิจ ตรวจสอบความกำกวมและแสดงกฎธุรกิจในรูปตารางการตัดสินใจ นอกจากนี้ยังสามารถสร้างตารางการตัดสินใจจากเงื่อนไขที่กำหนดได้ด้วย กฎธุรกิจที่สร้างขึ้นสามารถทำงานกับกระบวนการธุรกิจและเว็บเซอร์วิสได้

งานวิจัยนี้ออกแบบภาษาปีอาร์ดีแอลเพื่อกำหนดกฎธุรกิจ ภาษานี้อยู่บนพื้นฐานของภาษาเอกซ์เอ็มแอลซึ่งประกอบด้วยแท็กต่างๆ ภาษานี้แบ่งแท็กออกเป็น 3 กลุ่ม ได้แก่ กลุ่มแท็กกำหนดข้อเท็จจริง กลุ่มแท็กกำหนดเซตกฎธุรกิจ และกลุ่มแท็กกำหนดกฎธุรกิจ ทั้ง 3 กลุ่มถูกใช้ในไฟล์ 3 ประเภท คือ ไฟล์ข้อเท็จจริง ไฟล์เซตกฎธุรกิจและไฟล์กฎธุรกิจ ตามลำดับ

ไฟล์ข้อเท็จจริงเก็บแบบเงื่อนไข และการกระทำเพื่อเป็นวัตถุดิบสำหรับสร้างกฎธุรกิจ แบบแบบเหล่านี้กำหนดภาษาธรรมชาติ และตัวแปรปรับแต่งที่ผู้ใช้ต้องกำหนด แบบแบบการกระทำสามารถประกอบด้วยกิจกรรมการเรียกใช้เว็บเซอร์วิส และกิจกรรมการคืนค่ากลับไปยังผู้ใช้ นอกจากนี้ยังสามารถเรียกใช้ตารางการตัดสินใจอื่นได้อีกด้วย ผู้ใช้สามารถสร้างกฎธุรกิจจากเครื่องมือบรรณาธิกรณกฎธุรกิจซึ่งจะดึงข้อมูลแบบเงื่อนไขและการกระทำจากไฟล์ข้อเท็จจริง และแสดงเงื่อนไขและการกระทำในรูปของภาษาธรรมชาติ

ไฟล์เซตกฎธุรกิจทำหน้าที่เก็บกฎธุรกิจทั้งหมดในเซตกฎธุรกิจหรือตารางการตัดสินใจที่สร้างขึ้น ไฟล์เซตกฎธุรกิจสร้างโดยตัวสร้างกฎธุรกิจ ซึ่งระบุแบบเงื่อนไขและการกระทำที่ใช้ในกฎธุรกิจและค่าของตัวแปรปรับแต่งที่ผู้ใช้กำหนด โดยรับข้อมูลจากผู้ใช้ผ่านเครื่องมือบรรณาธิกรณกฎธุรกิจ

ไฟล์กฎธุรกิจจะทำหน้าที่เก็บข้อมูลเกี่ยวกับไฟล์เซตกฎธุรกิจและไฟล์ข้อเท็จจริงทั้งหมด เช่น ชื่อไฟล์เซตกฎธุรกิจ และไฟล์บีบอัดของมัน เป็นต้น จากนั้นกฎธุรกิจทั้งหมดจะถูกตัวแปลงบีบอัดเป็นไฟล์บีบอัดและไฟล์บรรยายเว็บเซอร์วิส ซึ่งสามารถนำไปประมวลผลด้วยเครื่องประมวลผลบีบอัดเพื่อให้ตารางการตัดสินใจสามารถทำงานให้บริการในรูปของเว็บเซอร์วิสธุรกิจ ซึ่งช่วยสนับสนุนการทำงานของกระบวนการธุรกิจอื่นๆ ได้ ด้วยแนวทางที่ว่านี้ผู้ใช้สามารถพัฒนากฎธุรกิจที่สามารถทำงานร่วมกับกระบวนการธุรกิจในระบบสารสนเทศขององค์กรได้ โดยไม่จำเป็นต้องมีความรู้เกี่ยวกับเว็บเซอร์วิสและภาษาบีบอัดแต่อย่างใด

จากการทดสอบพบว่าเว็บเซอวิสกฎธุรกิจให้บริการตามที่กำหนดในตารางการตัดสินใจได้อย่างถูกต้อง สำหรับประสิทธิภาพนั้นจำนวนผู้ใช้บริการเว็บเซอวิสพร้อมๆ กันมีผลต่อเวลาในการเรียกใช้เว็บเซอวิสกฎธุรกิจไม่มาก แต่ลักษณะของการกระทำในกฎธุรกิจมีผลต่อเวลาในการเรียกใช้ให้เห็นได้ชัด ถ้าการกระทำมีการเรียกใช้เว็บเซอวิสอื่นมากเท่าใด เวลาในการเรียกใช้เว็บเซอวิสกฎธุรกิจจะเพิ่มขึ้นอย่างมาก

5.2 ประโยชน์ที่ได้รับ

- 1) สามารถพัฒนาเครื่องมือที่ช่วยอำนวยความสะดวกในการสร้าง และแก้ไขกฎธุรกิจ
- 2) กฎธุรกิจที่ได้สามารถใช้ซ้ำ และแก้ไขได้อย่างคล่องตัว
- 3) กฎธุรกิจที่ได้สามารถสนับสนุนกระบวนการธุรกิจผ่านเทคโนโลยีเว็บเซอวิส และสามารถตอบสนองต่อเป้าหมาย และการดำเนินธุรกิจขององค์กร

5.3 ข้อจำกัด

- 1) ลักษณะของข้อมูลของพารามิเตอร์และผลลัพธ์แต่ละตัวของตารางการตัดสินใจต้องเป็นข้อมูลประเภทตัวเลข สายอักขระและบูลีนเท่านั้น ไม่สามารถเป็นข้อมูลประเภทวันที่หรือเวลาได้ นอกจากนี้ยังไม่สามารถเป็นข้อมูลแบบซับซ้อน (Complex type) เช่น ข้อมูลประเภทแถวลำดับ (array) เป็นต้น
- 2) การนำไฟล์บีเพลไปประมวลผลด้วยเครื่องประมวลผลบีเพลเพื่อให้ทำงานเป็นเว็บเซอวิสนั้น ยังไม่ใช่รูปแบบอัตโนมัติ นั่นคือต้องนำไฟล์บีเพลและไฟล์บรรยายเว็บเซอวิสของมันไปเปิดในโปรแกรมสำหรับแก้ไขไฟล์บีเพลก่อน เช่น NetBeans และดีพลอย (Deploy) ไฟล์บีเพลจึงจะสามารถทำงานในรูปเว็บเซอวิสได้

5.4 ข้อเสนอแนะ

- 1) การสร้างไฟล์ข้อเท็จจริงควรทำโดยผู้มีความรู้เกี่ยวกับเว็บเซอวิสและโครงสร้างของไฟล์อย่างดี เพราะผู้สร้างต้องเขียนไฟล์ข้อเท็จจริงซึ่งเป็นไฟล์เอกซ์เอ็มแอลโดยตรง ไม่มีเครื่องมือช่วยแต่อย่างใด
- 2) ผู้สร้างแผนแบบเงื่อนไขและการกระทำในไฟล์ข้อเท็จจริง ควรระบุชื่อ คำบรรยายและภาษาธรรมชาติของแผนแบบที่สื่อความหมายและเข้าใจง่ายสำหรับผู้สร้างกฎธุรกิจ
- 3) ถ้าตัวแปรของแผนแบบเงื่อนไขและการกระทำที่ค่าที่ชัดเจนและจำกัด ผู้สร้างแผนแบบเงื่อนไขและการกระทำใน ควรใช้แท็ก enum เพื่อระบุรายการของค่าที่เป็นไปได้ของตัวแปรปรับแต่ง เพื่อช่วยลดความผิดพลาดของการกำหนดกฎธุรกิจ

4) การเรียกใช้เว็บเซอวิสกฎธุรกิจ ต้องอ่านไฟล์บรรยายเว็บเซอวิสของเว็บเซอวิสนั้นก่อน ซึ่งในไฟล์จะระบุชื่อเครื่องหรือเลขที่อยู่ไอพี (IP address) ของเว็บเซอวิสไว้ แต่ไฟล์บรรยายเว็บเซอวิสของเว็บเซอวิสกฎธุรกิจจะระบุค่าที่อยู่โดยปริยายเป็น localhost ดังนั้นผู้เรียกใช้ต้องแก้ไขให้เป็นที่อยู่ที่แท้จริงที่เว็บเซอวิสทำงานเสียก่อน จึงจะเรียกใช้ได้

5) ถ้าตารางการตัดสินใจมีการเรียกตารางย่อย ควรสร้างตารางย่อยให้เสร็จเรียบร้อยก่อน จึงค่อยสร้างตารางการตัดสินใจหลักที่จะเรียกตารางย่อยนั้น

รายการอ้างอิง

- [1] Halle, B.V. Business rules applied. USA: John Wiley & Son, 2001.
- [2] ILOG. Decision services: The next soa challenge [Online]. 2006. Available from: <http://www.ilog.com/products/jrules/whitepapers/index.cfm?filename=WPSOA.pdf>
- [3] Norton, M. Decisioning: A new approach to systems development (part 1) [Online]. 2007. Available from: <http://www.brcommunity.com/b326a.php>
- [4] Andrews, T. et al. Business process execution language for web services version 2.0 [Online]. 2007. Available from: <http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel>
- [5] Nainai B. et al. Building flexible enterprise processes using oracle business rules and bpel process manager [Online]. 2005. Available from: http://www.oracle.com/technology/products/ias/business_rules/pdf/bpelAndRules.pdf
- [6] Charfi, A., and Mezini, M. Hybrid Web Service Composition: Business Processes Meet Business Rules. Proceedings of the 2nd international conference on Service oriented computing (2004): 30-38.
- [7] Monson-Haefel, R. J2EE™ web services. Boston: Addison Wesley Professional, 2003.
- [8] Harney, J. Decision table [Online]. Available from: <http://www.cems.uwe.ac.uk/jharney/table.html>
- [9] Vanthienen, J., Wets, G. Integration of the Decision Table Formalism with a Relational Database Environment. Information Systems archive 7 (1995): 595- 616.
- [10] Codasyl. A Modern Appraisal of Decision Tables. Report of the Decision Table Task Group (1982): 230-232.
- [11] Cerami, E. Web services essentials. UK: O'Reilly Media, 2002.
- [12] Newcomer, E., and Lomow, G. Understanding soa with web services. Boston: Addison Wesley Professional, 2004.
- [13] Michael, K. XPath 2.0 programmer's reference. USA: Wiley Publishing, 2004.

- [14] Merino, L.B., and Elguezabal, G.B. Business Process Definition Languages Versus Traditional Methods Towards Interoperability. Lecture Notes in Computer Science 3412 (2005): 25-35.
- [15] Erl, T. Service-oriented architecture: Concepts, technology, and design. New Jersey: Prentice Hall, 2005.
- [16] Deursen, A.V., Klint, P., and Visser, J. Domain-Specific Languages: An Annotated Bibliograph. ACM SIGPLAN Notices 35 (2000): 26-36.
- [17] Thibault, S.A., Marlet, R., and Consel, C. Domain-Specific Languages: From Design to implementation Application to video device drivers generation. IEEE Transactions on Software Engineering (1999): 363-377.
- [18] Boley, H., Tabet, S., and Wagner, G. Design Rationale of RuleML: A Markup Language for Semantic Web Rules. Proceeding of the International Semantic Web Working Symposium (2001).
- [19] Costello, C., and Molloy, O. Orchestrating Supply Chain Interactions Using Emerging Process Description Language and Business Rules. Sixth International Conference on Electronic Commerce (2004): 21-30.
- [20] Schmidt, R. Web Services Based Execution of Business Rules. Proceeding of the International Workshop on Rule Markup Language for Business Rules on the Semantic Web 60 (2002).
- [21] Hamdi, H., Mosbah, M., and Bouhoula, A. A Domain Specific Language for Securing Distributed Systems. Systems and Networks Communications International Conference (2007): 76.
- [22] Peltier, M. MTrans, a DSL for Model Transformation. Proceedings of the Sixth International ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference (2002): 190.
- [23] OMG. Object Constraint Language (OCL) specification version 2.0 [Online]. 2006. Available from: <http://www.omg.org/docs/ptc/03-10-14.pdf>
- [24] สุขโขทัยธรรมมาธิราช, มหาวิทยาลัย. สาขาวิชาวิทยาการจัดการ. การประกันชีวิต 2. นนทบุรี: สำนักพิมพ์มหาวิทยาลัยสุโขทัยธรรมมาธิราช, 2538.

- [25] ประกอบ ตันติยาพงศ์ รองปลัดกระทรวงการคลัง รักษาราชการแทน ปลัดกระทรวงการคลัง, “ประกาศกระทรวงการคลังเรื่อง หลักเกณฑ์การชดใช้ค่าสินไหมทดแทนกรณีเจ้าหน้าที่ของรัฐกระทำผิดละเมิดต่อบุคคลภายนอก”, 27 กันยายน 2545.

ภาคผนวก

ภาคผนวก ก

รายละเอียดคลาสและเม็ทอดของคลาสภายในแพ็คเกจ Business Logic

คลาส Fact

เป็นคลาสนามธรรม (Abstract class) ที่ใช้กำหนดคุณสมบัติ (Attribute) และ เม็ทอด (method) พื้นฐานที่จำเป็นต้องมีของคลาสที่เกี่ยวข้องกับข้อเท็จจริง มีรายละเอียดดังแสดงในตารางที่ ก-1

ตารางที่ ก-1 รายละเอียดของคุณสมบัติและเม็ทอดของคลาส Fact

คุณสมบัติ	รายละเอียด
id	รหัสของข้อเท็จจริง
name	ชื่อของข้อเท็จจริง
naturalLanguage	ภาษารวมชาติของเท็จจริงสำหรับแสดงแก่ผู้ใ้
description	คำบรรยายของข้อเท็จจริง
customVariables	เก็บชุดของอ็อบเจกต์ Variable เป็นชุดตัวแปรปรับแต่งของข้อเท็จจริง
เม็ทอด	รายละเอียด
AddCustomVariable()	เพิ่มอ็อบเจกต์ Variable ที่กำหนด
GetCustomVariable()	ส่งอ็อบเจกต์ Variable ตามค่า id ที่กำหนด

คลาส Condition

เป็นคลาสที่สืบทอดจากคลาส Fact ทำหน้าที่เป็นแผ่นแบบเงื่อนไขของกฎธุรกิจ คลาสไม่มีการเพิ่มเติมคุณสมบัติและเม็ทอดจากคลาส Fact

คลาส Action

เป็นคลาสที่สืบทอดจากคลาส Fact ทำหน้าที่เป็นแผ่นแบบการกระทำของกฎธุรกิจ มีรายละเอียดดังแสดงในตารางที่ ก-2

ตารางที่ ก-2 รายละเอียดของคุณสมบัติและเมธอดของคลาส Action

คุณสมบัติ	รายละเอียด
variables	เก็บชุดของอ็อบเจกต์ Variable หรือตัวแปรเฉพาะที่อยู่ในแผนแบบการกระทำ
activities	เก็บชุดของอ็อบเจกต์ Activity หรือกิจกรรมภายในแผนแบบการกระทำ
เมธอด	รายละเอียด
AddActivity()	เพิ่มอ็อบเจกต์ Activity ที่กำหนด
AddVariable()	เพิ่มอ็อบเจกต์ Variable ที่กำหนด
GetVariable()	ส่งอ็อบเจกต์ Variable ตามค่า id ที่กำหนด

คลาส CallSubRulesetAction

เป็นคลาสที่สืบทอดจากคลาส Action ทำหน้าที่เป็นแผนแบบการกระทำสำหรับเรียกเซตกฎธุรกิจย่อย

คลาส Activity

เป็นคลาสที่เก็บกิจกรรมที่อยู่ภายในแผนแบบการกระทำ มีรายละเอียดดังแสดงในตารางที่ ก-3

ตารางที่ ก-321 รายละเอียดของคุณสมบัติและเมธอดของคลาส Activity

คุณสมบัติ	รายละเอียด
parameters	เก็บชุดของอ็อบเจกต์ Parameter หรือพารามิเตอร์ของกิจกรรม
เมธอด	รายละเอียด
AddParameter()	เพิ่มอ็อบเจกต์ Parameter ที่กำหนด
RemoveParameter()	ลบอ็อบเจกต์ Parameter ที่กำหนด

คลาส ReturnValueActivity

เป็นคลาสที่สืบทอดจากคลาส Activity ทำหน้าที่เป็นกิจกรรมแบบคืนค่ากลับไปยังผู้เรียกใช้

คลาส InvokeWSActivity

เป็นคลาสที่สืบทอดจากคลาส Activity ทำหน้าที่เป็นกิจกรรมแบบเรียกใช้เว็บเซอร์วิส มีรายละเอียดดังแสดงในตารางที่ ก-4

ตารางที่ ก-4 รายละเอียดของคุณสมบัติและเมทอดของคลาส InvokeWSActivity

คุณสมบัติ	รายละเอียด
wSDL	ไฟล์บรรยายเว็บเซอร์วิสของเว็บเซอร์วิสที่กิจกรรมเรียกใช้
service	ชื่อเว็บเซอร์วิสที่กิจกรรมเรียกใช้
operation	ชื่อการดำเนินการของเว็บเซอร์วิสที่กิจกรรมเรียกใช้
partnerLinkType	ชื่อพาร์ตเนอร์ลิงค์ของเว็บเซอร์วิส
role	ชื่อบทบาทของเว็บเซอร์วิส
resultVariables	รายการผลลัพธ์ของเว็บเซอร์วิส
เมทอด	รายละเอียด
AddResultVariables ()	เพิ่มผลลัพธ์ของเว็บเซอร์วิส

คลาส Parameter

เป็นคลาสที่ทำหน้าที่เป็นพารามิเตอร์ของกิจกรรม มีรายละเอียดดังแสดงในตารางที่ ก-5

ตารางที่ ก-5 รายละเอียดของคุณสมบัติและเมทอดของคลาส Parameter

คุณสมบัติ	รายละเอียด
variableId	ตัวแปรเฉพาะที่ของแผนแบบการกระทำที่ถูกกำหนดค่าให้พารามิเตอร์
customVariableId	ตัวแปรปรับแต่งของแผนแบบการกระทำที่ถูกกำหนดค่าให้พารามิเตอร์
isConstantValue	ระบุว่าพารามิเตอร์เป็นค่าคงที่หรือไม่
constantValue	ค่าของค่าคงที่ของพารามิเตอร์
constantValueType	ประเภทข้อมูลของค่าคงที่ของพารามิเตอร์

ตารางที่ ก-5 รายละเอียดของคุณสมบัติและเมธอดของคลาส Parameter (ต่อ)

คุณสมบัติ	รายละเอียด
toElement	ชื่อพารามิเตอร์ของเว็บเซอริวิสที่กิจกรรมเรียก
เมธอด	รายละเอียด
GetVariable()	ส่งอ็อบเจกต์ Variable ซึ่งค่าถูกกำหนดให้แก่พารามิเตอร์

คลาส SubRulesetParameter

เป็นคลาสที่สืบทอดมาจากคลาส Parameter ทำหน้าที่เป็นพารามิเตอร์ของการเรียกเซตกฎธุรกิจย่อยมีรายละเอียดดังแสดงในตารางที่ ก-6

ตารางที่ ก-6 รายละเอียดของคุณสมบัติและเมธอดของคลาส SubRulesetParameter

คุณสมบัติ	รายละเอียด
customDtParameter	พารามิเตอร์ของเซตกฎธุรกิจที่กำหนดโดยผู้ใช้ซึ่งค่าถูกกำหนดให้แก่พารามิเตอร์ของเซตกฎธุรกิจย่อย
Condition	แผ่นแบบเงื่อนไขที่ค่าของตัวแปรปรับแต่งถูกกำหนดให้แก่พารามิเตอร์ของเซตกฎธุรกิจย่อย
returnVariable	ผลลัพธ์ของเซตกฎธุรกิจซึ่งค่าถูกกำหนดให้แก่พารามิเตอร์ของเซตกฎธุรกิจย่อย

คลาส Variable

เป็นคลาสที่ทำหน้าที่เป็นตัวแปรปรับแต่งของข้อเท็จจริงและตัวแปรเฉพาะที่ (Local variable) ของการกระทำ มีรายละเอียดดังแสดงในตารางที่ ก-7

ตารางที่ ก-7 รายละเอียดของคุณสมบัติและเมธอดของคลาส Variable

คุณสมบัติ	รายละเอียด
Id	รหัสของตัวแปร
Name	ชื่อของตัวแปร
Type	ประเภทข้อมูลตัวแปร

ตารางที่ ก-7 รายละเอียดของคุณสมบัติและเมทอดของคลาส Variable (ต่อ)

คุณสมบัติ	รายละเอียด
isConstantValue	ระบุว่าตัวแปรเป็นค่าคงที่หรือไม่
constantValue	ค่าคงที่ของตัวแปร
enumList	เก็บชุดของอ็อบเจกต์ VariableEnum เป็นชุดของค่าที่เป็นไปได้ของตัวแปร
เมทอด	รายละเอียด
AddEnum()	เพิ่มอ็อบเจกต์ VariableEnum ที่กำหนด
GetEnumTextByValue()	ส่งอ็อบเจกต์ VariableEnum ตามค่า value ที่กำหนด
GetEnumCount()	ส่งจำนวนอ็อบเจกต์ VariableEnum
GetOperatorText()	ส่งค่าตัวดำเนินการเปรียบเทียบสำหรับแสดงผลแก่ผู้ใช้

คลาส VariableEnum

เป็นคลาสที่ทำหน้าที่เป็นค่าที่เป็นไปได้ของตัวแปร มีรายละเอียดดังแสดงในตารางที่ ก-8

ตารางที่ ก-8 รายละเอียดของคุณสมบัติและเมทอดของคลาส VariableEnum

คุณสมบัติ	รายละเอียด
Text	ค่าสำหรับแสดงผลแก่ผู้ใช้
Value	ค่าจริงของอ็อบเจกต์

คลาส Rule

เป็นคลาสที่ทำหน้าที่เป็นกฎธุรกิจซึ่งประกอบด้วยเงื่อนไขและการกระทำ มีรายละเอียดดังแสดงในตารางที่ ก-9

ตารางที่ ก-9 รายละเอียดของคุณสมบัติและเมทอดของคลาส Rule

คุณสมบัติ	รายละเอียด
Id	รหัสของกฎธุรกิจ
Name	ชื่อของกฎธุรกิจ

ตารางที่ ก-9 รายละเอียดของคุณสมบัติและเมทอดของคลาส Rule (ต่อ)

คุณสมบัติ	รายละเอียด
Order	ลำดับของกฎธุรกิจในเซตกฎธุรกิจ
isDefault	ระบุว่ากฎธุรกิจนี้เป็นกฎปริยายหรือไม่
Group	กลุ่มที่กฎธุรกิจนี้อยู่
andOrOperator	ตัวดำเนินการตรรกะของเงื่อนไขในกฎธุรกิจ (แอนด์ หรือ ออร์)
conditionsOfRule	ชุดของอ็อบเจกต์ ValueOfCondition ซึ่งเป็นค่าของเงื่อนไขที่กำหนดโดยผู้ใช้
actionsOfRule	ชุดของอ็อบเจกต์ ValueOfAction ซึ่งเป็นค่าของการกระทำที่กำหนดโดยผู้ใช้
resultAssignment	รายการการกำหนดผลลัพธ์จากการกระทำ ให้อัปกับผลลัพธ์ของเซตกฎธุรกิจ
เมทอด	รายละเอียด
AddActionOfRule()	เพิ่มอ็อบเจกต์ ValueOfAction ของการกระทำที่ต้องการเพิ่มเข้าไปในกฎธุรกิจ
AddConditionOfRule()	เพิ่มอ็อบเจกต์ ValueOfCondition ของเงื่อนไขที่ต้องการเพิ่มเข้าไปในกฎธุรกิจ
DeleteAction()	ลบการกระทำตามid ที่กำหนด
DeleteCondition()	ลบการเงื่อนไขตามid ที่กำหนด
GetValuesOfAction()	ส่งชุดของอ็อบเจกต์ ValueOfAction ตาม id ของการกระทำที่กำหนด
GetValuesOfCondition()	ส่งชุดของอ็อบเจกต์ ValueOfCondition ตาม id ของเงื่อนไขที่กำหนด
GetConflictConditons()	ส่งชุดของ id ของแผ่นแบบเงื่อนไข ที่เงื่อนไขในกฎธุรกิจที่อ้างถึงที่ขัดแย้งกัน
IncreaseActionOrder()	เลื่อนลำดับของการกระทำขึ้น
IncreaseConditionOrder()	เลื่อนลำดับของเงื่อนไขขึ้น
DecreaseActionOrder()	เลื่อนลำดับของการกระทำลง
DecreaseConditionOrder()	เลื่อนลำดับของเงื่อนไขลง

ตารางที่ ก-9 รายละเอียดของคุณสมบัติและเมทอดของคลาส Rule (ต่อ)

เมทอด	รายละเอียด
GetResultAssignment()	ส่งรายการการกำหนดผลลัพธ์ของการกระทำที่กำหนด
AddResultAssignment()	เพิ่มรายการการกำหนดผลลัพธ์
RemoveResultAssignment()	ลบรายการการกำหนดผลลัพธ์ของการกระทำที่กำหนด

คลาส ValueOfCondition

เป็นคลาสที่ทำหน้าที่เก็บค่าของตัวแปรปรับแต่งของแผ่นแบบเงื่อนไข ซึ่งกำหนดโดยผู้ใช้ มีรายละเอียดดังแสดงในตารางที่ ก-10

ตารางที่ ก-10 รายละเอียดของคุณสมบัติและเมทอดของคลาส ValueOfCondition

คุณสมบัติ	รายละเอียด
variableId	รหัสของตัวแปรปรับแต่ง
conditionFactId	รหัสของแผ่นแบบเงื่อนไข
conditionId	รหัสของเงื่อนไข
order	ลำดับของเงื่อนไขในกฎธุรกิจ
boolOperator	ตัวดำเนินการเปรียบเทียบ
leftId	ค่าทางซ้ายของตัวแปร
rightId	ค่าทางขวาของตัวแปร
excludeLeft	กำหนดจะรวมค่าด้านซ้ายด้วยหรือไม่ (กรณีตัวดำเนินการเปรียบเทียบ คือ between)
excludeRight	กำหนดจะรวมค่าด้านขวาด้วยหรือไม่ (กรณีตัวดำเนินการเปรียบเทียบ คือ between)
เมทอด	รายละเอียด
GetValueText()	ส่งค่าของตัวแปรและตัวดำเนินการเปรียบเทียบสำหรับแสดงผล

คลาส ValueOfAction

เป็นคลาสที่ทำหน้าที่เก็บค่าของตัวแปรปรับแต่งของแผ่นแบบการกระทำ ซึ่งกำหนดโดยผู้ใช้ มีรายละเอียดดังแสดงในตารางที่ ก-11

ตารางที่ ก-11 รายละเอียดของคุณสมบัติและเมทอดของคลาส ValueOfAction

คุณสมบัติ	รายละเอียด
actionId	รหัสของการกระทำ
actionFactId	รหัสของแผ่นแบบการกระทำ
variableId	รหัสของตัวแปรปรับแต่ง
isConstantValue	ระบุว่าค่าของตัวแปรปรับแต่งเป็นค่าคงที่หรือไม่
constantValue	ค่าคงที่ของตัวแปรปรับแต่ง
customDtParameterId	รหัสของพารามิเตอร์ของเซตกฎธุรกิจที่กำหนดโดยผู้ใช้
paramCondFactId	รหัสของแผ่นแบบเงื่อนไขที่ใช้ในเซตกฎธุรกิจ
paramCustomVarId	รหัสของตัวแปรปรับแต่งของแผ่นแบบเงื่อนไข
returnVariable	อ็อบเจกต์ ReturnVariable หรือผลลัพธ์ของเซตกฎธุรกิจ
isCallingSubRuleSet	ระบุว่ากรกระทำเป็นการเรียกเซตกฎธุรกิจย่อยหรือไม่
subRulesetId	รหัสของเซตกฎธุรกิจย่อยที่ถูกเรียกใช้
description	คำบรรยายของการเรียกใช้เซตกฎธุรกิจย่อย
subRuleSetParams	ชุดของอ็อบเจกต์ SubTableParameter หรือพารามิเตอร์ของเซตกฎธุรกิจย่อย
เมทอด	รายละเอียด
AddSubRuleSetParam()	เพิ่มอ็อบเจกต์ SubTableParameter หรือพารามิเตอร์ของเซตกฎธุรกิจย่อย
GetSubtableParam()	ส่งอ็อบเจกต์ SubTableParameter ตามค่าของพารามิเตอร์ที่กำหนด
GetValueText()	ส่งค่าของตัวแปรปรับแต่งสำหรับแสดงผล

คลาส RuleSet

เป็นคลาสที่ทำหน้าที่เก็บเซตกฎธุรกิจหรือตารางการตัดสินใจไว้ มีรายละเอียดดังแสดงในตารางที่ ก-12

ตารางที่ ก-12 รายละเอียดของคุณสมบัติและเมทอดของคลาส RuleSet

คุณสมบัติ	รายละเอียด
id	รหัสของเซตกฎธุรกิจ
name	ชื่อของเซตกฎธุรกิจ
description	คำบรรยายของเซตกฎธุรกิจ
groups	ชุดของอ็อบเจกต์ RuleGroup หรือกลุ่มของกฎธุรกิจ
predefinedFacts	อ็อบเจกต์ PredefinedFacts หรือไฟล์ข้อเท็จจริง
category	ประเภทของเซตกฎธุรกิจ
customDtParameters	ชุดของอ็อบเจกต์ DtParameter หรือพารามิเตอร์ที่กำหนดโดยผู้ใช้ของเซตกฎธุรกิจ
returnVariables	ชุดของอ็อบเจกต์ ReturnVariable หรือผลลัพธ์ของเซตกฎธุรกิจ
rules	ชุดของอ็อบเจกต์ Rule หรือกฎธุรกิจ
bpel	อ็อบเจกต์ Bpel หรือไฟล์บีเพลที่ได้จากเซตกฎธุรกิจ
isLoop	ระบุว่าเซตกฎธุรกิจทำงานแบบวนซ้ำหรือไม่
isLoopFix	ระบุว่าการทำงานแบบวนซ้ำเป็นแบบกำหนดรอบตายตัวหรือไม่
loopCount	จำนวนรอบการทำงานแบบวนซ้ำ
loopConditions	ชุดของอ็อบเจกต์ LoopCondition หรือเงื่อนไขการวนซ้ำ
loopCondOpr	ตัวดำเนินการเปรียบเทียบของเงื่อนไขการวนซ้ำ
เมทอด	รายละเอียด
ChangeGroupOfRule()	เปลี่ยนกลุ่มของกฎธุรกิจที่กำหนด
CheckAmbiguous()	ตรวจสอบความกำกวมของเซตกฎธุรกิจ
CheckConflict()	ตรวจสอบความขัดแย้งของเงื่อนไขที่กำหนด
SaveToBpel()	แปลงเซตกฎธุรกิจเป็นไฟล์บีเพล

ตารางที่ ก-12 รายละเอียดของคุณสมบัติและเมทอดของคลาส RuleSet (ต่อ)

เมทอด	รายละเอียด
SaveToPersistence()	บันทึกเซตกฎธุรกิจลงไฟล์เซตกฎธุรกิจ
Split()	แยกเซตกฎธุรกิจออกเป็น 2 เซตกฎธุรกิจ
UpdateRule()	แก้ไขกฎธุรกิจที่กำหนดในไฟล์เซตกฎธุรกิจ

คลาส LoopCondition

เป็นคลาสที่ทำหน้าที่เก็บเงื่อนไขของการวนซ้ำ มีรายละเอียดดังแสดงในตารางที่ ก-13

ตารางที่ ก-13 รายละเอียดของคุณสมบัติและเมทอดของคลาส LoopCondition

คุณสมบัติ	รายละเอียด
id	รหัสของเงื่อนไข
returnVariableId	รหัสของผลลัพธ์ของเซตกฎธุรกิจนำมาเป็นเงื่อนไข
order	ลำดับของเงื่อนไข
boolOperator	ตัวดำเนินการเปรียบเทียบ
leftId	ค่าทางซ้ายของตัวแปร
rightId	ค่าทางขวาของตัวแปร
excludeLeft	กำหนดจะรวมค่าด้านซ้ายด้วยหรือไม่ (กรณีตัวดำเนินการเปรียบเทียบ คือ between)
excludeRight	กำหนดจะรวมค่าด้านขวาด้วยหรือไม่ (กรณีตัวดำเนินการเปรียบเทียบ คือ between)
เมทอด	รายละเอียด
GetValueText()	ส่งค่าของตัวแปรและตัวดำเนินการเปรียบเทียบสำหรับแสดงผล

คลาส RuleGroup

เป็นคลาสที่เก็บข้อมูลของกลุ่มกฎธุรกิจในเซตกฎธุรกิจ มีรายละเอียดดังแสดงในตารางที่

ก-14

ตารางที่ ก-14 รายละเอียดของคุณสมบัติและเมธอดของคลาส RuleGroup

คุณสมบัติ	รายละเอียด
id	รหัสของกลุ่ม
name	ชื่อของกลุ่ม
description	คำบรรยายของกลุ่ม
rules	ชุดของอีอบเจกต์ Rule หรือกฎธุรกิจในกลุ่ม
เมธอด	รายละเอียด
AddRule()	เพิ่มอีอบเจกต์ Rule ที่กำหนด
RemoveRule()	ลบอีอบเจกต์ Rule ที่กำหนด

คลาส Bpel

เป็นคลาสที่เก็บข้อมูลไฟล์บีเพลของเซตกฎธุรกิจ และทำหน้าที่แปลงเซตกฎธุรกิจเป็นภาษาบีเพล มีรายละเอียดดังแสดงในตารางที่ ก-15

ตารางที่ ก-15 รายละเอียดของคุณสมบัติและเมธอดของคลาส BusinessRules

คุณสมบัติ	รายละเอียด
wSDL	เก็บข้อมูลไฟล์บรรยายเว็บเซอร์วิสของไฟล์บีเพล
partnerLinkType	ชื่อพาร์ตเนอร์ลิงค์ไทม์ของไฟล์บีเพล
role	ชื่อบทบาทของไฟล์บีเพล
เมธอด	รายละเอียด
SetWSDL()	กำหนดข้อมูลและบันทึกไฟล์บรรยายเว็บเซอร์วิส
Transform()	แปลงเซตกฎธุรกิจเป็นภาษาบีเพลและบันทึกลงไฟล์บีเพล

คลาส Category

เป็นคลาสที่เก็บข้อมูลของประเภทเซตกฎธุรกิจ มีรายละเอียดดังแสดงในตารางที่ ก-16

ตารางที่ ก-16 รายละเอียดของคุณสมบัติและเมทอดของคลาส Category

คุณสมบัติ	รายละเอียด
id	รหัสของประเภท
name	ชื่อของประเภท
description	คำบรรยายของประเภท
rulesets	ชุดของอ็อบเจกต์ RuleSet หรือเซตกฎธุรกิจประเภท
AddRuleSet()	เพิ่มอ็อบเจกต์ RuleSet ที่กำหนด
เมทอด	รายละเอียด
RemoveRuleSet()	ลบอ็อบเจกต์ RuleSet ที่กำหนด

คลาส ReturnVariable

เป็นคลาสที่เก็บข้อมูลของผลลัพธ์ของเซตกฎธุรกิจ มีรายละเอียดดังแสดงในตารางที่ ก-17

ตารางที่ ก-17 รายละเอียดของคุณสมบัติและเมทอดของคลาส ReturnVariable

คุณสมบัติ	รายละเอียด
id	รหัสของผลลัพธ์
name	ชื่อของผลลัพธ์
type	ประเภทข้อมูลของผลลัพธ์
value	ค่าของผลลัพธ์

คลาส DtParameter

เป็นคลาสที่เก็บข้อมูลของพารามิเตอร์ที่กำหนดโดยผู้ใช้ของเซตกฎธุรกิจ มีรายละเอียดดังแสดงในตารางที่ ก-18

ตารางที่ ก-18 รายละเอียดของคุณสมบัติและเมทอดของคลาส DtParameter

คุณสมบัติ	รายละเอียด
id	รหัสของพารามิเตอร์
name	ชื่อของพารามิเตอร์

ตารางที่ ก-18 รายละเอียดของคุณสมบัติและเมทอดของคลาส DtParameter (ต่อ)

คุณสมบัติ	รายละเอียด
type	ประเภทข้อมูลของพารามิเตอร์

คลาส PredefinedFacts

เป็นคลาสที่เก็บข้อมูลของแผ่นแบบเงื่อนไขและการกระทำจากไฟล์ข้อเท็จจริง มีรายละเอียดดังแสดงในตารางที่ ก-19

ตารางที่ ก-19 รายละเอียดของคุณสมบัติและเมทอดของคลาส PredefinedFacts

คุณสมบัติ	รายละเอียด
id	รหัสของไฟล์ข้อเท็จจริง
name	ชื่อของไฟล์ข้อเท็จจริง
conditions	ชุดของอ็อบเจกต์ Condition หรือแผ่นแบบเงื่อนไข
actions	ชุดของอ็อบเจกต์ Action หรือแผ่นแบบเงื่อนไข
เมทอด	รายละเอียด
AddActions()	เพิ่มชุดของอ็อบเจกต์ Action ที่กำหนด
AddConditions()	เพิ่มชุดของอ็อบเจกต์ Condition ที่กำหนด
GetAction()	ส่งของอ็อบเจกต์ Action ตาม id ที่กำหนด
GetCondition()	ส่งของอ็อบเจกต์ Condition ตาม id ที่กำหนด

คลาส BusinessRules

เป็นคลาสที่เก็บข้อมูลไฟล์ข้อเท็จจริง เซตกฎธุรกิจและประเภทเซตกฎธุรกิจทั้งหมด มีรายละเอียดดังแสดงในตารางที่ ก-20

ตารางที่ ก-20 รายละเอียดของคุณสมบัติและเมทอดของคลาส BusinessRules

คุณสมบัติ	รายละเอียด
predefinedFacts	ชุดของอ็อบเจกต์ PredefinedFacts หรือไฟล์ข้อเท็จจริงทั้งหมด
rulesets	ชุดของอ็อบเจกต์ RuleSet หรือเซตกฎธุรกิจทั้งหมดในระบบ

ตารางที่ ก-20 รายละเอียดของคุณสมบัติและเมทอดของคลาส BusinessRules (ต่อ)

คุณสมบัติ	รายละเอียด
categories	ชุดของอ็อบเจกต์ Category หรือประเภทเซตกฎธุรกิจทั้งหมด
เมทอด	รายละเอียด
ChangeRulesetCategory()	เปลี่ยนประเภทของเซตกฎธุรกิจที่กำหนด
InitCategories()	สร้างอ็อบเจกต์ Category ทั้งหมดที่กำหนดในจากไฟล์กฎธุรกิจ
InitPredefinedFacts()	สร้างอ็อบเจกต์ PredefinedFacts ทั้งหมดที่กำหนดในจากไฟล์กฎธุรกิจ
InitRuleSets()	สร้างอ็อบเจกต์ RuleSet ทั้งหมดที่กำหนดในจากไฟล์กฎธุรกิจ
SaveRuleSetsToBpel()	สร้างไฟล์บีเพลสำหรับอ็อบเจกต์ RuleSet ทั้งหมด

ภาคผนวก ข

รายละเอียดคลาสและเมธอดของคลาสภายในแพ็คเกจ Data Access

คลาส RuleSetDAM

เป็นคลาสที่ทำหน้าที่อ่าน ลบ บันทึก และแก้ไขข้อมูลในไฟล์เซตกฎธุรกิจ รายละเอียดดังแสดงในตารางที่ ข-1

ตารางที่ ข-1 รายละเอียดของเมธอดของคลาส RuleSetDAM

เมธอด	รายละเอียด
InsertActionOfRule()	เพิ่มการกระทำในกฎธุรกิจที่กำหนด
InsertConditionOfRule()	เพิ่มเงื่อนไขในกฎธุรกิจที่กำหนด
InsertRule()	เพิ่มกฎธุรกิจ
InsertCustomDtParameter()	เพิ่มพารามิเตอร์ของเซตกฎธุรกิจที่กำหนดโดยผู้ใช้
InsertReturnVariable()	เพิ่มตัวแปรผลลัพธ์ของเซตกฎธุรกิจ
InsertGroup()	เพิ่มกลุ่มของกฎในเซตกฎธุรกิจ
InsertLoopCondition()	เพิ่มเงื่อนไขของเงื่อนไขการวนซ้ำของเซตกฎธุรกิจ
DeleteActionOfRule()	ลบการกระทำในกฎธุรกิจที่กำหนด
DeleteConditionOfRule()	ลบเงื่อนไขในกฎธุรกิจที่กำหนด
DeleteRule()	ลบกฎธุรกิจที่กำหนด
DeleteCustomDtParameter()	ลบพารามิเตอร์ของเซตกฎธุรกิจที่กำหนดโดยผู้ใช้
DeleteReturnVariable()	ลบตัวแปรผลลัพธ์ของเซตกฎธุรกิจ
DeleteGroup()	ลบกลุ่มของกฎในเซตกฎธุรกิจ
DeleteLoopCondition()	ลบเงื่อนไขของเงื่อนไขการวนซ้ำของเซตกฎธุรกิจ
UpdateActionOfRule()	แก้ไขการกระทำในกฎธุรกิจที่กำหนด
UpdateConditionOfRule()	แก้ไขเงื่อนไขในกฎธุรกิจที่กำหนด
UpdateRule()	แก้ไขกฎธุรกิจที่กำหนด
UpdateCustomDtParameter()	แก้ไขพารามิเตอร์ของเซตกฎธุรกิจที่กำหนดโดยผู้ใช้
UpdateReturnVariable()	แก้ไขตัวแปรผลลัพธ์ของเซตกฎธุรกิจ

ตารางที่ ข-1 รายละเอียดของเมธอดของคลาส RuleSetDAM (ต่อ)

เมธอด	รายละเอียด
UpdateGroup()	แก้ไขกลุ่มของกฎในเซตกฎธุรกิจ
UpdateLoopCondition()	แก้ไขเงื่อนไขของเงื่อนไขการวนซ้ำของเซตกฎธุรกิจ
GetAllRules()	คืนชุดของอ็อบเจกต์ Rule ซึ่งเป็นกฎธุรกิจที่ระบุในไฟล์
GetAllCustomDtParameters()	คืนชุดของอ็อบเจกต์ CustomDtParameter ซึ่งเป็นตัวแปรปรับแต่งที่ระบุในไฟล์
GetAllReturnVariables()	คืนชุดของอ็อบเจกต์ ReturnVariable ซึ่งเป็นตัวแปรผลลัพธ์ระบุในไฟล์
GetAllGroups()	คืนชุดของอ็อบเจกต์ RuleGroup ซึ่งเป็นกลุ่มที่ระบุในไฟล์
GetAllLoopConditions()	คืนชุดของอ็อบเจกต์ LoopCondition ซึ่งเป็นเงื่อนไขการวนซ้ำที่ระบุในไฟล์

คลาส BusinessRulesDAM

เป็นคลาสที่ทำหน้าที่อ่าน ลบ บันทึก และแก้ไขข้อมูลในไฟล์กฎธุรกิจ รายละเอียดดังแสดงในตารางที่ ข-2

ตารางที่ ข-2 รายละเอียดของเมธอดของคลาส RuleSetDAM

เมธอด	รายละเอียด
InsertCategory()	เพิ่มหมวดหมู่ของเซตกฎธุรกิจที่กำหนด
InsertRuleSet()	เพิ่มเซตกฎธุรกิจที่กำหนด
DeleteCategory()	ลบหมวดหมู่ของเซตกฎธุรกิจที่กำหนด
DeleteRuleSet()	ลบเซตกฎธุรกิจที่กำหนด
UpdateCategory()	แก้ไขหมวดหมู่ของเซตกฎธุรกิจที่กำหนด
UpdateRuleSet()	แก้ไขเซตกฎธุรกิจที่กำหนด
GetAllCategories()	คืนชุดของอ็อบเจกต์ Category ซึ่งเป็นหมวดหมู่ที่ระบุในไฟล์
GetAllRuleSets()	คืนชุดของอ็อบเจกต์ RuleSet ซึ่งเป็นเซตกฎธุรกิจที่ระบุในไฟล์
GetAllPredefinedFacts()	คืนชุดของอ็อบเจกต์ PredefinedFacts ซึ่งเป็นไฟล์ข้อเท็จจริงที่ระบุในไฟล์

คลาส PredefinedFactsDAM

เป็นคลาสที่ทำหน้าที่อ่านข้อมูลในไฟล์ข้อเท็จจริง รายละเอียดดังแสดงในตารางที่ ข-3
ตารางที่ ข-3 รายละเอียดของเมธอดของคลาส PredefinedFactsDAM

เมธอด	รายละเอียด
GetAllConditions()	คืนชุดของอ็อบเจกต์ Condition ซึ่งเป็นแผ่นแบบเงื่อนไขที่ระบุในไฟล์
GetAllActions()	คืนชุดของอ็อบเจกต์ Action ซึ่งเป็นแผ่นแบบการกระทำที่ระบุในไฟล์

คลาส BpelDAM

เป็นคลาสที่ทำหน้าที่แปลงข้อมูลกฎธุรกิจเป็นภาษาบีเพล รายละเอียดดังแสดงในตารางที่
ข-4
ตารางที่ ข-4 รายละเอียดของเมธอดของคลาส BpelDAM

เมธอด	รายละเอียด
SetNestedIfElse()	สร้างชุดของข้อความ If-Then จากกฎธุรกิจ
SaveBpelDoc()	สร้างและบันทึกข้อมูลที่ได้จากการแปลงเป็นไฟล์บีเพล
SaveBpelWsdI()	สร้างและบันทึกไฟล์บรรยายเว็บเซอร์วิสของบีเพล

ภาคผนวก ค

เอกซ์เอ็มแอลสคีมาของไฟล์ข้อเท็จจริง ไฟล์เซตกฎธุรกิจ ไฟล์กฎธุรกิจ

เอกซ์เอ็มแอลสคีมาของไฟล์ข้อเท็จจริง

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="facts">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="conditionTemplates">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="conditionTemplate">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="naturalLanguage" type="xs:string" />
                    <xs:element name="customVariables">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element maxOccurs="unbounded" name="customVariable"
                            type="CustomVariableType" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
          <xs:attribute name="id" type="xs:string" use="required" />
          <xs:attribute name="name" type="xs:string" use="required" />
          <xs:attribute name="description" type="xs:string" use="required" />
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="actionTemplates">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="actionTemplate">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="naturalLanguage" type="xs:string" />
              <xs:element minOccurs="0" name="customVariables">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="unbounded"
                      name="customVariable" type="CustomVariableType" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element minOccurs="0" name="activities">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="variable"
                      type="VariableType" />
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="invoke"
                      type="InvokeType"/>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="return"
                      type="ReturnType" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

รูปที่ ค-1 เอกซ์เอ็มแอลสคีมาของไฟล์ข้อเท็จจริง

เอกซ์เอ็มแอลสคีมาของไฟล์เซตกฎธุรกิจ

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ruleset">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="groups" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="group">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:string" use="required" />
                  <xs:attribute name="name" type="xs:string" use="required" />
                  <xs:attribute name="description" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="loopConditions" minOccurs="0">
          <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
              <xs:group ref="OperatorGroup"/>
            </xs:sequence>
            <xs:attribute name="operation" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
        <xs:element name="rule">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="conditions">
                <xs:complexType>
                  <xs:choice>
                    <xs:element name="and" type="AndType"/>
                    <xs:element name="or" type="OrType"/>
                  </xs:choice>
                </xs:complexType>
              </xs:element>
              <xs:element name="actions">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="action" type="ActionType" maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
            <xs:attribute name="name" type="xs:string" use="required" />
            <xs:attribute name="order" type="xs:integer" use="required" />
          </xs:complexType>
        </xs:element>
        <xs:element name="returnVariables">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="returnVariable" minOccurs="0">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:string" use="required" />
                  <xs:attribute name="name" type="xs:string" use="required" />
                  <xs:attribute name="type" type="xs:string" use="required" />
                  <xs:attribute name="value" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

รูปที่ ค-2 เอกซ์เอ็มแอลสคีมาของไฟล์เซตกฎธุรกิจ

```

<xs:element name="customDtParamters">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="customDtParamter">
        <xs:complexType>
          <xs:attribute name="id" type="xs:string" use="required" />
          <xs:attribute name="name" type="xs:string" use="required" />
          <xs:attribute name="type" type="xs:string" use="required" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="required" />
<xs:attribute name="loop" type="xs:boolean" use="required" />
<xs:attribute name="loopCount" type="xs:unsignedByte" />
</xs:complexType>
</xs:element>

<xs:group name="OperatorGroup">
  <xs:choice>
    <xs:element name="between" type="BetweenType"/>
    <xs:element name="equals" type="CompareType"/>
    <xs:element name="notEquals" type="CompareType"/>
    <xs:element name="lessThan" type="CompareType"/>
    <xs:element name="lessThanEqualTo" type="CompareType"/>
    <xs:element name="moreThan" type="CompareType"/>
    <xs:element name="moreThanEqualTo" type="CompareType"/>
  </xs:choice>
</xs:group>

<xs:complexType name="CompareType">
  <xs:attribute name="leftId" type="xs:string" use="required"/>
  <xs:attribute name="rightId" type="xs:string" use="required"/>
  <xs:attribute name="variableId" type="xs:string" use="required" />
</xs:complexType>

<xs:complexType name="BetweenType">
  <xs:attribute name="leftId" type="xs:string" use="required"/>
  <xs:attribute name="rightId" type="xs:string" use="required"/>
  <xs:attribute name="variableId" type="xs:string" use="required" />
  <xs:attribute name="excludeLeft" type="xs:boolean" use="required" />
  <xs:attribute name="excludeRight" type="xs:boolean" use="required" />
</xs:complexType>

<xs:complexType name="ConditionType">
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:group ref="OperatorGroup"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required" />
  <xs:attribute name="order" type="xs:integer" use="required" />
  <xs:attribute name="conditionFactId" type="xs:string" use="required" />
</xs:complexType>

<xs:complexType name="OrType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="condition" type="ConditionType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AndType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="condition" type="ConditionType"/>
  </xs:sequence>
</xs:complexType>

```

รูปที่ ค-2 เอกซ์เอ็มแอลสคีมาของไฟล์เซตกฎธุรกิจ (ต่อ)


```

<xs:complexType name="SubtableType">
  <xs:sequence>
    <xs:element name="param" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="toElement" type="xs:string" use="required" />
        <xs:attribute name="dtParameterId" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="description" type="xs:string" use="required" />
  <xs:attribute name="rulesetId" type="xs:string" use="required" />
</xs:complexType>

<xs:complexType name="ActionType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="variableValue" minOccurs="1" maxOccurs="unbounded" >
        <xs:complexType>
          <xs:attribute name="id" type="xs:string" use="required" />
          <xs:attribute name="value" type="xs:string" use="required" />
        </xs:complexType>
      </xs:element>
      <xs:element name="subtable" type="SubtableType"/>
    </xs:choice>
    <xs:element minOccurs="0" name="assign" maxOccurs="unbounded" >
      <xs:complexType>
        <xs:attribute name="actionResultId" type="xs:string" use="required" />
        <xs:attribute name="returnVariableId" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

รูปที่ ค-2 เอกซ์เอ็มแอลสคีมาของไฟล์เซตกฎธุรกิจ (ต่อ)

เอกซ์เอ็มแอลสคีมาของไฟล์กฎธุรกิจ

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="businessRules">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="fact" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="id" type="xs:string" use="required" />
            <xs:attribute name="name" type="xs:string" use="required" />
            <xs:attribute name="location" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
        <xs:element name="category" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="id" type="xs:string" use="required" />
            <xs:attribute name="name" type="xs:string" use="required" />
            <xs:attribute name="description" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
        <xs:element maxOccurs="unbounded" name="rs">
          <xs:complexType>
            <xs:attribute name="id" type="xs:string" use="required" />
            <xs:attribute name="name" type="xs:string" use="required" />
            <xs:attribute name="location" type="xs:string" use="required" />
            <xs:attribute name="description" type="xs:string" use="required" />
            <xs:attribute name="categoryId" type="xs:string" use="required" />
            <xs:attribute name="factsId" type="xs:string" use="required" />
            <xs:attribute name="bpel" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

รูปที่ ค-3 เอกซ์เอ็มแอลสคีมาของไฟล์กฎธุรกิจ

ภาคผนวก ง

คู่มือการใช้งานเครื่องมือบรรณาธิกรณกฎธุรกิจ

รายละเอียดของการใช้เครื่องมือบรรณาธิกรณกฎธุรกิจมีดังต่อไปนี้

การแบ่งหมวดหมู่เซตกฎธุรกิจ

ก่อนที่ผู้ใช้จะสร้างตารางการตัดสินใจหรือเซตกฎธุรกิจนั้น ต้องสร้างหมวดหมู่ก่อนดังรูปที่ ง-1 โดยกำหนดชื่อและคำบรรยาย เมื่อสร้างหมวดหมู่แล้ว ผู้ใช้สามารถลบ แก้ไข หรือเปลี่ยนหมวดของเซตกฎธุรกิจภายหลังได้

Categories:	
Name	Description
Discount Calculation	Discount calculation for placing order
Insurance Policy	Policy about insurance and compensation for e
Place order process rulesets	Policies about placing order.

[Back to Categories page](#)

Create a new Category:

Name:

Description:

รูปที่ ง-18 ตัวอย่างการสร้างหมวดหมู่ของตารางการตัดสินใจ

การสร้างตารางการตัดสินใจ

เมื่อสร้างหมวดหมู่แล้ว ผู้ใช้สามารถเพิ่มตารางการตัดสินใจในหมวดหมู่นั้นได้ โดยเริ่มต้นต้องโดยกำหนดชื่อ คำบรรยายและไฟล์ข้อเท็จจริงที่ประกอบด้วยแผนแบบเงื่อนไขและการกระทำที่กฎธุรกิจภายในตารางต้องการใช้ ดังรูปที่ ง-2

Name	Description	Predefined F.
Minimum wage policy	Minimum wage per day depend on the province.	factForDiscountCa
Compensation fee calculation rules	Calculation the compensation fee depend on employees status and wage per month.	factForDiscountCa
Unworkable-day compensation fee calculation	Calculating the compensation fee from unworkable days of the employee	factForDiscountCa

Create a new Ruleset:

Name:

Description:

Predefined Facts:

รูปที่ ง-19 ตัวอย่างการสร้างตารางการตัดสินใจ

การสร้างกฎธุรกิจในตารางการตัดสินใจ

การสร้างกฎธุรกิจภายในตารางการตัดสินใจทำได้ 2 วิธี คือ สร้างกฎธุรกิจทีละกฎ หรือสร้างกฎธุรกิจจากเงื่อนไขที่ผู้ใช้เลือก เครื่องมือนี้จะสร้างกฎธุรกิจจากทุกกรณีที่เป็นไปได้ของค่าเริ่มต้นของเงื่อนไข จากรูปที่ ง-3 มีเงื่อนไขคือ ประเภทลูกค้ามีค่าเริ่มต้นที่ผู้ใช้สามารถกำหนดเองได้ คือ เป็นสมาชิกหรือไม่เป็นสมาชิก และเงื่อนไขเวลาในการส่งสินค้ามีค่าเริ่มต้น คือ 2 วันหรือมากกว่า 2 วัน

#	Name	Add Value	Delete	Generate Rules
1	Customer Type Condition		Delete	<input checked="" type="checkbox"/>
	Customer Type ?	Add		
	Member	Edit	Delete	
	Non Member	Edit	Delete	
2	Time of Arrival for Shipment Condition		Delete	<input checked="" type="checkbox"/>
	Time of Arrival ?	Add		
	equal to 2	Edit	Delete	
	more than 2	Edit	Delete	

รูปที่ ง-20 ตัวอย่างการสร้างตารางการตัดสินใจจากเงื่อนไข

ผู้ใช้สามารถเลือกที่จะให้นำเงื่อนไขมาสร้างกฎธุรกิจจากทุกกรณีของเงื่อนไขนั้น โดยเลือกที่ช่อง Generate Rules แต่ละเงื่อนไขที่เพิ่มเข้าไปผู้ใช้สามารถ เพิ่มค่าเริ่มต้นให้เงื่อนไขนั้นได้ เพื่อนำค่านั้นไปสร้างกฎธุรกิจ เมื่อสร้างตารางการตัดสินใจจากเงื่อนไขในรูป ง-4 จะได้กฎธุรกิจทั้งหมด 16 กฎ หรือ 2^N โดย N จำนวนค่าเริ่มต้นของเงื่อนไขทั้งหมด ในที่นี้ค่าเริ่มต้นของเงื่อนไขแรก มี 2 ค่า คือ เป็นสมาชิกหรือไม่เป็นสมาชิก และค่าเริ่มต้นของเงื่อนไขที่สองมี 2 ค่าเช่นกัน คือ 2 วันหรือมากกว่า 2 วัน ดังนั้นจึงมีจำนวนค่าเริ่มต้นของเงื่อนไขทั้งหมดเท่ากับ $2+2 = 4$ และมีกฎธุรกิจทั้งหมดที่เป็นไปได้คือ $2^4 = 16$ กฎ เมื่อสร้างตารางการตัดสินใจจะได้ดังรูปที่ ง-2

Name	Condition	
	The customer type is ?	Delivery within ? business days.
rule_1	1. [Member]	-
rule_2	1. [Non Member]	-
rule_3	-	1. [equal to 2]
rule_4	-	1. [more than 2]
rule_5	1. [Member] 2. [Non Member]	-
rule_6	1. [Member]	2. [equal to 2]
rule_7	1. [Member]	2. [more than 2]
rule_8	1. [Non Member]	2. [equal to 2]
rule_9	1. [Non Member]	2. [more than 2]
rule_10	-	1. [equal to 2] 2. [more than 2]
rule_11	1. [Member] 2. [Non Member]	3. [equal to 2]
rule_12	1. [Member] 2. [Non Member]	3. [more than 2]
rule_13	1. [Member]	2. [equal to 2] 3. [more than 2]
rule_14	1. [Non Member]	2. [equal to 2] 3. [more than 2]
rule_15	1. [Member] 2. [Non Member]	3. [equal to 2] 4. [more than 2]
rule_16 (Default)	-	-

รูปที่ ง-21 ตัวอย่างตารางการตัดสินใจที่สร้างจากเงื่อนไข

จากรูปที่ ง-4 กฎ rule_1 มีเงื่อนไขคือ ลูกค้าเป็นสมาชิก ส่วนกฎ rule_12 มีเงื่อนไขคือ ลูกค้าเป็นสมาชิกและไม่เป็นสมาชิกและเวลาในการส่งสินค้ามากกว่า 2 วัน จะเห็นว่ามีบางกฎที่มีเงื่อนไขที่เป็นไปไม่ได้ เช่น ลูกค้าเป็นสมาชิกและไม่เป็นสมาชิก ผู้ใช้สามารถลบและแก้ไขกฎเหล่านี้ภายหลังได้

ถ้าผู้ใช้เพิ่มเงื่อนไข โดยที่ไม่เลือกที่ช่อง Generate Rules เงื่อนไขนั้นจะไม่นำมาใช้สร้างกฎธุรกิจจากทุกกรณีของเงื่อนไข แต่เงื่อนไขนั้นจะถูกเพิ่มเข้าไปในกฎธุรกิจที่มีอยู่แล้ว เช่น ตัวอย่างจากรูปที่ ง-4 เงื่อนไขจะถูกเพิ่มต่อท้ายเข้าไปในกฎทุกกฎ แต่ไม่มีการสร้างกฎใหม่แต่อย่างใด

การแก้ไขกฎธุรกิจ

ผู้ใช้งานสามารถสร้าง แก้ไขและเปลี่ยนลำดับกฎธุรกิจที่ละกฎได้ โดยมีเว็บแอปพลิเคชันเป็นส่วนต่อประสานกับผู้ใช้งาน (User interface) ดังรูปที่ ง-5 จากรูปเป็นกฎธุรกิจกฎหนึ่งซึ่งประกอบด้วย ส่วน If และ Then ในส่วน If ประกอบด้วยเงื่อนไข ในส่วน Then ประกอบด้วยการกระทำ

Rule: rule_1		Group: Grp 1		Delete rule
If Operation: And Change operation				
Condition		Edit	Delete	Change Order
1. Order size is less than 22		Edit	Delete	Up Down
2. The total price is more than 12		Edit	Delete	Up Down
Add Condition				
Then				
Action		Edit	Delete	Change Order
1. Allow 11% discount		Edit	Delete	Up Down
2. Get cost of item id 44		Edit	Delete	Up Down
Add Action				
Edit: Name	rule_1	<input type="checkbox"/> Default rule	Group	Grp 1 Ok

รูปที่ ง-22 การสร้างและแก้ไขกฎธุรกิจที่ละกฎ

การกำหนดเงื่อนไข

ผู้ใช้งานสามารถเพิ่ม ลบ กำหนดลำดับและแก้ไขเงื่อนไข รวมทั้งเลือกตัวดำเนินการตรรกะ (Logical operator) ที่ดำเนินการในเงื่อนไขได้ ตัวดำเนินการตรรกะ ได้แก่ แอนด์ (and) ออร์ (or) เงื่อนไขในกฎธุรกิจของงานวิจัยนี้สามารถเลือกตัวดำเนินการตรรกะแอนด์หรือออร์อันใดอันหนึ่งเท่านั้น นั่นคือทุกเงื่อนไขในกฎหนึ่งจะแอนด์กันหรือออร์กันอย่างใดอย่างหนึ่งเท่านั้น

เมื่อผู้ใช้งานเพิ่มหรือแก้ไขเงื่อนไขในกฎธุรกิจ ผู้ใช้งานต้องระบุตัวดำเนินการเปรียบเทียบและค่าของตัวแปรปรับแต่งของเงื่อนไขนั้น ดังรูปที่ ง-6

If Operation: And Change operation		Edit
Condition		
Order size is	less than 22	Ok Cancel
1. The total price is more than 12		Edit
Add Condition		

รูปที่ ง-23 ระบุตัวดำเนินการเปรียบเทียบและค่าของตัวแปรที่เกี่ยวกับเงื่อนไข

ลักษณะตัวดำเนินการเปรียบเทียบ และค่าของตัวแปรขึ้นอยู่กับประเภทของตัวแปร (Variable type) ดังนี้

1) ตัวแปรประเภทตัวเลข ตัวดำเนินการเปรียบเทียบที่สามารถเลือกได้ ได้แก่ “น้อยกว่า” “น้อยกว่าเท่ากับ” “มากกว่า” “มากกว่าเท่ากับ” “เท่ากับ” “ไม่เท่ากับ” และ “ระหว่าง” ส่วนค่านั้น ถ้าต้องเป็นตัวเลข ถ้าเป็นตัวแปรประเภท integer ต้องใส่ตัวเลขจำนวนเต็ม ถ้าตัวแปรประเภท double สามารถจำนวนทศนิยมได้

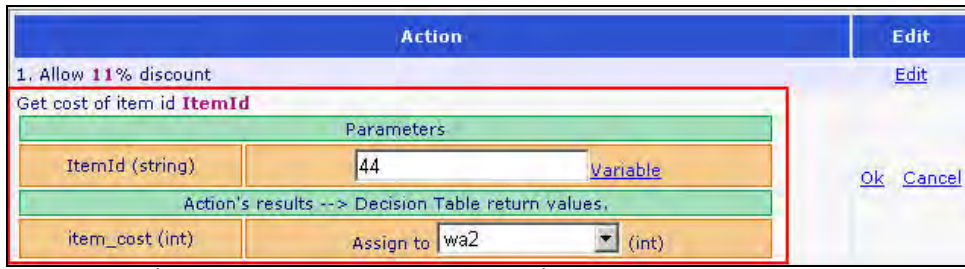
2) ตัวแปรประเภทสายอักขระ ตัวดำเนินการเปรียบเทียบที่สามารถเลือกได้ ได้แก่ “เท่ากับ” และ “ไม่เท่ากับ” ส่วนค่านั้นเป็นตัวอักษร ตัวเลข หรือสายอักขระใดๆ

3) ตัวแปรประเภทบูลีน ไม่มีตัวดำเนินการเปรียบเทียบให้เลือก เพราะเป็นค่า “เท่ากับ” เสมอ ส่วนค่าเป็น “จริง” หรือ “เท็จ”

จะเห็นได้ว่าเงื่อนไขประกอบด้วยตัวแปรปรับแต่งที่ผู้ใช้ต้องกำหนดค่า เมื่อตารางการตัดสินใจถูกเรียกใช้เงื่อนไขเหล่านี้จะถูกประเมิน (Evaluate) ว่าจริงหรือเท็จ ดังนั้นตารางการตัดสินใจต้องมีการรับพารามิเตอร์จากผู้เรียกใช้ เพื่อนำพารามิเตอร์นั้นไปประเมินกับเงื่อนไขของกฎธุรกิจต่างในตารางการตัดสินใจ เช่น เงื่อนไขเกี่ยวกับจำนวนสินค้าที่สั่ง มีตัวแปรปรับแต่ง คือ OrderSize และผู้ใช้กำหนดเงื่อนไขว่า “OrderSize > 20” ดังนั้นผู้เรียกใช้ต้องส่งค่าของ OrderSize มาให้ตารางการตัดสินใจ เพื่อใช้ในการประเมินเงื่อนไข นั่นคือทุกตัวแปรปรับแต่งของทุกเงื่อนไขที่มีในตารางการตัดสินใจ จะต้องเป็นพารามิเตอร์ของตารางการตัดสินใจนั้น เมื่อตารางการตัดสินใจทำงานในรูปแบบเว็บเซอวิส พารามิเตอร์เหล่านี้จะกลายเป็นพารามิเตอร์ของเว็บเซอวิสนั่นเอง รายละเอียดเกี่ยวกับพารามิเตอร์ของตารางการตัดสินใจจะกล่าวในหัวข้อการกำหนดพารามิเตอร์ของตารางการตัดสินใจ

การกำหนดการกระทำทั่วไป

การกระทำทั่วไป คือ การกระทำที่ไม่ใช่การเรียกตารางย่อย รูปแบบการกระทำหรือกิจกรรมมีการระบุชัดเจนในไฟล์ข้อเท็จจริง ผู้ใช้สามารถเพิ่ม แก้ไข กำหนดลำดับและลบการกระทำ การกำหนดการกระทำมี 2 ส่วน ได้แก่ การกำหนดค่าพารามิเตอร์ และการจัดการผลลัพธ์ ดังรูปที่ ง-7



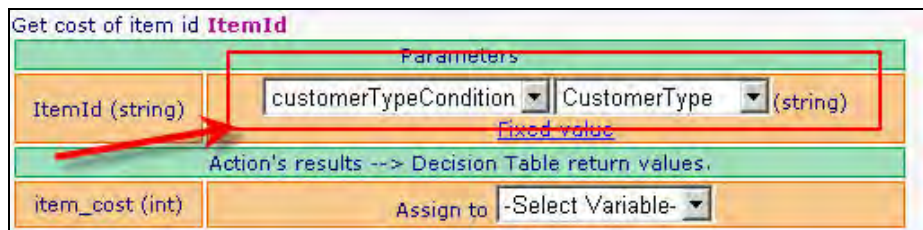
รูปที่ ง-24 การแก้ไขการกระทำด้วยเครื่องมือบรรณาธิกรณกฎธุรกิจ

1) การกำหนดค่าพารามิเตอร์ เป็นการกำหนดค่าตัวแปรปรับแต่ง (Custom variable) ของการกระทำซึ่งจะเป็นพารามิเตอร์ที่ส่งให้การกระทำนั้น ค่ากำหนดให้มีดังนี้

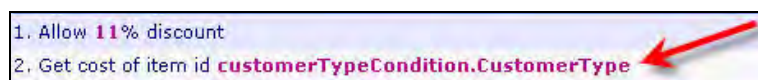
ก) ค่าคงที่ (Fixed value) เช่น 1 “ABC” true เป็นต้น

ข) ค่าจากพารามิเตอร์ของตารางการตัดสินใจ ตามที่กล่าวในส่วนการกำหนดเงื่อนไขว่าตารางการตัดสินใจมีพารามิเตอร์ที่เกิดตัวแปรปรับแต่งในแผ่นแบบเงื่อนไข และพารามิเตอร์ที่ผู้ใช้สร้างขึ้นเอง ผู้ใช้สามารถนำพารามิเตอร์เหล่านั้นมาเป็นพารามิเตอร์ของการกระทำได้ ดังรูปที่ ง-8 กำหนดค่าพารามิเตอร์ของการกระทำ ItemId จากพารามิเตอร์ CustomerType ซึ่งมาจากตัวแปรปรับแต่งของเงื่อนไข customerTypeCondition ประเภทข้อมูลของพารามิเตอร์ของตารางการตัดสินใจที่เลือกมานั้นต้องสอดคล้องกับพารามิเตอร์ของการกระทำที่กำหนด เมื่อกำหนดค่าแล้วค่าพารามิเตอร์จะแสดงในรูปแบบ “<ชื่อเงื่อนไข>.<ชื่อตัวแปรปรับแต่ง>” เช่น “customerTypeCondition.CustomerType” ดังรูปที่ ง-9

ค) ค่าของตัวแปรผลลัพธ์ของตารางการตัดสินใจ ซึ่งเป็นตัวแปรที่เก็บค่าผลลัพธ์ของตารางการตัดสินใจซึ่งจะคืนกลับไปยังผู้เรียกใช้ ค่าของตัวแปรผลลัพธ์โดยปกติถูกกำหนดจากผลลัพธ์ของการกระทำ การจัดการผลลัพธ์ของการกระทำกล่าวในหัวข้อถัดไป ผู้ใช้สามารถนำค่าของตัวแปรผลลัพธ์นี้เป็นพารามิเตอร์ของการกระทำได้

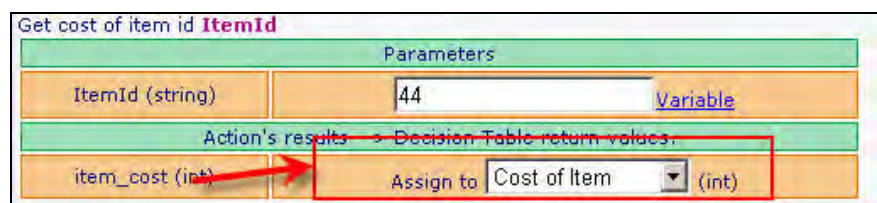


รูปที่ ง-25 การกำหนดค่าพารามิเตอร์ของการกระทำด้วยพารามิเตอร์ของตารางการตัดสินใจ



รูปที่ ง-26 การแสดงค่าพารามิเตอร์ของการกระทำ

2) การจัดการผลลัพธ์ เป็นการนำค่าผลลัพธ์ที่ได้จากการกระทำมากำหนดให้กับตัวแปรผลลัพธ์ของตารางการตัดสินใจซึ่งจะถูกส่งกลับไปยังผู้เรียกใช้ตารางการตัดสินใจ หรือผู้เรียกใช้เว็บเซอวิสในกรณีที่ตารางการตัดสินใจทำงานในรูปแบบของเว็บเซอวิสแล้ว ดังรูปที่ ง-10 ผลลัพธ์ของการกระทำ item_cost ถูกกำหนดให้แก่ตัวแปรผลลัพธ์ของตารางการตัดสินใจ Cost of Item รายละเอียดของตัวแปรผลลัพธ์ของตารางการตัดสินใจจะกล่าวในหัวข้อการกำหนดตัวแปรผลลัพธ์ของตารางการตัดสินใจ



รูปที่ ง-27 การนำผลลัพธ์จากการกระทำมากำหนดค่าให้ผลลัพธ์ของตารางการตัดสินใจ

การกำหนดการกระทำที่เรียกตารางย่อย

เมื่อแผนแบบการกระทำที่เป็นการเรียกตารางย่อยเพิ่มเข้าสู่กฎธุรกิจ ผู้ใช้ต้องระบุตารางย่อยที่ต้องการเรียก (ซึ่งก็คือตารางการตัดสินใจอื่นที่มีอยู่ก่อนแล้วในระบบ) และคำบรรยาย ดังรูปที่ ง-11 จากนั้นต้องกำหนดค่าพารามิเตอร์และจัดการผลลัพธ์ของตารางย่อยที่เรียกนั้น



รูปที่ ง-28 การเพิ่มการกระทำแบบเรียกตารางย่อยเข้าสู่กฎธุรกิจ

1) กำหนดค่าพารามิเตอร์ของตารางย่อย คือการกำหนดค่าพารามิเตอร์ที่จะส่งให้ตารางย่อย ตารางย่อยก็คือตารางการตัดสินใจอันหนึ่งซึ่งต้องมีการรับพารามิเตอร์เพื่อนำค่าไปประเมินเงื่อนไขของกฎธุรกิจ รายละเอียดเกี่ยวกับพารามิเตอร์ของตารางย่อยหรือตารางการตัดสินใจ จะอธิบายในหัวข้อตารางการตัดสินใจ

การกำหนดค่าพารามิเตอร์ของตารางย่อยคล้ายกับการกำหนดค่าพารามิเตอร์ของการกระทำทั่วไปที่ได้กล่าวก่อนหน้า สามารถกำหนดค่าเป็นค่าคงที่ หรือค่าพารามิเตอร์ของตารางการตัดสินใจ ดังรูปที่ ง-12

Action	
1. call discount book rule subtable [Edit parameters] [Set Result]	
orderSizeCondition	Checking the total order sizes
OrderSize (int)	orderSizeCondition OrderSize (int) Fixed value
totalPriceCondition	Checking the total price
TotalPrice (double)	2000.00 Variable
customerTypeCondition	Checking the type of customer
CustomerType (string)	bookstore Variable
Submit Cancel	

รูปที่ ง-29 การกำหนดค่าพารามิเตอร์ของการกระทำด้วยพารามิเตอร์ของตาราง

2) การจัดการผลลัพธ์ของตารางย่อย คือ การนำค่าผลลัพธ์จากตารางย่อยมา กำหนดให้กับผลลัพธ์ของตารางการตัดสินใจ คล้ายกับการจัดการผลลัพธ์ของการกระทำทั่วไปที่ได้กล่าวมาแล้ว ถ้าไม่ต้องการผลลัพธ์ของตารางย่อย ก็ไม่จำเป็นต้องเลือกตัวแปรผลลัพธ์ของตารางการตัดสินใจแต่อย่างใด ดังรูปที่ ง-13 ซึ่งผลลัพธ์ discount ไม่ได้กำหนดให้ตัวแปรใด

Action	
1. call discount book rule subtable [Edit parameters] [Set Result]	
Action's results --> Decision Table return values.	
discount (double)	Assign to -Select Variable-
total_price (double)	Assign to total_book_price (double)
Submit Cancel	

รูปที่ ง-30 การนำผลลัพธ์จากตารางย่อยมากำหนดค่าให้ผลลัพธ์ของตารางการตัดสินใจ

การกำหนดพารามิเตอร์ของตารางการตัดสินใจ

พารามิเตอร์ของตารางการตัดสินใจ คือ ข้อมูลที่ผู้เรียกใช้ตารางการตัดสินใจส่งมาให้ตาราง เพื่อใช้ในการประเมินเงื่อนไขของกฎธุรกิจในตารางนั้น เช่น เงื่อนไข "OrderSize > 3" ผู้เรียกต้อง ส่งค่าพารามิเตอร์ของ OrderSize เพื่อให้กฎธุรกิจในตารางการตัดสินใจสามารถประเมินเงื่อนไขได้ เมื่อตารางการตัดสินใจทำงานในรูปเว็บเซอวิซ พารามิเตอร์เหล่านี้จะกลายเป็นพารามิเตอร์ของเว็บเซอวิซ พารามิเตอร์ของตารางการตัดสินใจมี 2 ประเภท ดังนี้

1) พารามิเตอร์จากตัวแปรปรับแต่งของเงื่อนไข พารามิเตอร์ประเภทนี้มาจากทุกตัวแปรปรับแต่ง ของทุกแผ่นแบบเงื่อนไขที่อยู่ในกฎธุรกิจทุกกฎของตารางการตัดสินใจ ขออธิบายดังนี้

DT = ตารางการตัดสินใจ

B = กฎธุรกิจทั้งหมดใน DT

C = แผ่นแบบเงื่อนไขทั้งหมดที่ใช้ใน B

พารามิเตอร์ของตารางการตัดสินใจ คือ เซตของตัวแปรปรับแต่งทั้งหมดของทุกแผ่นแบบเงื่อนไขที่ไม่ซ้ำกันใน C

C เป็นแผ่นแบบเงื่อนไข เพราะว่า อาจมีกฎธุรกิจหลายกฎใช้แผ่นแบบเงื่อนไขเดียวกัน เช่น กฎธุรกิจ B1 กำหนดเงื่อนไข “OrderSize > 3” และกฎธุรกิจ B2 กำหนดเงื่อนไข “OrderSize <= 3” เป็นต้น ในกรณีนี้ตารางการตัดสินใจต้องการเพียงพารามิเตอร์ OrderSize ตัวเดียว

2) พารามิเตอร์ที่ผู้ใช้สร้างขึ้นเอง เป็นพารามิเตอร์ที่ไม่ได้ใช้ในเงื่อนไขเหมือนพารามิเตอร์แบบแรก แต่อาจใช้เพื่อกำหนดค่าให้กับตัวแปรปรับแต่งของการกระทำ เช่น มีเงื่อนไข “OrderSize > 3” เมื่อเงื่อนไขเป็นจริงทำการกระทำของสินค้าซึ่งต้องใช้ชื่อลูกค้าด้วย แต่ว่าชื่อลูกค้าไม่ถูกใช้ในเงื่อนไขดังนั้น ผู้ใช้ต้องสร้างพารามิเตอร์ชื่อลูกค้าขึ้นมาเอง

การกำหนดตัวแปรผลลัพธ์ของตารางการตัดสินใจ

ตัวแปรผลลัพธ์ของตารางการตัดสินใจ คือ ตัวแปรที่เก็บค่าผลลัพธ์ของตารางการตัดสินใจ ซึ่งจะคืนกลับไปยังผู้เรียกใช้ เมื่อตารางการตัดสินใจทำงานในรูปแบบเซอวิวิส ตัวแปรผลลัพธ์นี้จะเก็บค่าผลลัพธ์ของเว็บเซอวิวิสนั้นเอง ตัวแปรผลลัพธ์ของตารางการตัดสินใจสร้างและกำหนดค่าโดยผู้ใช้ การสร้างคล้ายการประกาศตัวแปรที่ต้องระบุชื่อและประเภทข้อมูล ส่วนการกำหนดค่าทำได้ 2 วิธี ดังนี้

- 1) ระบุเป็นค่าคงที่ เช่น 56 “ABC” true เป็นต้น
- 2) กำหนดจากค่าผลลัพธ์ของการกระทำที่ถูกเรียกในกฎธุรกิจ ของตารางการตัดสินใจ นั่นคือ เมื่อการกระทำในกฎธุรกิจทำงานเสร็จ และส่งค่าผลลัพธ์กลับมา ผู้ใช้สามารถนำค่าที่วานี้เก็บในตัวแปรผลลัพธ์ ดังที่ได้กล่าวไว้ในส่วนการกำหนดค่าการกระทำทั่วไป

การกำหนดการทำงานแบบวนซ้ำ

ตารางการตัดสินใจสามารถทำงานแบบวนซ้ำได้ หมายถึง ตารางการตัดสินใจถูกประเมินไปเรื่อยๆ トラバドที่เงื่อนไขของการวนซ้ำเป็นจริง เมื่อสิ้นสุดการวนซ้ำตารางการตัดสินใจจะคืนผลลัพธ์สุดท้ายกลับไปยังผู้เรียกใช้ จำนวนรอบการวนซ้ำมี 2 แบบ คือ

- 1) ระบุจำนวนรอบแบบตายตัว คือ ระบุชัดเจนว่าจะวนซ้ำกี่รอบ เช่น 2 รอบ 5 รอบ เป็นต้น
- 2) กำหนดด้วยเงื่อนไขโดยใช้ตัวแปรผลลัพธ์ของตารางการตัดสินใจ เช่น ตารางการตัดสินใจมีตัวแปรผลลัพธ์ คือ rv_1 และ rv_2 เป็นตัวแปรประเภทจำนวนเต็ม ผู้ใช้สามารถ

กำหนดเงื่อนไขเป็น $rv_1 \leq 0$ and $rv_2 \leq 100$ ได้ ดังรูปที่ ง.14 อย่างไรก็ตามเงื่อนไขทั้งหมดต้องแอนด์หรือออร์กันอย่างใดอย่างหนึ่งเท่านั้น

Enable loop for this ruleset
 Fix loop count times
 Use condition

Operation: And [Change operation](#)

Condition	Edit	Delete	Change Order
1. rv_1 is less than or equal to 0	Edit	Delete	Up Down
2. rv_2 is less than or equal to 100	Edit	Delete	Up Down
3. ccc is equal to yyy	Edit	Delete	Up Down

รูปที่ ง-31 การกำหนดเงื่อนไขการวนซ้ำของตารางการตัดสินค้า

การตรวจสอบความกำกวมของตารางการตัดสินค้า

ผู้ใช้งานสามารถตรวจสอบความกำกวมของตารางการตัดสินค้าได้ โดยตรวจสอบว่าเงื่อนไขในกฎธุรกิจที่เหลื่อมกันหรือไม่ ดังตัวอย่างรูป ง-15 จะเห็นว่าทั้ง 2 เงื่อนไขของทั้ง 2 กฎมีช่วงที่เหลื่อมกัน ดังนั้นตารางการตัดสินค้านี้จึงกำกวม

	Condition	
Name	Order size is ?	The total price is ? baht
rule1	1. [more than or equal to 4]	2. [more than 50]
rule2	1. [more than 43]	2. [more than or equal to 60]

รูปที่ ง-32 ตัวอย่างการตรวจสอบความกำกวมของตารางการตัดสินค้า

การตรวจสอบความขัดแย้งกันของเงื่อนไขในกฎธุรกิจ

เนื่องจากในกฎธุรกิจหนึ่งสามารถมีการใช้แผ่นแบบเงื่อนไขซ้ำกันได้ เช่น “Order Size > 2 AND Order Size <= 2” เป็นต้น เครื่องมือบรรณาธิกรณกฎธุรกิจสามารถตรวจสอบได้ว่าเงื่อนไขที่มาจากแผ่นแบบเงื่อนไขเดียวกันมีความขัดแย้งหรือไม่ ดังรูปที่ ง-16 จะเห็นว่ากำหนดเงื่อนไขเป็น “Order Size < 7 AND Order Size > 7” ซึ่งไม่สามารถเกิดขึ้น นั่นคือถ้าเงื่อนไขที่มาจากแผ่นแบบเงื่อนไขเดียวกันและอยู่ในกฎธุรกิจเดียวกัน จะขัดแย้งกันเมื่อไม่มีช่วงเหลื่อมกันใน

กรณีที่ตัวแปรปรับแต่งเป็นตัวเลข ถ้าเป็นสายอักขระหรือบูลีนจะขัดแย้งกันในกรณีค่าไม่เท่ากัน
อย่างไรก็ตามความขัดแย้งกันของเงื่อนไขเกิดเฉพาะกรณีที่เงื่อนไขทั้งหมดแอนด์กันเท่านั้น

Rule: rule_1 Group: No Group Delete rule

If Operation: And Change operation

The conditions [Order Size Condition] are conflicted. Please edit the condition!!

Condition

1. The total price is more than 6 baht
2. Order size is less than 7
3. Order size is more than 7

รูปที่ ง-33 ตัวอย่างการตรวจสอบความขัดแย้งกันของเงื่อนไขในกฎธุรกิจ

การแยกตารางการตัดสินใจ

ในกรณีที่กฎธุรกิจในตารางการตัดสินใจเพิ่มจำนวนมากขึ้น และอาจมีบางกฎที่ไม่สอดคล้องหรือสัมพันธ์กับกฎอื่น ผู้ใช้สามารถแบ่งกฎธุรกิจเหล่านี้ ไปเป็นอีกตารางการตัดสินใจหนึ่งได้ ดังรูป ง-17 ผู้ใช้เลือกกฎที่ต้องการแยกจากรายการตัดสินใจ แยกกลุ่ม Split Table จากนั้นตั้งชื่อตารางใหม่ กฎธุรกิจที่ถูกเลือกจะถูกแยกออกอยู่ในตารางใหม่นั้น

Action	Group	Order	Select rule
Minimum wage per day is ? baht	The eastern part	up down	<input type="checkbox"/>
	The eastern part	up down	<input type="checkbox"/>
	The eastern part	up down	<input type="checkbox"/>
	The eastern part	up down	<input type="checkbox"/>
	The western part	up down	<input checked="" type="checkbox"/>
	The western part	up down	<input checked="" type="checkbox"/>
	The western part	up down	<input checked="" type="checkbox"/>
	The western part	up down	<input checked="" type="checkbox"/>
	The western part	up down	<input checked="" type="checkbox"/>

Deleted Selected Rules Split Table

รูปที่ ง-34 ตัวอย่างการแยกตารางการตัดสินใจ

ภาคผนวก จ

บทความที่นำเสนอในงานประชุมวิชาการ

1. บทความเรื่อง “A Development of Business Rules with Decision Tables for Business Processes” นำเสนอในงานประชุมวิชาการ The International Conference of IEEE TENCON 2007 จัดที่ Taipei International Convention Center เมืองไทเป ประเทศไต้หวัน ระหว่างวันที่ 30 ตุลาคม – 2 พฤศจิกายน พ.ศ. 2550
2. บทความเรื่อง “Generating BPEL Scripts for Business Rules from a Decision Table and Domain-Specific Language” นำเสนอในงานประชุมวิชาการ The 5th International Joint Conference on Computer Science and Software Engineering (JCSSE 2008) จัดที่โรงแรม Felix River Kwai จังหวัดกาญจนบุรี ระหว่างวันที่ 7 – 9 พฤษภาคม พ.ศ. 2551

A Development of Business Rules with Decision Tables for Business Processes

Thanawut Auechaikul and Wiwat Vatanawood
 Department of Computer Engineering, Chulalongkorn University
 Bangkok, Thailand
 Thanawut.A@Student.chula.ac.th, wiwat@chula.ac.th

Abstract-At present, business rules and business processes are increasingly significant for organizations. Business processes enable organization to quickly respond to changes and align with business objectives. Business rules enable organization to decouple business logic from the applications. However, the people who truly know the business rules like business managers essentially implement the business rules through software developers. This may cause the business rules used in IT systems not align with organization goals and lack of agility to change. This paper presents a method that enables the business managers to create and edit the business rules themselves by constructing decision tables as a guideline and exposing them as web services in BPEL (Business Process Execution Language) to support the expected business processes.

I. INTRODUCTION

Business rules are widespread in any organization for example discount policy, budget approval and insurance approval. These business rules are considered as the core of business processes as they are used in decision-making, defining policy, responding to events and being constraint of organization [1]. In short, business rules synchronize activities occurring in business processes of organizations. However, the business rules are traditionally embedded within the applications implemented in programming languages and consequently difficult for ordinary users who truly know them to understand. It is also difficult to maintain and verify that the implementation matches the requirements [7]. Therefore, we should specially manage and define the business rules explicitly.

Web services provide interoperability solutions for sharing data and functionality across applications. Web services can be composed as to perform a new business process using orchestration concept. BPEL (Business Process Execution Language) [3] and our previous work [6] are languages that define the business processes by specifying the control and data flows among the web services. BPEL and web services provide agility and interoperability solutions for sharing data and functionality across heterogeneous applications.

This paper presents a method to bridge the gap between the business rule experts like business managers and the actual business rules implemented in IT systems by using decision tables. Typically, business managers lack of programming knowledge background but are familiar with decision tables. Therefore, our work selects decision tables to represent business rules and transforms them into BPEL. The BPEL business rules are exposed as web services to

support business processes. This approach enables business managers to define and directly implement the business rules aligning with the business objectives. Also, the decision tables enable business managers to easily create and modify the business rules without programming. Additionally, these business rules separated from business logic can collaborate with other applications by invoking other web services or can be invoked from other systems to compose agile business processes. The mentioned approach enables business rules experts in organizations can implement business rules by themselves and thus decreasing time consumed to modify and create business rules and increasing reusability, maintainability and verifiability of them.

II. BUSINESS RULES AND BPEL

A. Business Rules

A business rule is a statement that defines or constraints some aspects of the business [4]. It can be a policy, agreement or definition relating to criteria organizations do with the customers, partners or employees. The business rules typically consist of a collection of terms, facts and rules. A term is a basic word or word phrase in any natural language e.g. English that workers can recognize and share in the business [20]. A fact is a statement that connects terms through prepositions and verb phrases into sensible business-relevant observations. A rule is a statement that is applied in decision-making or computation to produce new information [1]. All business rules can be expressed by using IF-THEN format.

Business rules are particularly good for decisions and policy-intensive business domains such as the finance and insurance sectors [5]. Some of these rules exist in a formalized way, e.g., in an organizational handbook, others are not documented and exist only informally [9]. Business rules capture the unique competitive proposition of a business. Because of this central role, the specification, enforcement, and management of rules should be a separated and specialized portion of business logic [11].

B. BPEL

Process composition languages e.g. BPEL [3] and [6] provide collaboration among web services and define business process flows. However, developers are increasingly using BPEL based on XML for modeling business processes within the Web service architecture [8]. It supports two different kinds of processes i.e. executable business processes that can be executed by a BPEL engine and abstract business processes that are not intended to be

executed and specify some agreed behavior that partners in a communication scenario can agree upon [10].

There are many researches relating to combining web services and business rules to support business processes [16]. Since business rules management separates the maintenance and execution of business rules, leveraging web services technology to expose business rules as callable web services is the next approach to system integration [12]. This approach increases reusability and loose coupling of the business rules in IT systems.

III. DECISION TABLES

A decision table [13] is a structure for describing a set of related decision rules. The basic parts of a decision table are shown in Table 1. The upper left portion called the condition stub contains statements of the conditions. The lower left portion called the action stub contains statements of the actions. The condition entries and action entries are in the upper right and lower right portions, respectively. Each column in the entry portions (condition and action) forms a decision rule. From Table 1, Y means condition is true, N means condition is false, X means action is performed and – means irrelevant condition (condition does not matter) or action doesn't performed.

TABLE I
THE BASICS PARTS OF A DECISION TABLE

	Business Rules		
	1	2	3
Condition Stub	Condition Entries		
Condition 1	Y	Y	N
Condition 2	Y	N	-
Action Stub	Action Entries		
Action 1	X	-	X
Action 2	-	X	X

IV. RULE MARKUP LANGUAGE (RULEML)

RuleML is a markup language for publishing and sharing rule bases on the World Wide Web [15]. RuleML that is open and neutral language represents rules in the form XML. There are many works relating to RuleML [15]. For example, [16] uses RuleML to store rules from the Rule Editor application. Reference [17] proposes XRML to represent business rules by embedding the rules in web pages in order to control software agent and web page display. Furthermore, [18] and [19] propose General Rule Markup Language and Mathematical Markup Language, respectively.

V. APPROACH

This paper presents an approach to define business rules in IT system with decision tables in order to users who truly know the business can create business rules aligning the organizations' goal. The decision tables based on predefined conditions and actions are stored in RuleML format to persist the created business rules. Then the business rules are transformed into BPEL collaborating web services in order to create and support business processes. The model is shown following Fig. 1.

This approach consists of four parts:

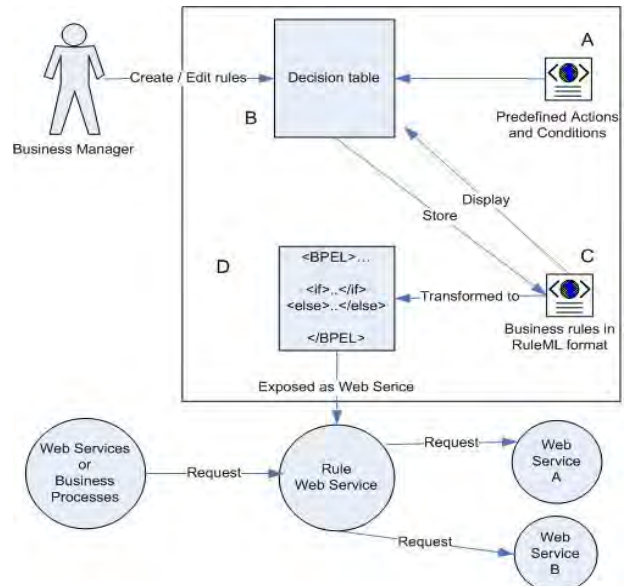


Figure 1. Creation business rules from decision tables approach

A. Predefined Actions and Conditions

This part kept in a XML file defines terms or variables used in actions and conditions, variable data types and activity types in actions. Activity types can be to return a value to client or web service calls. If the activity is to return a value, the returned variable or value has to be defined. If the activity is a web service call, necessary artifacts e.g. the WSDL location of the called web service and a list of web service parameters are required. Furthermore, it includes natural language sentences representing condition and actions in decision tables. These sentences enable general users can easily understand terms in business rules. For example, if a integer variable in condition is accidentNo, the sentence presented in decision table is "Number of accident more than?". The "?" will be replace with a user-defined value when create decision tables.

B. Decision Table

A decision table represents business rules to users. Users like business managers can create business rules based on predefined actions and conditions as mentioned above. After business rules are created and stored in RuleML files, users can edit them via decision tables again.

C. Business Rules in RuleML Format

After business rules are created in a decision table, they are stored in a RuleML file to persist them. Each of the business rules in the RuleML file includes conditions and actions customized by users, variables used in conditions and actions, variable data types and activity types in actions according to previous predefined actions and conditions. From the previous example, if a user defines "Number of accident more than 12", this condition is transformed to pseudo code like "AccidentNo > 12" in a RuleML file.

D. BPEL files

BPEL files contain the business rules in form *if conditions then actions* implemented by BPEL. These files can be executed by BPEL engines to expose the business rules as web services that can

invoke other web services to support business processes. The approach used to transform business rules in a decision table to business rules in a BPEL file is explained as following:

If there is a decision table DT, the other variables are defined as following:

- $C = \{C_i\}$ ($i = 1..cn$) is the set of conditions in the condition stub of DT;
- $A = \{A_i\}$ ($i = 1..an$) is the set of actions in the action stub of DT;
- $R = \{R_i\}$ ($i = 1..m$) is the set of business rules in DT;
- $CTR_j = \{CT_iR_j\}$ ($i = 1..tcn$) is the set of true conditions of the business rule R_j and R_j is a member of R ;
- $CFR_j = \{CF_iR_j\}$ ($i = 1..fcn$) is the set of false conditions of the business rule R_j and R_j is a member of R ;
- $AR_j = \{A_iR_j\}$ ($i = 1..pan$) is the set of performed action of the business rule R_j and R_j is a member of R ;

With above definitions, the pseudo code of BPEL into which the decision table DT is transformed is shown as following:

```

IF (CT1R1 AND CT2R1 AND ... AND CTtcnR1 AND
NOT (CF1R1 OR CF2R1 OR ... OR CFfcnR1))
THEN A1R1 AND A2R1 AND ... AND ApanR1
ELSE IF (CT1R2 AND CT2R2 AND ... AND CTtcnR2 AND
NOT (CF1R2 OR CF2R2 OR ... OR CFfcnR2))
THEN A1R2 AND A2R2 AND ... AND ApanR2
...
ELSE IF (CT1Rm AND CT2Rm AND ... AND CTtcnRm AND
NOT (CF1Rm OR CF2Rm OR ... OR CFfcnRm))
THEN A1Rm AND A2Rm AND ... AND ApanRm
    
```

Each IF and ELSE IF statement in BPEL represents each business rule in a decision table that stored in a RuleML file. The conditions in each IF and ELSE IF statement are true conditions and false conditions (preceded by NOT operator) of that business rules. The actions in THEN statements are the actions performed when all true conditions in that business rule are true and all false conditions are false.

VI. CASE STUDY

Our case study is a discount policy decision table of a bookstore as shown in Table 2:

TABLE 2
THE DISCOUNT POLICY DECISION TABLE

	1	2	3
Condition Stub	Condition Entries		
Customer is a student	Y	Y	N
Order size is more than 7	Y	N	-
Action Stub	Action Entries		
Allow 15% discount	X	-	-
Allow 10% discount	-	X	-
No discount	-	-	X

From the table, there are obviously two variables in the condition part i.e. customer type and order size whereas there is only one variable in the action part i.e. discount. Consequently, the predefined actions and conditions of the decision table are shown as following:

```

<facts>
  <conditions>
    <condition name="customerType">
      <naturalLanguage>Customer is ?</naturalLanguage>
      <variable name="CustomerType" type="string" />
    </condition>
    ...
  </conditions>
  <actions>
    <action name="allowDiscount" type="return">
      <naturalLanguage>
        Allow ?% discount
      </naturalLanguage>
      <variable name="Discount" type="double"/>
    </action>
  </actions>
</facts>
    
```

The conditions and actions tags contain conditions and actions that can be used in the decision table, respectively. Each of the conditions and actions can be used multiple times by customized the variable value. The naturalLanguage tags contain values that are shown to represent condition or action in the decision table. The variable tags specify the name and type of variable used in the condition or action. The type attribute in action specifies type of action i.e. returning value to client or web service calls.

After users successfully create the decision table, it is store in a RuleML file to persist the business rules in the decision table.

```

<RuleML>
  <Implies>
    <body> <Atom>
      <op> <Rel>not equal</Rel> </op>
      <Var type="xsd:string">CustomerType</Var>
      <Ind>student</Ind>
    </Atom> </body>
    <head><Atom>
      <op><Rel>return</Rel></op>
      <Ind type="xsd:int">Discount</Ind>
      <Ind>0</Ind>
    </Atom></head>
  </Implies>
  ...
</RuleML>
    
```

The RuleML tag contains all of the business rules in the decision table and each business rule exists in an Implies tag. The body tag contains conditions whereas the head tag contains actions. The above business rule means “If a customer is not a student, not allow discount” or the third

business rule in the above decision table. Finally, all business rules in the decision table are transformed into BPEL as following pseudo code:

```

IF (CustomerType == "student" AND OrderSize > 7)
  THEN Discount = 15
ELSE IF (CustomerType == "student" AND NOT
(OrderSize > 7))
  THEN Discount = 10
ELSE IF (CustomerType != "student")
  THEN Discount = 0
RETURN Discount

```

The web service resulted from the above BPEL has an interface as following pseudo code:

```
double GetDiscount (String CustomerType, int orderSize)
```

The web service receives two parameters i.e. customerType, orderSize and returns percent discount to web service clients.

VII. RELATED WORK

There are many works that focus on integrating between business rules and business processes. In [5], the authors propose a method in order to separate business rules from business processes. They define business processes in BPEL and use principle of Aspect-Oriented Programming to separate business rules from business processes. However, their business rules have to be directly defined with BPEL not decision tables. Furthermore, the business rules are finally merged with existing BPEL processes whereas our business rules can work as independent web services. Reference [16] proposes a rule engine that inferences RuleML business rules. It also proposes a rule editor to create and edit rules but the result rules are RuleML format that have to work with its rule engine and can not work as web services. Reference [21] applies business rules to control business processes by inserting business rules into the header of an SOAP-envelope. Business rules are evaluated among intermediate web services thus the central rule engine is not required. Nevertheless, the business rules are embedded in SOAP messages and can not work independently.

VIII. CONCLUSION

In this paper we have presented the approach to define business rules separated from business logic by using decision tables. With our approach, the non-technical users expert in business rules can define business rules closely aligning organization goals without programming skill.

Therefore the approach decreases time consumed to create business rules and increase maintainability and verifiability of them. After business rules are created in decision tables, they are stored in RuleML files to persist them and are finally transformed into BPEL files. The BPEL files can be executed by BPEL engines and exposed as business rule web services. The result business rules, therefore, are reusable and have agility to change. Because of BPEL specification, the business rules can be specified to invoke other web services. Consequently, the business rules from our approach provide collaboration between heterogeneous applications or web services to construct and support business processes efficiently.

REFERENCES

- [1] B.V. Halle, *Business Rules Applied*, John Wiley & Son, Inc, 2001.
- [2] ILOG, *Decision Services: The Next SOA Challenge*, <http://www.ilog.com/products/jrules/whitepapers/index.cfm?filename=WP-SOA.pdf>, June 2006.
- [3] A.Alves et al., *Web Services Business Process Execution Language for version 2.0*, <http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.html>, January 2007.
- [4] The Business Rules Group, *Defining Business Rules, What are they really?* www.businessrulesgroup.org, July 200.
- [5] A. Charfi, M. Mezini. *Hybrid Web Service Composition: Business Processes Meet Business Rules*, Darmstadt University of Technology, 2004.
- [6] T. Auechaikul and W. Vatanawood, *A Development of Web Services Sequencer for Business Process*, Department of Computer Engineering, Chulalongkorn University, 2006.
- [7] M. Norton, *Decisioning : A New Approach to Systems Development (Part 1)*, <http://www.brcommunity.com/b326a.php>, January 2007.
- [8] J. Pasley, *How BPEL and SOA are changing Web Services Development*, IEEE Computer Society, June 2005.
- [9] G. Knolmayer et el, *Modeling Processes and Workflows by Business Rules*, 2000.
- [10] Stefan Tilkov, *WS-BPEL 2.0 becomes an OASIS Standard*, <http://www.infoq.com/news/2007/04/ws-bpel-2.0>, April 2007.
- [11] H. Bowers, *Building Rules into Web Services Applications*, <http://www2.sys-con.com/ITSG/virtualcd/WebServices/archives/0212/bowers/index.html>, 2004.
- [12] K. Molay, *Business Rules: The Perfect Complement to Web Services*, <http://webservices.sys-con.com/read/39597.htm>, October 2002.
- [13] Codasyl, *A Modern Appraisal of Decision Tables*, Report of the Decision Table Task Group, ACM, New York, 1982.
- [14] J. Vanthienen, G. Wets: *Integration of the Decision Table Formalism with a Relational Database Environment*, Elsevier Science Ltd, 1995.
- [15] <http://www.ruleml.org/#Participants-Systems>
- [16] C. Costello, O. Molloy. *Orchestrating Supply Chain Interactions Using Emerging Process Description Language and Business Rules*. Sixth International Conference on Electronic Commerce, 2004.
- [17] J. K. Lee, M.M. Sohn. *The eXtensible Rule Markup Language*. Communications of The ACM, May 2003.
- [18] Gerd Wagner: [How to Design a General Rule Markup Language?](http://www.ruleml.org/#Participants-Systems), Invited Talk, Workshop XML Technologien für das Semantic Web (XSW 2002), Berlin, June 2002.
- [19] <http://www.w3.org/Math/>
- [20] Ronal G.Ross, *Principles of the Business Rule Approach*, Addison Wesley, January 2003.
- [21] R. Schmidt, *Web Services Based Execution of Business Rules*,.Proceeding of the International Workshop on Rule Markup Language for Business Rules on the Semantic Web, volume 60, Sardinia, Italy, 2002.

Generating BPEL Scripts for Business Rules from a Decision Table and Domain-Specific Language

Thanawut Auechaikul and Wiwat Vatanawood
 Department of Computer Engineering, Chulalongkorn University
 Bangkok, Thailand

Email: Thanawut.A@Student.chula.ac.th, wiwat@chula.ac.th

Abstract

At present, business rules and business processes are increasingly significant for organizations. The business processes enable organization to quickly respond to changes and align with business objectives. The business rules are recognized as core of the business processes and one of key factors in successful organizations. However, these rules are traditionally embedded in the business process logic. The business rule experts who are typically not experienced programmer, such as business analysts essentially implement the business rules through developers. This causes the business rules used in IT systems not align with organization goals and lack of agility to change. This paper presents a business analyst friendly method that enables them to manipulate and manage the business rules themselves with a DSL (Domain-specific language) and decision tables. Finally, these rules are exposed as web services in BPEL (Business Process Execution Language) to support the expected business processes.

Key Words: business rules, business processes, decision tables, web services, BPEL, Domain-specific languages.

1. Introduction

Business processes nowadays are one of main determining factors for any successful organization. Meanwhile, business rules perceived as the important part of business processes are leveraged in decision-making, defining policy, reacting to events and being constraint of organizations [7]. Also, they are widely used in any organization for example discount policy, budget approval and insurance approval. In short, the business rules synchronize activities occurring in the business processes of organizations. Typically, these rules are defined by business rule experts, such as business analysts, who are normally inexperienced programmers. However, they are traditionally embedded within the applications implemented in programming languages and consequently difficult for the business analysts, who are probably ordinary users but truly know them, to understand. Moreover, it is difficult to maintain and verify that the implementation satisfies the requirements

[9]. Such business rules lack a solution for agile making changes without modifying existing business process. Because of all of the mentioned difficulties, we should specially manage and define the business rules explicitly and make them can be directly defined by domain experts who are probably non-programmers.

Web services provide interoperability solutions for sharing data and functionality across heterogeneous applications. They can be composed as to perform a new business process using orchestration concept. BPEL (Business Process Execution Language) [1] and BPML [2] are languages that define the business processes by specifying the control and data flows among the web services. We focus on BPEL since it is becoming the preferred standard for web service composition and encouraged by many vendors.

Domain-specific languages (DSLs) provide a solution to directly represent and implement particular problem domains [6]. They can represent domain concepts as textual or graphics. Also, they enable the development of specific solution easier than general-purpose programming languages. Many DSLs are designed to be comprehensible for non-programmer domain experts by covering complexity and implementation details and providing familiar interface. Consequently, the DSLs enable the domain experts and programmers to identically realize problem domain.

This paper presents a method to bridge the gap between the business rule experts like business analysts and the actual business rules implemented in IT systems by leveraging decision tables and a DSL. Typically, the business analysts do not have experience in programming. Therefore, our approach applies decision tables and the DSL to represent the business rules and transforms them into BPEL. The generated BPEL business rules are exposed as web services to support business processes. This approach enables the business analysts to define and directly implement the business rules aligning with business objectives. Moreover, these business rules extracted from other business logic can repeatedly collaborate with other applications by invoking other web services or can be invoked from other systems to compose agile business processes. The mentioned approach allows business rules experts in organizations to implement the business rules themselves, in contrast to traditional approach which rules are implemented through developers. Therefore, our approach achieves decreasing

time consumed to define the business rules and increasing reusability, maintainability and verifiability of them.

This paper is organized as follows. In next section we introduce some basic concepts of our approach. We then briefly describe the architecture and components of proposed solution. Section 4 presents the detail of design and implementation. Section 5 demonstrates the case study. Section 6 presents related works and in section 7 we finally conclude this work.

2. Background

2.1 Business rules

According to the Business Rules Group [3], a business rule is a statement that defines or constraints some aspects of the business. It can be a policy, agreement or definition relating to manner which organizations do with their customers, partners and employees. A business rule typically consists of a collection of terms, facts and rules. A term is a basic word or word phrase in any natural language (such as English) that employees can realize and share in the business process such as “customer” and “order” [13]. A fact is a statement that connects terms through prepositions and verb phrases into sensible business-relevant observations such as “Customer place order”. A rule is a statement that is applied in decision-making or computation to produce new information. All rules can be expressed by using “IF-THEN” format.

Traditionally, the business rules are embedded within the business process logic. This causes problems of reusability, maintaining and management. The modification of the business rules is not agile and impacts the other business processes. To address the problems, the business rules have to be managed and defined separately from the rest of business processes, and then be enabled to work independently. This approach increases reusability and loose coupling of the business rules in IT systems.

2.2 BPEL

Process composition languages such as BPEL and BPML provide collaboration among web services and define business process flows. However, developers are increasingly using BPEL based on XML for modeling business processes within the Web service architecture [10]. It supports two different kinds of processes i.e. executable business processes that can be executed by a BPEL engine and abstract business processes that are not intended to be executed and specify some agreed behavior that partners in a communication scenario can agree upon.

BPEL includes several features for web services collaboration such as invoking a web service and assigning value to an XML document. It also has commands for control flow such as if and else if.

2.3 Decision table

A decision table is a structure that describes a set of related decision rules [5]. The basic parts of a decision table are shown in Table 1. The upper left portion called the condition stub contains the statements of the conditions. The lower left portion called the action stub contains the statements of the actions. The condition entries and action entries are in the upper right and lower right portions, respectively. Each column in the entry portions forms a decision rule that can be written in an “IF-THEN” statement. For example in Table 1, the first column can be expressed in “IF Age \geq 15 and Gender is Female THEN Charge member fee \$100”. The “-” in the last column means irrelevant condition (condition does

Table 1 The extended entries decision table

Condition Stub	Condition Entries		
	Age	\geq 15	\geq 15
Gender	Female	Male	-
Action Stub	Action Entries		
Charge member fee	100	150	0

not matter or don’t care). In an unambiguous decision table, every possible case of conditions matches only one column [16]. Therefore, a decision table can be implemented with an “IF-THEN-ELSE_IF-THEN-ELSE_IF-THEN...” statement.

3. Architecture of our approach

This paper presents an approach to define business rules in IT systems with the DSL and decision tables in order that users who truly know the business can create the business rules aligning the organization goals themselves. Our approach provides interface between business analysts, which are business rule expert but inexperienced programmer, and the implemented business rules. They can define and edit the business rules via the DSL and can verify and consider the specific rules via the decision tables. Both decision tables and DSL express the business rules by based on predefined facts that we will describe later. After creating the business rules, all of them are stored in XML format for persistence purpose. Finally they are transformed into BPEL. The business rules in BPEL can collaborate with web services in order to support business processes. The architecture of our approach is shown in Figure 1.

The approach consists of five components: the predefined facts, the DSL rule editor, the decision table, business rules in XML format, and rule in BPEL.

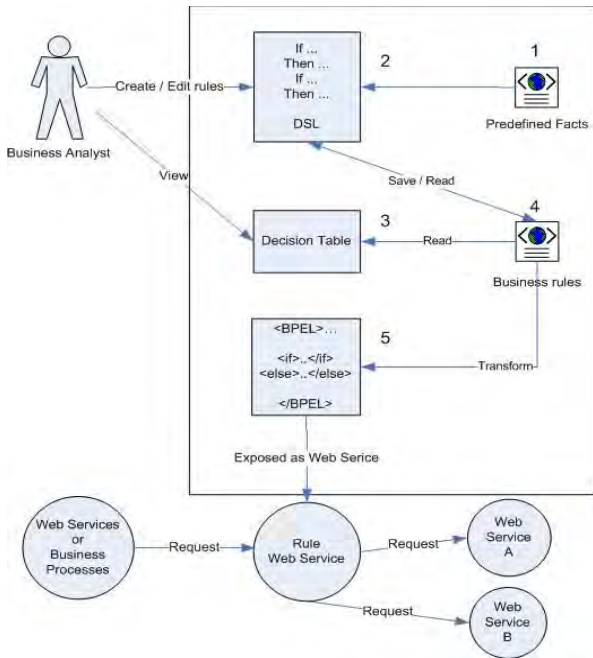


Figure 1 The architecture of the approach

3.1 Predefined facts

The predefined facts persisted in a XML file define templates of facts available in the business rules. There are two types of fact: condition and action as Figure 2. Both of them have variables that users have to define.

The action type can be to return a value to client or web service call. If the action is to return a value, the returned variable or value has to be specified. If the action is a web service call, necessary artifacts such as the WSDL location of the called web service and a list of web service parameters are required.

The predefined facts also include natural language sentences (in <naturalLanguage> tag). This enables the DSL and decision tables represent conditions and actions in ordinary user-friendly manner. For example, if there is a evaluated variable in a condition as CustomerType, the sentence presented in the DSL and decision tables is "Customer is ?". The "?" will be replaced with a user-defined value when displayed in the DSL and decision tables.

3.2 DSL rule editor

The DSL rule editor provides an environment that allows creation and modification of business rules. The editor extracts facts from the mentioned predefined facts and displays them in easily understandable way as Figure

```
<facts>
  <conditions>
    <condition name="customerTypeCond">
      <naturalLanguage>
        Customer is ?
      </naturalLanguage>
      <variable name="CustomerType"
        type="string" />
    </condition>
    ...
  </conditions>
  <actions>
    <action name="allowDiscount">
      <naturalLanguage>
        Allow ?% discount
      </naturalLanguage>
      <variable name="Discount"
        type="double"/>
      <return variable="Discount" />
    </action>
    <action name="RecordLog" >
      ...
      <variable name="CustomerName"
        type="string"/>
      <invoke service="RecordLogWS"
        operation="RecordLog"
        wsdl="LogWS.wsdl">
        <parameter variable="CustomerName"/>
      </invoke>
      ...
    </action>
  </actions>
</facts>
```

Figure 2 Predefined facts contain the templates of conditions and actions

3. The business analysts choose and add actions and conditions to a rule, and then customize the related variables. For example, from previous example, users can define the condition as "Customer is library". Since the DSL is similar to natural language, this obviously enables non-programmer business rule experts can define rules themselves. After the business rules are created and stored in XML files, the users can edit via the editor and view through the decision tables later.

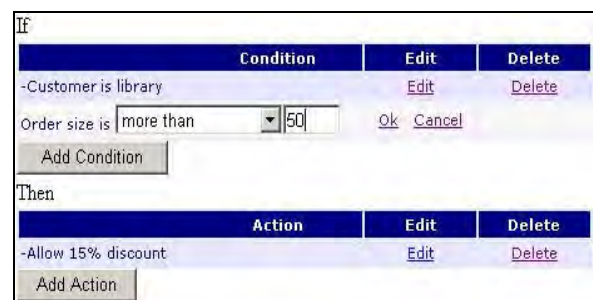


Figure 3 The business rule is displayed as rows in the decision table.

3.3 Decision table

The decision tables represent business rules to users. Users without programming experience can verify and consider the business rules via them. In our decision table,

each rule is represented as a row. Meanwhile, each of the conditions and actions is represented as a column and expressed with the natural language sentence as shown in Table 2.

Table 2 The business rule is displayed as rows in the decision table.

	Condition	Condition	Action
#	Customer is ?	Order size is ?	Allow ?% discount
1	bookstore	more than or equal 6	25
2	bookstore	less than 6	0
3	library	more than or equal 50	15
4	library	between 6-49	10
5	library	less than 6	5

3.4 Business rules in XML format

After the business rules are created in the DSL rule editor, they are stored in XML files to persist them. The XML files contain details of the conditions and actions customized by users such as the values of variables used in them.

3.5 Business rule in BPEL

BPEL files contain business rules in form of “IF-THEN-ELSE_IF-THEN-ELSE_IF-THEN...”. They are generated from the previous business rules in XML. These files can be executed by BPEL engines to expose the business rules as web services (called business rule services). The business rule services can serve and invoke other web services to support business process collaboration.

4. Creation and transforming of the business rule

As mentioned previously, a decision table describes a set of related rules or a rule set (a business rule). Therefore, we apply a decision table to describe a business rule.

4.1 Creating business rule from the predefined facts with the DSL rule editor

For the creation of the business rule, we design a predefined facts XML file as shown in Figure 2. It acts like material for business rules creation and consists of condition and action templates. The <conditions> and <actions> tags contain a set of conditions and actions that can be applied in the business rules, respectively. The variable that is related to them is specified with a

<variable> tag. This tag specifies the name and type of the variable used in the condition and action.

The <action> tag specifies an activity performed in the business rule. The activity can be to return value to client (<return> tag) or to call web service (<invoke> tag). These tags specify the applied variables as shown in Figure 2.

4.2 Business rules in XML file

All created business rules are stored in a business rules XML file. It keeps the information of the decision tables. The DSL rule editor and the decision tables always read this file to extract the rule sets. After creating the business rule, the rule set from the decision table will be

```
<BusinessRules>
  <facts name="factsForSale"
    location="predefined_facts_sale.xml"/>

  <ruleset name="discount_book_rules"
    location="discount_ruleset.xml"
    facts="factsForSale"
    description="Discount for novel" />

  <ruleset name="book_promotion_rules" .../>
</BusinessRules>
```

Figure 5 All of the decision tables are stored in form of rule sets in the business rules XML file.

transformed into the business rules XML file as shown in Figure 5.

A <BusinessRules> tag contains all of the decision tables or rule sets. A <ruleset> tag specifies the location of the XML file that stores a rule set or a decision table, and the applied facts. Meanwhile, a <facts> tag specifies the location of the predefined facts file that can be used by several decision tables.

Each decision table is stored in its own XML file as shown in Figure 6, called the rule set file. Each rule is stored in a <rule> tag. The <condition> tag specifies the Boolean expression. Each condition of the rule from the table, such as “OrderSize is more than 6” is kept in a logical operator tag, such as <moreThan> tag depending on the operator in the condition. These tags have three attributes i.e. condition, leftId and rightId. The first attribute refers to the condition name in the predefined facts. The rest attributes describe a left side and right side of the logical expression. For example in Figure 6, the Boolean expression is “OrderSize > 6 AND CustomerType= 'bookstore’”. The <variable> tag in <action> tag specifies a value of the related variable in the action.

```

<ruleset name="DiscountCalculation" >
  <rule name="rule_1" >
    <condition>
      <And>
        <moreThan
          condition="orderSizeCond"
          leftId="OrderSize"
          rightId="6"/>
        <equals
          condition="customerTypeCond"
          leftId="CustomerType"
          rightId="bookstore" />
      </And>
    </condition>
    <action name="allowDiscount">
      <variable name="Discount"
value="25"/>
    </action>
  </rule>

  <rule name="rule_2" >...</rule>
</ruleset>

```

Figure 6 The rule set file store a decision table.

4.3 Business rule BPEL generation

Each of the rule set XML files is transformed into a BPEL file in form of “IF-THEN-ELSE_IF-THEN-ELSE_IF-THEN...” as Figure 7. This file compromises with the BPEL 2.0 specification and can be executed by compatible BPEL engines. As Figure 7, an `<if>` tag stores all rules of decision table. For the first rule, the condition and action are specified in the first `<condition>` and `<sequence>` tag, respectively. From the rule set file, the `<And>` tag inside the `<condition>` tag is transformed in a Boolean expression i.e. “OrderSize \geq 6 and CustomerType= “bookstore”” as Figure 7. The each of rest rules is specified in an `<elseif>` tag.

The XML schema is also generated to define the complex type of the input variable and output variable used in the BPEL. These types depend on the type of the variables used in the actions and conditions.

As mentioned previously, the action can be to return value or web service call. The first type is implemented with an `<assign>` tag. This tag assigns the returned value to the variable that is replied to the client. For example from Figure 7, it assigns 25 to Discount element of DiscountOut variable. Finally, all of the returned values are combined as the output variable and are replied to the client with a `<reply>` tag. In the BPEL, the web service call action is implemented with an `<invoke>` tag. This tag specifies the called web service and parameters.

5. Case study

We apply our approach with a case study i.e. a book discount policy. The result decision table is shown as Table 2. From the table, there are obviously two variables in the condition part i.e. customer type and order size.

```

<process name="DiscountCalculation">
  <variables>
    <variable name="DiscountIn"
      type="DiscountInType"/>
    <variable name="DiscountOut"
      type="DiscountOutType"/>
  </variables>
  <sequence>
    <if>
      <condition>
        OrderSize  $\geq$  6 and
        CustomerType='bookstore'
      </condition>
      <sequence>
        <assign name="Assign1">
          <copy>
            <from>25</from>
            <to>${DiscountOut/Discount}</to>
          </copy>
        </assign>
      </sequence>
    <elseif>...</elseif>
    ...
  </if>
  <reply variable="DiscountOut" />
</sequence>
</process>

```

Figure 7 The generated business rules in BPEL

Meanwhile, there is only one variable in the action part i.e. discount. Consequently, the predefined facts of the decision table are declared as Figure 2 (the RecordLog action is irrelevant. It is shown to be example of the action of web service call). The result business rule and BPEL are generated as Figure 6 and 7, respectively. When the BPEL is executed by a BPEL engine, it can work as a web service to support business processes as shown in Figure 8.

6. Related work

There are many works that focus on integrating between business rules and business processes. In [4], the authors propose a method in order to separate business rules from business processes. They define the business processes in BPEL and use principle of Aspect-Oriented Programming to separate the business rules from the business processes. However, their business rules have to be directly defined with BPEL not decision tables. Furthermore, they are finally merged with existing BPEL processes whereas our business rules can work as independent web services. Schmidt [14] applies business rules to control business processes by inserting the business rules into the header of an SOAP-envelope. The business rules are evaluated among intermediate web services. However, the business rules are embedded in SOAP messages and can not work independently. In [12], the authors propose a method that integrates business rules into business process implemented by BPEL. They

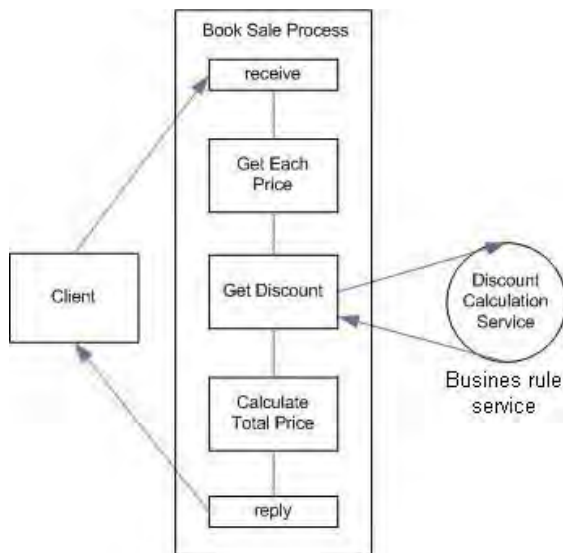


Figure 8 The business rule service (Discount Calculation Service) supports the business process.

present the Rule Interceptor Service that intercepts all incoming and outgoing BPEL Web service calls and then apply business rules with the exchanged messages. However, their work does not focus on business rule creation.

The DSLs are also applied in the wide area of researches. Hamdi et al. [8] have developed a DSL named PPL for specify security policy in distributed systems. They claim that the DSL allows the expression of complicated security policy easily. Peltier [11] has proposed a DSL for model transformation in Model-Driven Architecture. He demonstrates that the DSL enables transformations to be expressed in a formal, executable, short learning curve required and human-readable way.

7. Conclusion

In this paper we present an approach to define and verify business rules separated from business logic by using the DSL and decision tables. The DSL offers the readable textual. Meanwhile, the decision tables present the easily comprehensible graphics for non-programmer business rule experts. Both of them cover complexity and implementation details, and provide familiar interface for the business rule experts. With our approach, they can define the business rules closely aligning with organization goals without experienced programming skill. The business analysts can create or modify the business rules in implemented business process directly. Therefore this approach decreases time consumed to manage business rules and increase maintainability and verifiability of them. After the business rules are created in the DSL, they are stored in the business rule XML files to persist them and are finally transformed into the BPEL

files. With this approach, the result business rules can be reusable and have the ability to change. They also provide collaboration between heterogeneous applications or web services to construct and support business processes efficiently.

8. Acknowledgement

This research is part of the Engineering New Paradigm Software for Enterprises with Service-Oriented Architecture Project, supported by Thailand's Software Industry Promotion Agency (Public Organization).

9. References

- [1] Alves, A. et al., 'Web Services Business Process Execution Language for version 2.0', <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>, Jan 2007.
- [2] Arkin, A., 'Business Process Modeling Language', <http://www.bpmi.org>, Nov 2002.
- [3] Business Rule Group, www.businessrulesgroup.org, 2007.
- [4] Charfi, A. et al., 'Hybrid Web Service Composition: Business Processes Meet Business Rules', International Conference on Service Oriented Computing, Nov 2004.
- [5] Codasyl, 'A Modern Appraisal of Decision Tables', Report of the Decision Table Task Group, 1982, pp. 230-232.
- [6] Wada, H. and Suzuki, J., 'A Domain Specific Modeling Framework for Secure Network Applications', Proceedings of the 30th Annual International Computer Software and Applications Conference, Sept 2006, pp.353-355.
- [7] Halle, B.V., 'Business Rules Applied', John Wiley & Son, Inc. 2001.
- [8] Hamdi, H., Mosbah, M. and Bouhoula, A., 'A Domain Specific Language for Securing Distributed Systems', Systems and Networks Communications International Conference 2007, Aug 2007.
- [9] Norton, M., 'Decisioning: A New Approach to Systems Development (Part 1)', <http://www.brcommunity.com/b326a.php>, Jan 2007.
- [10] Pasley, J., 'How BPEL and SOA are changing Web Services Development', IEEE Computer Society, June 2005.
- [11] Peltier, M., 'MTrans, a DSL for Model Transformation', Proceedings of the Sixth International ENTERPRISE DISTRIBUTED OBJECT COMPUTING Conference, Sept 2002, pp.190-199.
- [12] Rosenberg, F. and Dustdar, S., 'Business Rules Integration in BPEL – A Service-Oriented Approach', International Conference of E-Commerce Technology, Nov 2005.
- [13] Ross, R., 'Principles of the Business Rule Approach', Addison Wesley, Jan 2003.
- [14] Schmidt, R., 'Web Services Based Execution of Business Rules', Proceeding of the International Workshop on Rule Markup Language for Business Rules on the Semantic Web, June 2002.
- [15] Thibault, S., Marlet, R. and Consel, C., 'Domain-specific languages: from design to implementation application to video device drivers generation', Software Engineering, IEEE Transactions, Issue 3, June 1999, pp.363-377.
- [16] Vanthienen, J. and Wets, G., 'Integration of the decision table formalism with a relational database environment', Information Systems, Issue 7, Nov 1995, pp. 595-616

ประวัติผู้เขียนวิทยานิพนธ์

นาย ธนาวุฒิ เอื้อชัยกุล เกิดเมื่อวันที่ 26 กรกฎาคม พ.ศ.2526 จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2549 หลังจากนั้นเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2549