

การออกแบบส่วนต่อขยายเอกซ์เอตี่แอลสำหรับการออกแบบโครงสร้างสถาปัตยกรรมเชิงบริการ

นายภาสกร ปานนอก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the Graduate School.

Design of an xADL Extension for Service Oriented Architecture Design

Mr. Passakorn Pannok

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Copyright of Chulalongkorn University

นายภาสกร ปานนอก : การออกแบบส่วนต่อขยายเอกซ์เอดีแอลสำหรับการออกแบบ
 โครงสร้างสถาปัตยกรรมเชิงบริการ (Design of an xADL Extension for Service
 Oriented Architecture Design) อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : รองศาสตราจารย์ ดร.
 วิวัฒน์ วัฒนาวุฒิ, 88 หน้า

สถาปัตยกรรมเชิงบริการคือ การนำทรัพยากรที่มีความหลากหลายและซับซ้อนภายในองค์กร
 มาทำงานร่วมกันในลักษณะซอฟต์แวร์เชิงบริการ ซึ่งนิยมใช้ภาษาเอสโอเอเอ็มแอลเป็นแนวทางในการ
 ออกแบบโครงสร้างของสถาปัตยกรรมบริการ แต่ในสภาพแวดล้อมจริงนั้นแบบจำลองสถาปัตยกรรม
 ซอฟต์แวร์จะประกอบด้วยรูปแบบสถาปัตยกรรมที่หลากหลายรวมทั้งสถาปัตยกรรมเชิงบริการ การใช้
 ภาษาอธิบายสถาปัตยกรรมเป็นทางเลือกที่เหมาะสมสำหรับแบบจำลองที่หลากหลายนี้ ซึ่งภาษา
 อธิบายสถาปัตยกรรมเอกซ์เอดีแอลเป็นหนึ่งในหลายภาษาที่มีความยืดหยุ่นและได้รับการขยายต่อให้
 ครอบคลุมประเภทสถาปัตยกรรมและโครงสร้างใหม่ๆได้

ในวิทยานิพนธ์นี้จึงนำเสนอวิธีการออกแบบส่วนต่อขยายภาษาอธิบายสถาปัตยกรรม
 เอกซ์เอดีแอลให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ ซึ่งมีการอ้างอิงแนวคิดจาก
 ดับบลิวเอสดีแอล โดยเพิ่มอิลิเมนต์สำหรับอธิบายบริการภายในอินเตอร์เฟซด้วยเอกซ์ลิงก์ จำนวน 1
 อิลิเมนต์ และเพิ่มอิลิเมนต์เชื่อมโยงการอธิบายคุณสมบัติอยู่ภายนอกโครงสร้างสถาปัตยกรรม จำนวน
 2 อิลิเมนต์ ได้แก่ อิลิเมนต์ ServiceInterfaceProvided สำหรับข้อมูลอธิบายคุณสมบัติของผู้
 ให้บริการ และอิลิเมนต์ ServiceInterfaceRequired สำหรับอธิบายคุณสมบัติของผู้ร้องขอบริการ
 โดยในส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอลได้กำหนดอย่างระมัดระวังเพื่อที่จะรักษา
 โครงสร้างที่ดีของสคีมาภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอลเดิม และพัฒนาเครื่องมือที่สามารถ
 วิเคราะห์โครงสร้างของสถาปัตยกรรมเชิงบริการ รวมทั้งรายงานผลข้อมูลจากการวิเคราะห์โครงสร้าง
 ของสถาปัตยกรรมเชิงบริการได้

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....

สาขาวิชา.....วิศวกรรมซอฟต์แวร์.....ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก.....

ปีการศึกษา.....2556.....

5470485021 : MAJOR SOFTWARE ENGINEERING

KEYWORDS : SOFTWARE ARCHITECTURE; ARCHITECTURAL DESCRIPTION LANGUAGE (ADL);
SERVICE ORIENTED ARCHITECTURE; XADL

PASSAKORN PANNOK: DESIGN OF AN XADL EXTENSION FOR SERVICE ORIENTED
ARCHITECTURE DESIGN. ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, PH.D, 88
pp.

Service Oriented Architecture demonstrates the concept that the diverse and complex resources in an organization distributedly work together as a service application. Service Oriented Architectural Modeling Language (SOAML) becomes lately a potential metalanguage in the design of Service Oriented architecture. But in the real environment, a software architectural model is composed of the hybrid architectural styles including SOA. We believe the usage of Architectural description language is more appropriate choice for hybrid architecture model. Among the variety of the Architectural description language, xADL is more flexible and extensible to the new architectural types and templates.

In this thesis, propose how to design an xADL extension to cope with SOA structure and reference the concept of WSDL. We added elements for describing the service interface one xLink element and two external elements called ServiceInterfaceProvided for describing provided services and ServiceInterfaceRequired for describing required services. The extended part of xADL is carefully defined in order to preserve the standard well-formalness of the original xADL schema. And a tool is developed to analyze the SOA structure and report the results of the analysis.

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

Field of Study : Software Engineering Advisor's Signature.....

Academic Year : 2013.....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จไปด้วยดี เนื่องจากด้วยความกรุณาของผู้ที่เกี่ยวข้องทุกฝ่าย ทางผู้วิจัยขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ เป็นอย่างยิ่งที่ได้สละเวลา ดูแลเอาใจใส่ ให้คำแนะนำ คำปรึกษา และให้แนวทางที่เป็นประโยชน์ สำหรับการทำวิทยานิพนธ์นี้ จนสำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ รองศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์ ประธานการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์ ผู้ช่วยศาสตราจารย์ ดร.ทรงศักดิ์ รองวิริยะพานิช ที่กรุณาสละเวลามาเป็นกรรมสอบวิทยานิพนธ์ และให้คำแนะนำ ข้อคิดเห็นเพื่อปรับปรุงแก้ไขวิทยานิพนธ์

ขอขอบคุณ คณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ และเพื่อนๆวิศวกรรมซอฟต์แวร์ทุกคน ที่คอยช่วยเหลือ ให้ความรู้ คำแนะนำที่เป็นประโยชน์ และกำลังใจตลอดมา

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา และสมาชิกในครอบครัวทุกท่านที่ให้การสนับสนุนและเป็นกำลังใจ จนสามารถวิทยานิพนธ์เล่มนี้สำเร็จนี้ได้สำเร็จ

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ	ฎ
บทที่ 1 บทนำ.....	1
1.1. ที่มาและความสำคัญของปัญหา	1
1.2. วัตถุประสงค์	2
1.3. ขอบเขตการดำเนินงาน	2
1.4. ขั้นตอนการวิจัย	4
1.5. ลำดับขั้นตอนการเสนอวิทยานิพนธ์นี้	4
1.6. ผลงานที่ตีพิมพ์จากวิทยานิพนธ์	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1. ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 สถาปัตยกรรมซอฟต์แวร์	6
2.1.2 ภาษาอธิบายสถาปัตยกรรม	7
2.1.3 ภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดดีแอล 2.0.....	9
2.1.4 สถาปัตยกรรมเชิงบริการ	15
2.1.5. ภาษาเอสโอเอเอ็มแอล.....	17

2.1.6.	ภาษาดับบลิวเอสดีแอล.....	18
2.1.7.	ภาษาเอกซ์เอ็มแอล	19
2.1.8.	เอกซ์เอ็มแอลสคีมา	20
2.2.	งานวิจัยที่เกี่ยวข้อง.....	22
2.2.1	Extending xADL with Statechart Behavioral Specification	22
2.2.2.	A New Architecture Description Language for Service-Oriented Architecture.....	23
2.2.3.	An xADL Extension for Managing Dynamic Deployment in Distributed Service Oriented Architectures.....	24
2.2.4.	Adding Aspects to xADL 2.0 for Software Product Line Architectures.....	25
บทที่ 3	การวิเคราะห์และวิธีการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล.....	26
3.1.	ภาพรวมและแนวคิดในการพัฒนาส่วนต่อขยาย.....	26
3.2.	การพัฒนาส่วนต่อขยาย	28
บทที่ 4	การพัฒนาเครื่องมือ	36
4.1.	สภาพแวดล้อมในการพัฒนาระบบ	36
4.2.	สภาพแวดล้อมในใช้งานเครื่องมือ.....	36
4.3.	ภาพรวมการทำงานของเครื่องมือ.....	37
4.4.	การวิเคราะห์และออกแบบเครื่องมือ	39
4.4.1.	แผนภาพยูสเคส.....	39
4.4.2.	แผนภาพคลาส	42
4.4.3.	แผนภาพลำดับ	45

4.5. โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ	47
บทที่ 5 การทดสอบ.....	50
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....	64
รายการอ้างอิง.....	67
ภาคผนวก ผลลัพธ์จากตัวอย่างทดสอบระบบสั่งซื้อสินค้า.....	69
ประวัติผู้เขียนวิทยานิพนธ์	88

สารบัญตาราง

หน้า

ตารางที่ 3.1 ตารางเปรียบเทียบคุณสมบัติของการอธิบายโครงสร้างสถาปัตยกรรมเชิงบริการ.....	35
ตารางที่ 4.1 รายละเอียดยูสเคสนำเข้าเพิ่มข้อมูลเอกซ์เอตีแอล	40
ตารางที่ 4.2 รายละเอียดยูสเคสตรวจสอบโครงสร้างของสถาปัตยกรรม	41
ตารางที่ 4.6 รายละเอียดยูสเคสดูรายงาน HTML.....	41
ตารางที่ 5.1 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ServiceOrderList.....	52
ตารางที่ 5.2 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqShippingStatus	52
ตารางที่ 5.3 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqPayCredit	53
ตารางที่ 5.4 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqCheckStock	53
ตารางที่ 5.5 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqOrderList.....	54
ตารางที่ 5.6 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqDecreaseStock	54
ตารางที่ 5.7 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ShippingStatus.....	54
ตารางที่ 5.8 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส CheckStock.....	55
ตารางที่ 5.9 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส DecreaseStock.....	55
ตารางที่ 5.10 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqAddAccountPayable	56
ตารางที่ 5.11 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส AddAccountPayable.....	57
ตารางที่ 5.12 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส AddAccountPayable.....	57
ตารางที่ 5.13 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส ReqAddAccountsReceivable	58
ตารางที่ 5.14 คุณสมบัติของบริการภายในอินเทอร์เน็ตเฟส PayCredit	58

สารบัญภาพ

หน้า

รูปที่ 2.1 โครงสร้างนามธรรมของโครงสร้างสถาปัตยกรรม	7
รูปที่ 2.2 โครงสร้างการเชื่อมต่อระหว่างอินเทอร์เน็ตเฟสผ่านลิงก์.....	11
รูปที่ 2.3 การเชื่อมต่อของโครงสร้างสถาปัตยกรรมย่อย	12
รูปที่ 2.4 เมตาโมเดลการอ้างอิงชนิดของแบบจำลอง.....	14
รูปที่ 2.5 ตัวอย่างการใช้ภาษาเอสไอเอ็มแอล	18
รูปที่ 2.6 โครงสร้างของเอกสารภาษาเอกซ์เอ็มแอล.....	20
รูปที่ 2.7 สร้างภาพแบบจำลองโครงสร้างเอกซ์เอ็มแอลสคีมิตัวด้วยเครื่องมือ XMLSpy XML Editor.....	21
รูปที่ 2.8 ตัวอย่างการสร้างเอกซ์ลิงก์แบบพื้นฐาน	21
รูปที่ 2.9 แนวคิดของโมเดลองค์ประกอบของแผนภาพเชิงพฤติกรรม	22
รูปที่ 2.10 เอกซ์เอ็มแอลสคีมิตัวของส่วนต่อขยายแผนภาพเชิงพฤติกรรมในภาษา xADL	23
รูปที่ 2.11 โครงสร้างของภาษา SOADL.....	24
รูปที่ 2.12 โครงสร้างเอกซ์เอ็มแอลสคีมิตัวของส่วนต่อขยาย.....	25
รูปที่ 3.1 อธิบายการเชื่อมต่อของบริการในอินเทอร์เน็ตเฟสของคอมโพเนนท์	27
รูปที่ 3.2 เอกซ์เอ็มแอลสคีมิตัว การเพิ่มอิลิเมนต์ soalink ภายในอิลิเมนต์อินเทอร์เน็ตเฟส.....	29
รูปที่ 3.3 ตัวอย่างการขยายเพิ่ม soalink ภายในอิลิเมนต์อินเทอร์เน็ตเฟส	30
รูปที่ 3.4 เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ ServiceInterfaceProvided	30
รูปที่ 3.5 แผนภาพ เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ Type.....	31
รูปที่ 3.6 เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ Message	31
รูปที่ 3.7 แผนภาพ เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ PortType.....	32
รูปที่ 3.8 เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ Binding.....	32
รูปที่ 3.9 เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ Service.....	33
รูปที่ 3.10 ตัวอย่างเอกซ์เอ็มแอลของ ServiceInterfaceProvided	33
รูปที่ 3.11 เอกซ์เอ็มแอลสคีมิตัว โครงสร้างของ ServiceInterfaceRequired	34
รูปที่ 3.12 ตัวอย่างเอกซ์เอ็มแอลของ ServiceInterfaceRequired.....	34

รูปที่ 4.1 ข้อมูลเข้าและข้อมูลออกของเครื่องมือประมวลผลที่พัฒนาขึ้น..... 37

รูปที่ 4.2 แผนภาพกิจกรรมการทำงานของระบบ เครื่องมือประมวลผลโครงสร้างสถาปัตยกรรม 38

รูปที่ 4.3 แผนภาพยูสเคสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรม 40

รูปที่ 4.4 คลาสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรม 42

รูปที่ 4.5 คลาส MainApp 42

รูปที่ 4.6 คลาส ReportViewFrame 43

รูปที่ 4.7 คลาส AboutFrame..... 43

รูปที่ 4.8 คลาส FindExtensionValidator..... 43

รูปที่ 4.9 คลาส xADL4SOAGenerator 44

รูปที่ 4.10 แผนภาพลำดับ Input xADL structure 45

รูปที่ 4.11 แผนภาพลำดับ Check SOA structure 46

รูปที่ 4.12 แผนภาพโครงสร้างของจอภาพ เครื่องมือการประมวลผลโครงสร้างของสถาปัตยกรรม.. 47

รูปที่ 4.13 หน้าจอหลักของเครื่องมือ 48

รูปที่ 4.14 หน้าจอข้อมูลเกี่ยวกับเครื่องมือ 48

รูปที่ 4.15 หน้าจอเลือกเพิ่มข้อมูลเข้าสู่ระบบ 49

รูปที่ 4.16 หน้าจอรายงานของเครื่องมือ..... 49

รูปที่ 5.1 แผนภาพคอมโพเนนท์ไดอะแกรมระบบสั่งซื้อสินค้า 51

รูปที่ 5.2 โครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบสั่งซื้อสินค้า ด้วยภาษาอธิบายสถาปัตยกรรม
เอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio..... 59

รูปที่ 5.3 เพิ่มเติมไวยากรณ์ soalink ภายในอินเตอร์เฟซของคอมโพเนนท์ Order Taker 60

รูปที่ 5.4 รายละเอียดคุณสมบัติของบริการ GetOrderService..... 61

รูปที่ 5.5 ตัวอย่างรายงาน คอมโพเนนท์ OrderTaker 62

รูปที่ ก- 1 โครงสร้างสถาปัตยกรรมของตัวอย่างทดสอบ ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตี
แอล โดยใช้เครื่องมือ ArchStudio69

รูปที่ ก-2 รายละเอียดโครงสร้างสถาปัตยกรรมระบบสั่งซื้อสินค้า 74

รูปที่ ก-3 รายละเอียดบริการภายในคอมพิวเตอร์ OrderTaker	75
รูปที่ ก-4 รายละเอียดบริการภายในคอมพิวเตอร์ Shipping Management	76
รูปที่ ก-5 รายละเอียดบริการภายในคอมพิวเตอร์ Invoicer.....	77
รูปที่ ก-6 รายละเอียดบริการภายในคอมพิวเตอร์ ProductPurchase	77
รูปที่ ก-7 รายละเอียดบริการภายในคอมพิวเตอร์ Payment Gateway	78
รูปที่ ก-8 รายละเอียดบริการภายในคอมพิวเตอร์ Stock.....	80
รูปที่ ก-9 รายละเอียดบริการภายในคอมพิวเตอร์ Accounting	81
รูปที่ ก-10 รายงานผลลัพธ์จากการประมวลผลของเครื่องมือ.....	87

บทที่ 1

บทนำ

1.1. ที่มาและความสำคัญของปัญหา

การออกแบบสถาปัตยกรรมซอฟต์แวร์มีส่วนสำคัญที่จะสามารถช่วยให้ผู้พัฒนาสามารถเข้าใจโครงสร้างโดยรวมของซอฟต์แวร์ และใช้ในการออกแบบเพื่อให้ตรงตามข้อกำหนดของความต้องการซอฟต์แวร์เชิงคุณภาพ ซึ่งการออกแบบสถาปัตยกรรมนั้นจะช่วยให้สามารถลดเวลาในการพัฒนาซอฟต์แวร์ การประมาณการทรัพยากร ค่าใช้จ่ายของระบบ และนำไปสู่การพัฒนาซอฟต์แวร์ที่มีคุณภาพ [1]

ในการออกแบบสถาปัตยกรรมซอฟต์แวร์นั้น มีแนวคิดและรูปแบบที่หลากหลาย ทั้งนี้ขึ้นอยู่กับสถานะแวดล้อมการนำไปใช้งานของระบบสารสนเทศ และข้อจำกัดต่างๆของแต่ละองค์กร หนึ่งในแนวคิดของรูปแบบสถาปัตยกรรมที่น่าสนใจอีกแนวคิดหนึ่งก็คือ สถาปัตยกรรมเชิงบริการ (Service Oriented Architecture) ซึ่งมีแนวคิดจากการนำทรัพยากรที่มีความหลากหลาย และมีความซับซ้อนอยู่ภายในองค์กร ให้สามารถนำมาทำงานร่วมกันได้ในลักษณะโปรแกรมประยุกต์เชิงบริการ (Service Application) เพื่อให้เกิดการนำทรัพยากรที่มีอยู่เดิมกลับมาใช้ใหม่ โดยในแต่ละบริการของซอฟต์แวร์ที่พัฒนาขึ้นสามารถสื่อสารด้วยมาตรฐานเดียวกัน ซึ่งในการออกแบบสถาปัตยกรรมเชิงบริการนั้น มีเครื่องมือที่ช่วยในการอธิบายโครงสร้างสถาปัตยกรรมเชิงบริการ ที่เป็นที่ยอมรับก็คือภาษาเอสโอเอเอ็มแอล (SOAML) โดยเป็นภาษาที่ใช้ลักษณะของภาษายูเอ็มแอล (UML) ในออกแบบส่วนประกอบ การเชื่อมต่อ และรูปแบบของการสื่อสารในแต่ละส่วนประกอบ อีกทั้งยังสามารถเพิ่มส่วนต่อขยาย เช่น ส่วนต่อขยายเพื่ออธิบายพฤติกรรมของบริการ เป็นต้น [2] แต่ก็มีเครื่องมืออื่นนอกเหนือจากภาษาเอสโอเอเอ็มแอล คือ การใช้ภาษาอธิบายสถาปัตยกรรม (Architecture Description Language) มาใช้ในการออกแบบโครงสร้างของสถาปัตยกรรมในเชิงนามธรรม (Abstract) ทำให้สามารถอาศัยคอมโพเนนต์ (Component) การเชื่อมต่อ (Link) และการโต้ตอบกัน (Connection) ระหว่างคอมโพเนนต์ของระบบ ทั้งนี้ภาษาอธิบายสถาปัตยกรรมก็มีอยู่หลายภาษา ตัวอย่างภาษาดังกล่าวคือ Darwin, C2, Rapide, Aesop, Wright, SADL, Acme เป็นต้น ผู้วิจัยจึงมีแนวคิดที่จะหาภาษา

อธิบายสถาปัตยกรรมที่สามารถนำไปพัฒนาส่วนต่อขยายเพื่ออธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ และสามารถพัฒนาส่วนต่อขยายเพื่ออธิบายรูปแบบทางสถาปัตยกรรมอื่นที่ต้องการได้ในอนาคต

วิทยานิพนธ์นี้จึงนำเสนอวิธีการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล ให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการได้ โดยทำการเชื่อมบริการในอินเทอร์เน็ตเฟส เพื่ออธิบายความสามารถของส่วนประกอบในสถาปัตยกรรม และพัฒนาเครื่องมือที่สามารถวิเคราะห์โครงสร้าง นำเสนอและอธิบายโครงสร้าง ของสถาปัตยกรรมเชิงบริการ และรายงานผลข้อมูลจากการวิเคราะห์ โครงสร้างของสถาปัตยกรรมเชิงบริการได้โดยอัตโนมัติ

1.2. วัตถุประสงค์

- 1) เพื่อออกแบบส่วนต่อขยายของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ให้มีความสามารถในการอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการได้
- 2) เพื่อพัฒนาเครื่องมือที่ใช้ในการประมวลผลภาษาเอกซ์เอตีแอลฟอร์เอสไอเอใหม่ที่สร้างขึ้น และรายงานผลการวิเคราะห์สถาปัตยกรรมเชิงบริการ

1.3. ขอบเขตการดำเนินงาน

- 1) การออกแบบส่วนต่อขยายคุณสมบัติของภาษาเอกซ์เอตีแอล 2.0 มีการอ้างอิงแนวคิดจากภาษาเอสโอเอเอ็มแอล (SOAML) และดับบลิวเอสดีแอล (WSDL: Web Service. Description Language) โดยครอบคลุมคุณสมบัติดังต่อไปนี้
 - การมีส่วนร่วมระหว่างบริการ (Participates) โดยจะอธิบายได้ในส่วนของการระบุบทบาทของบริการที่มีความสัมพันธ์กันอยู่ในโครงสร้าง ว่าเป็นผู้ร้องขอ หรือผู้ให้บริการ
 - อินเทอร์เน็ตเฟสบริการ (Service interfaces) โดยจะสามารถอธิบายได้ในส่วนของการกำหนดคุณสมบัติของบริการว่าประกอบด้วยโอเปอร์เรชันใดบ้าง เช่นเดียวกับการประกาศอิลิเมนต์ portType และ อิลิเมนต์ operation ในมาตรฐานดับบลิวเอสดีแอล
 - ข้อมูลของบริการ (Service data) โดยสามารถอธิบายได้ในการกำหนดรูปแบบของข้อมูลที่ใช้เป็นส่วนเข้า (Input) และส่วนออก (Output) ของข้อมูล เช่นเดียวกับการประกาศอิลิเมนต์ Types และ อิลิเมนต์ Message ในมาตรฐานดับบลิวเอสดีแอล

- 2) การออกแบบส่วนต่อขยาย เพื่อเพิ่มความสามารถในการนำเสนอและอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการเท่านั้น
- 3) ผลที่ได้จากการออกแบบจะได้เอกซ์เอ็มแอลสคีมาใหม่ที่ได้มาจากการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ที่มีการเพิ่มการอธิบายสถาปัตยกรรมเชิงบริการ
- 4) พัฒนาเครื่องมือที่ใช้ในการประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอตีแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการได้แล้ว โดยจะมีสามารถในการวิเคราะห์และรายงานผลโครงสร้างของสถาปัตยกรรมเชิงบริการ โดยการนำโครงสร้างสถาปัตยกรรมจากกรณีตัวอย่างมาใช้ในการทดสอบเครื่องมือ เพื่อให้สามารถนำไปใช้ในการอธิบายคุณสมบัติของบริการที่อยู่บนโครงสร้างได้ชัดเจนมากยิ่งขึ้น ซึ่งจะออกมาอยู่ในรูปแบบของรายงานดังนี้
- จำนวนบริการ และการเชื่อมต่อที่อยู่บนโครงสร้างทั้งหมด โดยตรวจสอบจากจำนวนของอินเตอร์เฟซที่มีการระบุว่าเป็นบริการ และเส้นความสัมพันธ์ที่เชื่อมโยงอยู่ระหว่างในแต่ละบริการ
 - จำนวนการเข้าเรียกใช้บริการในแต่ละอินเตอร์เฟซ (Number of fan-in of an interface) โดยตรวจสอบจากจำนวนลิงก์ที่มีการเชื่อมต่อเข้าในแต่ละอินเตอร์เฟซของโครงสร้าง
 - รายการบริการที่มีการใช้งานในเซอร์วิสผู้ให้บริการ (Service Provided) โดยสามารถตรวจสอบได้จากการระบุคุณสมบัติของบริการว่าประกอบด้วยโอเปอเรชันใดบ้าง จากส่วนที่อธิบายไว้ในส่วนต่อขยาย
 - ความเข้ากันได้ของบริการที่เชื่อมโยงกัน โดยตรวจสอบจากการระบุความสัมพันธ์ระหว่างบริการในแต่ละบริการว่าบริการที่เป็นผู้ร้องขอ ได้ร้องขอถูกต้องเป็นไปตามสิ่งที่ให้บริการกำหนดไว้หรือไม่

1.4. ขั้นตอนการวิจัย

- 1) ศึกษาและทำความเข้าใจทฤษฎีของสถาปัตยกรรมเชิงบริการ และปัจจัยที่เกี่ยวข้องกับการออกแบบสถาปัตยกรรมเชิงบริการ
- 2) ศึกษาและวิเคราะห์ปัญหาของภาษาอธิบายสถาปัตยกรรม และงานวิจัยที่เกี่ยวข้องกับการพัฒนาภาษาอธิบายสถาปัตยกรรม
- 3) ศึกษาแนวทางในการปรับขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0
- 4) ออกแบบโครงสร้างของส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ให้มีความสามารถในการอธิบายสถาปัตยกรรมเชิงบริการ
- 5) พัฒนาส่วนต่อขยายของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ให้มีความสามารถในการอธิบายสถาปัตยกรรมเชิงบริการ และตรวจสอบความถูกต้องของภาษาอธิบายสถาปัตยกรรมหลังจากพัฒนาแล้ว
- 6) พัฒนาเครื่องมือในการตรวจสอบและวิเคราะห์ โครงสร้างของสถาปัตยกรรมเชิงบริการ ที่ใช้ภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ที่พัฒนาส่วนต่อขยายแล้ว ว่ามีความถูกต้องและเป็นไปตามปัจจัยในการออกแบบสถาปัตยกรรมเชิงบริการ
- 7) ทดสอบและประเมินผลวิธีวิจัยและระบบที่พัฒนาขึ้นเพื่อสนับสนุนแนวคิดในงานวิจัยนี้
- 8) ตีพิมพ์ผลงานทางวิชาการ
- 9) สรุปผลแนวทางการวิจัยและข้อเสนอแนะ และจัดทำวิทยานิพนธ์

1.5. ลำดับขั้นตอนการเสนอวิทยานิพนธ์นี้

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทดังนี้ บทที่ 1 เป็นบทนำซึ่งจะกล่าวถึงความ เป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย ขอบเขตของงานวิจัย ประโยชน์ที่ คาดว่าจะได้รับ วิธีการดำเนินการวิจัย ลำดับขั้นตอนการเสนอผลการวิจัย บทที่ 2 กล่าวถึงทฤษฎี พื้นฐาน และงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิดและวิธีการพัฒนาส่วนต่อขยาย ภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล ให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ ได้ บทที่ 4 กล่าวถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ และโครงสร้างของเครื่องมือ บทที่ 5 กล่าวถึงขั้นตอนในการทดสอบเครื่องมือ จากกรณีตัวอย่างที่ใช้ในการทดสอบ และบทที่ 6 เป็น ข้อเสนอสรุปและข้อเสนอแนะจากการวิจัย

1.6. ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความวิจัยเรื่อง “An xADL Extension for Service Oriented Architecture Design” โดย ภาสกร ปานนอก และ วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ The 4th International Conference on Information Science and Applications (ICISA2013) ระหว่างวันที่ 24-26 มิถุนายน 2556 ณ เมืองพัททยา จังหวัดชลบุรี ประเทศไทย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์นี้ประกอบด้วย สถาปัตยกรรมซอฟต์แวร์ ภาษาอธิบายสถาปัตยกรรม ภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดไอแอล 2.0 สถาปัตยกรรมเชิงบริการ ภาษาเอสไอเอเอ็มแอล ภาษาเอกซ์เอ็มแอล เอกซ์เอ็มแอลสคีม่า และงานวิจัยที่เกี่ยวข้อง ทั้ง 4 งานวิจัยมีรายละเอียดดังนี้

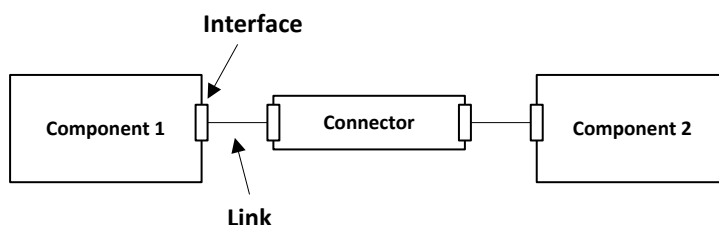
2.1. ทฤษฎีที่เกี่ยวข้อง

2.1.1. สถาปัตยกรรมซอฟต์แวร์ (Software Architecture)

สถาปัตยกรรมซอฟต์แวร์ [3] เป็นแบบจำลองเชิงนามธรรมในการอธิบายโครงสร้างของระบบซอฟต์แวร์ที่พัฒนา เป็นปัจจัยหนึ่งที่สำคัญ ที่จะช่วยในการกำหนด ความสามารถ คุณภาพ ความปลอดภัย และต้นทุน ของการพัฒนาซอฟต์แวร์ที่มีขนาดใหญ่และซับซ้อน การกำหนดสถาปัตยกรรมของซอฟต์แวร์ให้ชัดเจนอย่างเป็นขั้นตอนจะช่วยให้ การนำซอฟต์แวร์ไปพัฒนานั้น เป็นได้ไปได้ง่าย และสามารถนำไปใช้ในการวิเคราะห์ได้ โดยที่รายละเอียดทางสถาปัตยกรรมถูกใช้ประโยชน์ได้หลายอย่างเช่น การอธิบายระบบซอฟต์แวร์หรือการอธิบายการเปลี่ยนแปลงของระบบซอฟต์แวร์ การสื่อสารกับบุคคลที่เกี่ยวข้องกับการผลิตซอฟต์แวร์ การวางแผนการจัดการ และการดำเนินการพัฒนาซอฟต์แวร์ การพิสูจน์ความสอดคล้อง ของการอิมพลีเมนต์ระบบซอฟต์แวร์ เทียบกับข้อกำหนดทางสถาปัตยกรรม เป็นต้น

องค์ประกอบของสถาปัตยกรรมซอฟต์แวร์ สามารถแทนนามธรรม ที่อยู่ในรูปของข้อกำหนด สำหรับที่ใช้ในการแสดง โครงสร้าง (Structure) พฤติกรรม (Behavior) และคุณสมบัติหลักๆในแต่ละด้านของซอฟต์แวร์ เพื่ออธิบายสิ่งที่ซับซ้อนที่เกี่ยวกับการสร้างซอฟต์แวร์ให้อยู่ในรูปที่สามารถเข้าใจได้ [3]

- **โครงสร้าง** จะจำแนกออกเป็น คอมโพเนนต์ (Component) คอนเนคเตอร์ (Connector) และคอนฟิกูเรชัน (Configuration) แสดงในรูปที่ 2.1 ซึ่งในแต่ละองค์ประกอบมีบทบาทหน้าที่ดังนี้
 - คอมโพเนนต์แทนหน่วยการทำงานทางธุรกิจโดยตรงของระบบงานหนึ่ง
 - คอนเนคเตอร์แทนกฎการปฏิสัมพันธ์ต่างๆในแต่ละคอมโพเนนต์
 - คอนฟิกูเรชันแทนการกำหนดรูปแบบหรืออธิบายการเชื่อมต่อด้วยอินเตอร์เฟซ (Interface) กับลิงก์ (Link) ของคอมโพเนนต์ และคอนเนคเตอร์เป็นโครงสร้างสถาปัตยกรรมซอฟต์แวร์



รูปที่ 2.1 โครงสร้างนามธรรมของโครงสร้างสถาปัตยกรรม [3]

- **พฤติกรรม** อธิบายคุณสมบัติเชิงพฤติกรรมในระบบซอฟต์แวร์ที่ต้องการและถูกกำหนดเป็นข้อตกลงที่ชัดเจน เช่น โพรโตคอลการสื่อสาร (Communication Protocol) การประสานการทำงานร่วมกัน (Synchronization) หรือวิธีการเข้าถึงข้อมูล ที่เกิดขึ้นระหว่างคอมโพเนนต์
- **คุณสมบัติ** อธิบายถึงคุณสมบัติด้านที่ไม่ใช่ฟังก์ชันงานโดยตรงของระบบซอฟต์แวร์ เช่น ประสิทธิภาพของระบบ เป็นต้น

2.1.2. ภาษาอธิบายสถาปัตยกรรม (Architecture Description Language) [4]

จากที่กล่าวมาข้างต้น การออกแบบสถาปัตยกรรมซอฟต์แวร์ เป็นส่วนสำคัญส่วนหนึ่งที่จะช่วยให้มองเห็นถึงภาพรวมของระบบที่พัฒนาในมุมมองเชิงนามธรรมได้ชัดเจนมากยิ่งขึ้น ซึ่งในการออกแบบสถาปัตยกรรมซอฟต์แวร์นั้น จึงต้องใช้ภาษาอธิบายสถาปัตยกรรม เพื่อให้สามารถอธิบายส่วนประกอบ คอมโพเนนต์ คอนเนคเตอร์ และการเชื่อมต่อกับของแต่ละองค์ประกอบในสถาปัตยกรรมที่ออกแบบได้ โดยจำลองให้อยู่ในรูปของสัญลักษณ์ และไวยากรณ์ เพื่อสร้างความหมาย

อย่างเป็นทางการแบบแผน จากที่ไม่เป็นระเบียบ เพื่อสร้างความเข้าใจ และการสื่อสารที่ตรงกันของสถาปัตยกรรมซอฟต์แวร์ที่ออกแบบขึ้น กับผู้พัฒนาซอฟต์แวร์อื่นๆ

ภาษาอธิบายสถาปัตยกรรมเป็นเพียงภาษาที่จำลองภาพรวมในระดับนามธรรมของสถาปัตยกรรมที่สร้างขึ้นเท่านั้น ไม่ได้เป็นภาษาที่ใช้การในการเขียนโปรแกรม ภาษาระดับสูงที่ใช้ในการออกแบบ แบบจำลองเชิงวัตถุ (Object-oriented modeling) หรือ ภาษาข้อกำหนดเชิงรูปนัย (Formal specification language) [4]

องค์ประกอบของภาษาอธิบายสถาปัตยกรรมใดๆ จะต้องมียุคประกอบ คอมโพเนนท์ คอนเนคเตอร์ ที่ประกอบไปด้วยคุณลักษณะ ด้านอินเทอร์เฟซ และชนิด (Type) ของทั้งสองตามลำดับ และคอนฟิกรูชัน เพื่อการอธิบายสถาปัตยกรรมซอฟต์แวร์ได้อย่างชัดเจน ไม่กำกวมต่อการแปลความหมายเป็นอย่างอื่น และสนับสนุนคุณสมบัติด้านการนำเสนอและการวิเคราะห์สถาปัตยกรรม ซึ่งจะอธิบายรายละเอียดขององค์ประกอบแต่ละส่วนของภาษาอธิบายสถาปัตยกรรมได้ดังนี้ [3]

- **คอมโพเนนท์** คือหน่วยการประมวลผลใดๆของระบบ ซึ่งอาจจะเป็นหน่วยเล็กๆ ในลักษณะของขั้นตอนการทำงาน (Procedure) หรืออาจจะใหญ่เท่ากับหน่วยของแอปพลิเคชัน (Application) ใดๆ สามารถทำงานสำเร็จได้ด้วยตัวเอง โดยที่แต่ละคอมโพเนนท์อาจจะมีข้อมูล หรือพื้นที่หน่วยความจำในการปฏิบัติการของตัวเอง หรืออาจจะใช้ร่วมกับคอมโพเนนท์อื่นๆก็ได้ คุณลักษณะที่สำคัญของคอมโพเนนท์ที่ต้องมีก็คือ อินเทอร์เฟซ หรือที่เพิ่มเติมมากกว่านั้นหรือชนิด ความหมาย (Semantic) ข้อจำกัด (Constraint) การเปลี่ยนแปลงเชิงพัฒนาการ (Evolution) คุณสมบัติที่ไม่ใช่ฟังก์ชันการทำงานของระบบโดยตรง (Non-functional Properties)
- **อินเทอร์เฟซ** คือสิ่งที่แสดงข้อกำหนดในการปฏิสัมพันธ์และข้อจำกัดในการใช้งานระหว่าง คอมโพเนนท์และคอนเนคเตอร์ หรือ คอนเนคเตอร์และคอมโพเนนท์ เช่น การระบุการบริการ หรือความต้องการของบริการต่างๆที่อยู่ในรูปของข้อความ (Message) การปฏิบัติการ (Operation) หรือ ตัวแปร (Variable)
- **คอนเนคเตอร์** ทางสถาปัตยกรรมคอนเนคเตอร์ คือกล่องที่ใช้สำหรับจำลองการปฏิสัมพันธ์ระหว่างแต่ละคอมโพเนนท์ และกฎที่ใช้ในการควบคุมการปฏิสัมพันธ์ ความ

แตกต่างจากคอมโพเนนท์คือ คอนเนคเตอร์อาจจะไม่ใช่องค์ประกอบหลักที่ใช้พิมพ์ลิเมนต์ซอฟต์แวร์ คอนเนคเตอร์อาจจะอยู่ในลักษณะของ คำสั่งที่ทำหน้าที่เป็นตัวเชื่อมตัวแปรต่างๆในระบบที่ใช้ร่วมกัน ตารางงานบัฟเฟอร์ โครงสร้างข้อมูลแบบพลวัต (Dynamic Data Structure) ลำดับการเรียกการบวนการทำงาน (Procedure) ที่ฝังอยู่ในรหัสคำสั่ง (Code)

- **คอนฟิกูเรชัน** คอนฟิกูเรชันจำแนกออกเป็น 2 บทบาทที่สำคัญในทางสถาปัตยกรรม คือ การอธิบายลักษณะโครงสร้างสถาปัตยกรรมโดยการระบุการเชื่อมต่อกันระหว่างคอมโพเนนท์กับคอนเนคเตอร์ เพื่อประกอบเป็นโครงสร้างสถาปัตยกรรมซอฟต์แวร์ และการสนับสนุนการนำไปสู่การวิเคราะห์คุณสมบัติทางสถาปัตยกรรมโดยใช้ข้อมูลที่มีความหมาย (Semantic data) ที่ถูกติดต่อกับคอมโพเนนท์หรือคอนเนคเตอร์ที่มาดำเนินการร่วมกันผ่านการอ้างอิงการเชื่อมต่อระหว่างกันระหว่างคอมโพเนนท์กับคอมโพเนนท์ หรือคอนเนคเตอร์กับคอนเนคเตอร์ จากความเข้ากันได้ของอินเตอร์เฟซ คอนเนคเตอร์มีการสื่อสารที่ถูกต้อง หรือผลลัพธ์จากการดำเนินการทางแบบจำลองที่มีความหมายที่ให้ผลสอดคล้องกับพฤติกรรมที่ต้องการ เพื่อการพิสูจน์หรือสรุปผลคุณสมบัติที่มีผลต่อการทำงานของระบบ

2.1.3. ภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล 2.0 (xADL 2.0) [5]

ภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล 2.0 เป็นภาษาที่ใช้ในการอธิบายมุมมองเชิงนามธรรมของสถาปัตยกรรมซอฟต์แวร์ ซึ่งเป็นภาษาที่นำเสนอโครงสร้างของสถาปัตยกรรมด้วยภาษาเอกซ์เอ็มแอล ด้วยเป้าหมายของการสร้างเพื่อต้องการแก้ไขปัญหาที่ในภาษาอธิบายสถาปัตยกรรมในแต่ละภาษาสามารถอธิบายคุณลักษณะทางสถาปัตยกรรมเฉพาะด้านหนึ่งด้วยแบบจำลองเชิงความหมาย (Semantic model) ที่เฉพาะงานในลักษณะนั้นเท่านั้น [3] ทำให้เวลาที่ต้องการอธิบายหรือจัดการกับคุณลักษณะด้านอื่นๆของระบบงานหนึ่งต้องมีการเปลี่ยนแปลงไปมาระหว่างภาษา ซึ่งทำให้เกิดความยุ่งยากในการทำงาน ภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล 2.0 จึงเป็นภาษาที่มีความเรียบง่าย มีความครอบคลุมตามข้อกำหนดของภาษาอธิบายสถาปัตยกรรมโดยทั่วไป แต่สามารถที่จะพัฒนาส่วนต่อขยายของภาษาในส่วนของแบบจำลองเชิงความหมาย เพื่อสนับสนุนในงานด้านใดด้านหนึ่งเองได้

ความสามารถในการอธิบายสถาปัตยกรรม ของภาษาอธิบายสถาปัตยกรรมสามารถอธิบายได้
ใน 4 กลุ่มลักษณะได้แก่

- **กลุ่ม xArch - Run-time (Architectural Instances Core)** เป็นส่วนประกอบหลักที่ใช้ในการอธิบายโครงสร้างของสถาปัตยกรรม ได้แก่ คอมโพเนนท์ อินเตอร์เฟซ คอนเนคเตอร์ ลิงก์ ซึ่งเป็นส่วนประกอบที่ภาษาอธิบายสถาปัตยกรรม โดยทั่วไปจะต้องมี
- **กลุ่ม Architecture structure and types** ทำหน้าที่ในการอธิบายรายละเอียดของโครงสร้างสถาปัตยกรรม การอธิบายองค์ประกอบสถาปัตยกรรมในลักษณะ อินสแตนซ์ และการประกาศเป็นชนิด (Type) องค์ประกอบของสถาปัตยกรรม
- **กลุ่ม Production families** ทำหน้าที่อธิบายรายละเอียดที่สอดคล้องต่อการผลิตซอฟต์แวร์ เช่น เวอร์ชันของซอฟต์แวร์, การเปลี่ยนแปลงของซอฟต์แวร์
- **กลุ่ม Implementation mappings** ทำหน้าที่ในการอ้างอิงองค์ประกอบสถาปัตยกรรมระดับนามธรรมไปสู่สถาปัตยกรรมระดับรูปธรรมที่เป็นส่วนของโปรแกรมคำสั่งของการอิมพลิเมนต์ซอฟต์แวร์

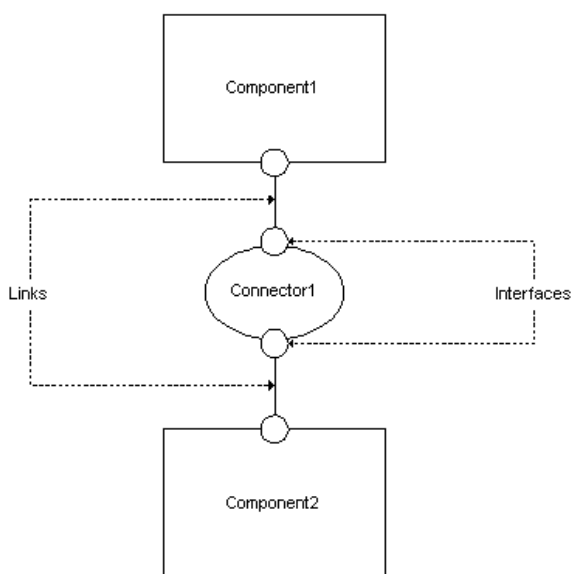
องค์ประกอบภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล 2.0 ประกอบไปด้วย 3 ส่วน
คือ

2.1.3.1. เอกซ์เอดีแอล อินสแตนซ์ (xADL Instance)

เป็นการอธิบายสถาปัตยกรรมซอฟต์แวร์ในมุมมอง ณ ขณะเวลาระบบซอฟต์แวร์กำลังทำงาน โดยที่อธิบายรายละเอียดสถาปัตยกรรมที่เป็น อินสแตนซ์จะกำหนดอยู่ในไฟล์เอกซ์เอ็มแอลสคิม่าที่ชื่อ instance.xsd แสดงในรูปที่ 2.2 โดยมีองค์ประกอบที่ถูกอ้างถึงในไวยากรณ์ (Syntax) และความหมาย (Semantic) โดยมีโครงสร้างหลักๆคือ

- คอมโพเนนท์อินสแตนซ์ (Component Instances) รวมไปถึง สถาปัตยกรรมย่อย (Sub-architecture)

- คอนเนคเตอร์อินสแตนซ์ (Connector Instances) รวมไปถึง สถาปัตยกรรมย่อย (Sub-architecture)
- อินเตอร์เฟซอินสแตนซ์ (Interface Instances)
- ลิงก์อินสแตนซ์ (Link Instances)
- กลุ่ม (General Groups)



รูปที่ 2.2 โครงสร้างการเชื่อมต่อระหว่างอินเตอร์เฟซผ่านลิงก์ [3]

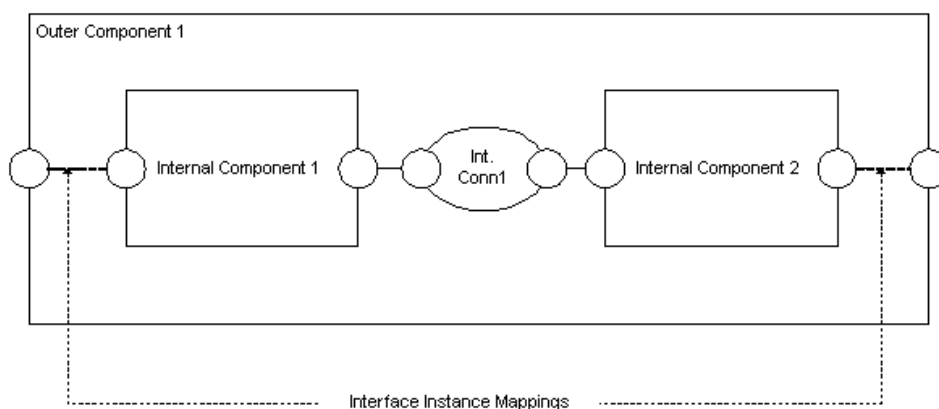
คอมโพเนนท์ หรือ คอนเนคเตอร์อินสแตนซ์ใดๆ จะประกอบไปด้วยอินเตอร์เฟซอินสแตนซ์ที่ประกอบอยู่ในตัว หน้าที่ของอินเตอร์เฟซอินสแตนซ์ คือทางผ่านสำหรับข้อมูลใดที่ผ่านไประหว่างคู่การอ้างอิงปฏิสัมพันธ์

ลิงก์อินสแตนซ์ใช้สำหรับการเชื่อมต่อระหว่าง อินเตอร์เฟซอินสแตนซ์อิมพลิเมนต์ จากเอกซ์เอ็มแอลลิงก์ (XML Link) การเชื่อมต่อระหว่างอินเตอร์เฟซใดๆ โดยผ่านลิงก์อินสแตนซ์อยู่ภายใต้ข้อจำกัดของอินเตอร์เฟซดังนี้คือ ต้องเป็นอินเตอร์เฟซที่มีทิศทางตรงกันข้ามเท่านั้น เช่น ระหว่าง in กับ out ของทิศทางของอินเตอร์เฟซเพื่อบ่งชี้ความหมายในการนำพาข้อมูล

ลิ่งก์อินสแตนซ์ ไม่มีทิศทางการไหลของข้อมูลความหมายของทิศทางการไหลของข้อมูลระดับนามธรรมตามความหมายของทิศทางที่ระบุไว้ที่อินเทอร์เฟซอินสแตนซ์ และ ลิ่งก์อินสแตนซ์ไม่ควรที่จะเพิ่มเติมหรือขยายความหมายในลิ่งก์อินสแตนซ์ใดๆ

การอธิบายสถาปัตยกรรมที่องค์ประกอบสถาปัตยกรรมอยู่ในลักษณะที่ซับซ้อน จะมีการกำหนดลักษณะ สถาปัตยกรรมย่อย ซึ่งเป็นหนึ่งในการอิมพลีเมนต์คุณลักษณะ การประกอบกัน (Compositionality) และความไม่เป็นเนื้อเดียว (Heterogeneity) ที่ภายในองค์ประกอบ คอมโพเนนท์หรือคอนเนคเตอร์ ประกอบด้วยองค์ประกอบย่อยเหล่านี้อยู่ในตัวของตัวเอง แสดงในรูปแบบที่ 2.3 และจะต้องเป็นไปตามข้อกำหนดดังต่อไปนี้

- ทุกองค์ประกอบภายใน หน่วยที่เป็นโครงสร้างสถาปัตยกรรมย่อย (Super-Architecture) จะเชื่อมต่อรายละเอียดทั้งหมดที่ชัดเจนทุกองค์ประกอบ
- การเชื่อมต่อระหว่างหน่วยที่เป็นแบบสถาปัตยกรรมย่อย จะเชื่อมต่อกันผ่าน Interface instance mapping



รูปที่ 2.3 การเชื่อมต่อของโครงสร้างสถาปัตยกรรมย่อย [5]

2.1.3.2. การออกแบบโครงสร้างของสถาปัตยกรรม (Modeling Architectural Structure)

โดยปกติแล้วงานในระดับสถาปัตยกรรมจะดำเนินการอยู่ในกิจกรรมการออกแบบเป็นส่วนใหญ่ แม้ว่าภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดไอแอล 2.0 จะจัดส่วนของการประกาศองค์ประกอบสถาปัตยกรรมแบบอินสแตนซ์ แต่อย่างไรก็ตามองค์ประกอบสถาปัตยกรรมแบบอินสแตนซ์จะใช้ใน

การจำลองสถาปัตยกรรมในระดับของการปฏิบัติงาน ดังนั้นการจำลององค์ประกอบสถาปัตยกรรม ในขณะการออกแบบ ที่มองเสมือนว่าในแต่ละองค์ประกอบย่อย คอมโพเนนท์ คอนเนคเตอร์ เป็นเสมือนกับอินสแตนซ์ จะให้การประกาศภายใต้การควบคุมของโครงสร้างทางสถาปัตยกรรม (Architectural Structure) ซึ่งถูกกำหนดไว้ในสคิม่าโครงสร้างและชนิด ในไฟล์ type.xsd ซึ่งประกอบด้วยองค์ประกอบดังนี้

- คอมโพเนนท์
- คอนเนคเตอร์
- อินเตอร์เฟส
- ลิงก์
- กลุ่ม

องค์ประกอบข้างต้นเหล่านี้ใช้ในการอธิบายแทนทุกองค์ประกอบอินสแตนซ์ ดังนั้นข้อกำหนดการประกาศจะเป็นในลักษณะเช่นเดียวกับที่ระบุในองค์ประกอบของอินสแตนซ์

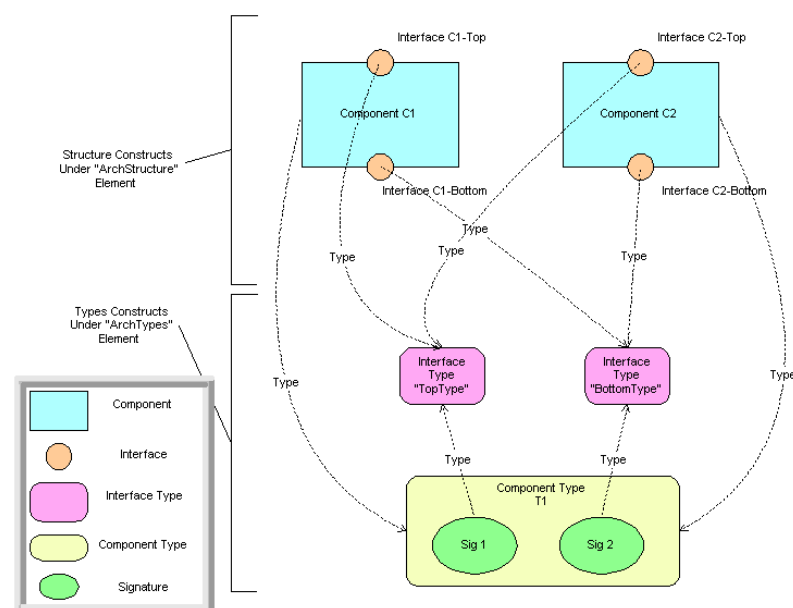
2.1.3.3. การกำหนดชนิดของแบบจำลอง (Type of Model)

ชนิดของแบบจำลองของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ได้ประยุกต์มาจากหลายแบบจำลองที่พบในภาษาที่ใช้ในการเขียนโปรแกรม หรือการเขียนโค้ดโปรแกรมที่ต้องมีหลายองค์ประกอบภายในสถาปัตยกรรมซอฟต์แวร์ หรือโค้ดของโปรแกรมที่มีพฤติกรรมการทำงานที่เหมือนกัน ดังนั้นแนวคิดการแบ่งปัน (Share) พฤติกรรม ระดับนามธรรมหรือองค์ประกอบทางอิมพลีเมนต์ เช่น คลาสในแนวคิดเชิงวัตถุ (Object-Oriented) เพื่อที่จะไม่ต้องเขียนองค์ประกอบเหล่านี้ซ้ำซ้อน โดยนิยามให้องค์ประกอบเหล่านี้ต้องมีการใช้งานอยู่ในแนวคิดของการประกาศชนิด ในระดับนามธรรม

องค์ประกอบในชนิดของแบบจำลอง ของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ประกอบด้วย ชนิดของคอมโพเนนท์ (Component Type) ชนิดของคอนเนคเตอร์ (Connector Type) ชนิดของอินเตอร์เฟส (Interface Type) ที่แยกออกเป็นแต่ละองค์ประกอบอย่างชัดเจน ซึ่งจะ

กำหนดไว้ในเอกซ์เอ็มแอลสคีมาไฟล์ชื่อ type.xsd ซึ่งทุกองค์ประกอบจะอยู่ภายใต้โวลีเมนต์ (Element) ที่ชื่อว่า ArchTypes

การอธิบายของชนิดของแบบจำลองใดๆ จะเป็นอิสระ ไม่สามารถประกอบรวมจากชนิดของแบบจำลองอื่นได้ ในรูปที่ 2.4 แสดงแบบจำลองที่ประกาศชนิดของคอมโพเนนท์ หรือคอนเนคเตอร์ จะสัมพันธ์กับชนิดของอินเตอร์เฟซ และจะมีลายเซ็น (Signature) เป็นองค์ประกอบ โดยที่ภายในลายเซ็นจะมีชนิดที่กำหนดเป็นโวลีเมนต์เป็นองค์ประกอบ ซึ่งจะกำหนดอยู่ในสคีมาไฟล์ชื่อ type.xsd และข้อกำหนดในภาษาอธิบายสถาปัตยกรรมใดๆ จะต้องมีว่าองค์ประกอบคอมโพเนนท์ หรือคอนเนคเตอร์ จะต้องมียูนิคไนด์เป็นส่วนหนึ่งขององค์ประกอบนั้น ซึ่งจากการที่ชนิดของคอมโพเนนท์หรือคอนเนคเตอร์จะมีองค์ประกอบที่สมบูรณ์ตามข้อกำหนดของภาษาอธิบายสถาปัตยกรรมในภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็มแอล 2.0 ทำให้คุณสมบัตินี้เป็นจริงที่ต้องประกอบชนิดอินเตอร์เฟซ เข้ากับคอมโพเนนท์หรือคอนเนคเตอร์ จากการอ้างอิงให้ความหมายโดยใช้ชนิดของโวลีเมนต์ที่อยู่ภายในลายเซ็นเป็นตัวชี้เชื่อมโยงไปยังชนิดของอินเตอร์เฟซ เพื่อระบุเป็นความหมายว่าชนิดคอมโพเนนท์ หรือคอนเนคเตอร์อะไร เป็นต้น ซึ่งลักษณะแบบจำลองความหมายนี้ เหมือนกับในแบบจำลองเชิงวัตถุ ที่มีการประกาศอินเตอร์เฟซ และมีการอิมพลิเมนต์ในอินเตอร์เฟซ ด้วยคลาสใดๆที่อิมพลิเมนต์อินเตอร์เฟซนั้น



รูปที่ 2.4 เมตาโมเดลการอ้างอิงชนิดของแบบจำลอง [5]

2.1.3.4. ความหมายของการประกาศโครงสร้างสถาปัตยกรรม (Semantics of Modeling Architectural Structure)

จะยึดตามความหมายในการประกาศโครงสร้างแบบอินสแตนซ์ ส่วนที่แตกต่างคือการประกาศสถาปัตยกรรมย่อย (Sub-Architecture) โดยที่ในอินสแตนซ์ โมเดลอธิบายแบบในบรรทัด (in-line) ซึ่งอิลิเมนต์ย่อยของสถาปัตยกรรมย่อยจะอยู่ภายใต้อินสแตนซ์ คอมโพเนนต์ หรือคอนเนคเตอร์ แต่ในขณะที่โครงสร้างประกาศสถาปัตยกรรมย่อยผ่าน ชนิดของคอมโพเนนต์ และชนิดของคอนเนคเตอร์ โดยความหมายของการประกาศชนิดของสถาปัตยกรรมย่อย ตามรูปที่ [6] เป็นการเชื่อมโยงระหว่างอินเตอร์เฟซขององค์ประกอบย่อยที่อยู่ภายในองค์ประกอบหลักที่เป็นคอมโพเนนต์ กับ ลายเซ็นขององค์ประกอบหลักที่อยู่ภายนอก ซึ่งจะต้องเป็นไปตามข้อจำกัดดังนี้คือ ชนิดของอินเตอร์เฟซ ที่องค์ประกอบภายนอกสามารถอ้างอิงถึงได้ต้องเป็นชนิดเดียวกับชนิดของอินเตอร์เฟซขององค์ประกอบย่อยที่อ้างอิงกันแบบ Signature-Interface Mapping

2.1.4. สถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture: SOA) [6]

สถาปัตยกรรมเชิงบริการหรือเอสโอเอ เป็นรูปแบบสถาปัตยกรรมแบบหนึ่งที่เกิดจากการนำซอฟต์แวร์หลายตัว ซึ่งในแต่ละชิ้นจะเรียนกันว่าเซอร์วิส (Service) หรือบริการ ที่มีอาจจะมี ความแตกต่างแพลตฟอร์ม หรือไม่แตกต่างกัน เซอร์วิสมีการให้บริการที่แตกต่างกันมารวมกัน การติดต่อกันระหว่างเซอร์วิสอาจเพื่อการส่งผ่านข้อมูลระหว่างกันตามธรรมดา หรือเป็นการประสานกันระหว่างหลายๆ เซอร์วิสในการทำงานหรือทำกิจกรรมบางอย่างร่วมกัน สถาปัตยกรรมเชิงบริการจะมีความสัมพันธ์เชื่อมโยงอย่างหลวม ๆ (Loose Coupling) ระหว่างเซอร์วิส เพื่อใช้ในการสร้างแอปพลิเคชัน (Applications) และสนับสนุนการนำเซอร์วิสที่มีอยู่แล้วไปใช้ซ้ำ (Reuse) ในหลาย ๆ แอปพลิเคชัน และถอดออกได้อย่างง่าย การเชื่อมต่อระหว่างเซอร์วิสเพื่อให้ติดต่อสื่อสารกันได้นั้นทำได้โดยเทคโนโลยีการเชื่อมต่อที่เป็นมาตรฐาน เพื่อให้บริการที่เข้ามาเชื่อมต่อกันสามารถเข้าใจความหมายและใช้ข้อมูลร่วมกันได้ เช่น เว็บเซอร์วิส (Web Services) ลักษณะเด่นอีกอย่างหนึ่งของสถาปัตยกรรมเชิงบริการก็คือ การสนับสนุนการจัดการกระบวนการทาง ธุรกิจ (Business Process) อย่างอัตโนมัติ กระบวนการทางธุรกิจเกิดขึ้นจากการเรียกใช้เซอร์วิส และการประสานการทำงานระหว่างเซอร์วิสเป็นลำดับตามเงื่อนไขหรือตรรกะทางธุรกิจ (Business Logic) ที่กำหนดขึ้นในรูปของ กระแสงาน (Workflow) เพื่อให้บรรลุการทำงานตามขั้นตอนของธุรกิจ ตัวอย่างเช่น เมื่อมีลูกค้า

เตรียมเดินทางไปต่างประเทศ การประมวลผลของธุรกิจจัดการการท่องเที่ยวจะเป็นไปตามขั้นตอนอันประกอบด้วย การบันทึกคำขอของลูกค้า การติดต่อสายการบินการติดต่อโรงแรม การติดต่อรถเช่า การตรวจสอบเครดิตของลูกค้าและตัดบัญชีบัตรเครดิต และการจัดส่งตั๋วเครื่องบินพร้อมทั้งแผนการเดินทางและใบจองต่าง ๆ ลำดับขั้นตอนเหล่านี้สามารถดำเนินการโดยการเรียกใช้บริการของเซอร์วิสต่าง ๆ ที่เกี่ยวข้อง รวมทั้งสามารถควบคุมกระบวนการได้ [6]

2.1.4.1. คุณลักษณะที่สำคัญของสถาปัตยกรรมเชิงบริการ

ใน [6] ได้กำหนดคุณลักษณะที่สำคัญของสถาปัตยกรรมเชิงบริการไว้ดังนี้

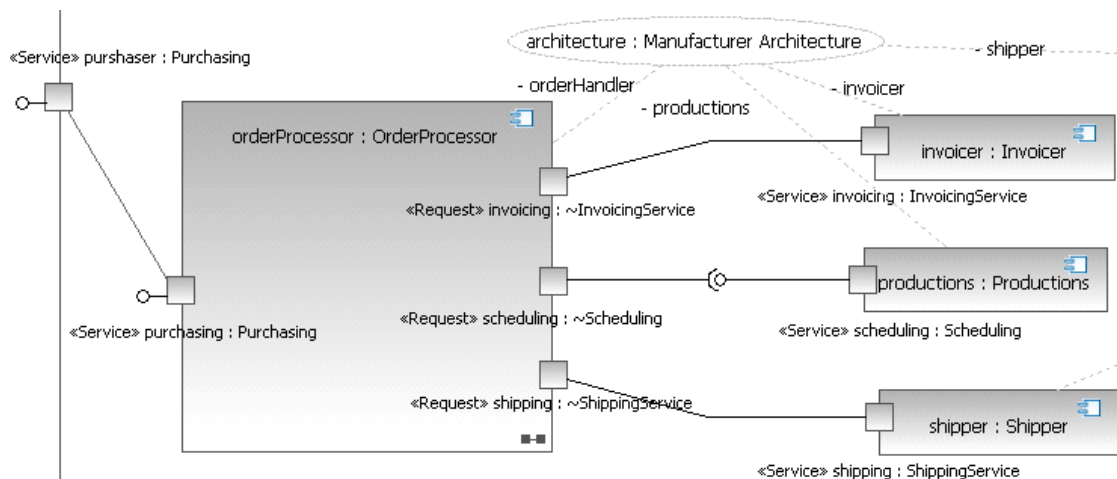
- การติดต่อสื่อสารระหว่างบริการ (Service) จะใช้เทคโนโลยีเอกซ์เอ็มแอล (XML) ในการสื่อสารร่วมกัน ซึ่งไม่ขึ้นกับแพลตฟอร์มและเทคโนโลยีใดๆ ทำให้สามารถเรียกใช้งานได้โดยไม่ต้องคำนึงถึงความไม่เข้ากันระหว่างบริการ
- ในแต่ละบริการจะมีอินเตอร์เฟซ (Interface) ที่อธิบายบริการ เช่น Service Name, Input Parameter, Output Parameter และข้อมูลอื่นๆ ในรูปแบบของไฟล์เอกซ์เอ็มแอล (XML) ทำให้ไม่ขึ้นกับแพลตฟอร์มและเทคโนโลยีที่เซอร์วิสนั้นใช้อยู่ โดยมากมักจะใช้มาตรฐานดับบลิวเอสดีแอล (Web Service Description Language) ในการอธิบายสิ่งที่บริการจะมีให้
- โปรแกรมประยุกต์ (Application) หรือกระบวนการทางธุรกิจต่างๆ สามารถพัฒนาขึ้นมาจากการใช้บริการเดิมที่มีอยู่ ซึ่งมาตรฐานที่นิยมใช้คือ WS-BPEL (Web Service Business Process Execution Language)
- เซอร์วิสแต่ละตัวจะมีส่วนการควบคุมคุณรูปที่เป็น QoS (Quality of Service) อาทิเช่น การควบคุมความปลอดภัยด้าน การยืนยันตัวตน (Authentication), การอนุญาตการเข้าถึง (Authorization), ข้อมูลที่เชื่อถือได้ (Reliable Message) และ นโยบายหรือ บทบาทการใช้งาน (Policy)

2.1.5. ภาษาเอสโอเอเอ็มแอล (SoaML) [7]

ภาษาเอสโอเอเอ็มแอลเป็นภาษาที่ในกำหนดรายละเอียดโครงสร้างของสถาปัตยกรรมเชิงบริการ ซึ่งพัฒนาโดยกลุ่ม OMG (Object Management Group) โดยการกำหนดรายละเอียดด้วยแผนภาพยูเอ็มแอล (UML) และ เมตาโมเดล (Meta model) ในการออกแบบ ด้วยการขยายความสามารถของแผนภาพยูเอ็มแอล เพื่อรองรับการต้องการในการสร้างแบบจำลองสำหรับสถาปัตยกรรมเชิงบริการ รวมทั้งข้อกำหนดของการให้บริการ การเชื่อมต่อระหว่างบริการ และการใช้งานบริการ เพื่อมองให้เห็นส่วนประกอบของบริการที่ทำงานร่วมกันตามเป้าหมายทางธุรกิจ โดยภาษาเอสโอเอเอ็มแอล ได้ขยายความสามารถของภาษายูเอ็มแอล ไว้ได้แก่

- 1) การมีส่วนร่วมระหว่างบริการ (Participants) ใช้ในการกำหนดว่าระหว่างผู้ให้บริการ กับ ผู้ใช้บริการ ในระบบ เมื่อบริการอยู่ในลักษณะของผู้ให้บริการจะกำหนดว่าเป็นพอร์ตบริการ (Service Port) แต่เมื่อเป็นผู้ใช้บริการจะกำหนดว่าเป็นพอร์ตร้องขอ (Request ports)
- 2) อินเทอร์เฟซของบริการ (Service interfaces) ใช้ในการอธิบายสิ่งที่ผู้บริกรมให้ และ ข้อตกลงในการร้องขอ หรือ การให้บริการ
- 3) ข้อตกลงการให้บริการ (Service contracts) ใช้สำหรับเป็นการกำหนดข้อตกลงระหว่าง บริการที่มีความสัมพันธ์ร่วมกัน
- 4) สถาปัตยกรรมบริการ (Service architectures) อธิบายถึงภาพรวมการทำงานร่วมกันของบริการที่มีความสัมพันธ์ร่วมกัน
- 5) ข้อมูลของบริการ (Service data) ใช้ในการอธิบายถึงข้อมูล ที่ข้อความที่แนบไป ในการสื่อสารกันระหว่างผู้ให้บริการและผู้รับบริการ

จากรูปที่ 2.5 เป็นภาพตัวอย่างการใช้ภาษาเอสโอเอ็มแอลในการออกแบบโครงสร้างสถาปัตยกรรมเชิงบริการ ของระบบสั่งซื้อสินค้า ซึ่งมีการเรียกใช้บริการที่มากจากระบบที่ทำงานแตกต่างกันออกไป



รูปที่ 2.5 ตัวอย่างการใช้ภาษาเอสโอเอ็มแอล [2]

สำหรับงานวิจัยนี้ ผู้วิจัยได้นำแนวคิดลักษณะองค์ประกอบของโครงสร้างในการออกแบบสถาปัตยกรรมบริการด้วยภาษาเอสโอเอ็มแอล มาใช้กับงานวิจัยการออกแบบส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอ็กเอตีแอล

2.1.6. ภาษาดับบลิวเอสดีแอล (Web Services Description Language - WSDL)

เป็นมาตรฐานสำหรับสื่อสารแอปพลิเคชันที่ต้องการเรียกใช้ เพื่ออธิบายรายละเอียดของบริการ การแลกเปลี่ยนข้อมูล ของเว็บเซอร์วิส โดยถูกกำหนดด้วยรูปแบบไวยากรณ์เอกซ์เอ็มแอล (XML) ภายในเอกสารดับบลิวเอสดีแอลจะมีการนิยามคุณสมบัติของบริการ รูปแบบข้อมูลที่ใช้ในการติดต่อของบริการ ลักษณะการเชื่อมต่อ ที่อยู่ที่ใช้ในการเชื่อมต่อบริการ ซึ่งภายในเอกสารดับบลิวเอสดีแอล จะประกอบไปด้วยส่วนประกอบที่ในการนิยามคุณสมบัติของบริการดังนี้ [8]

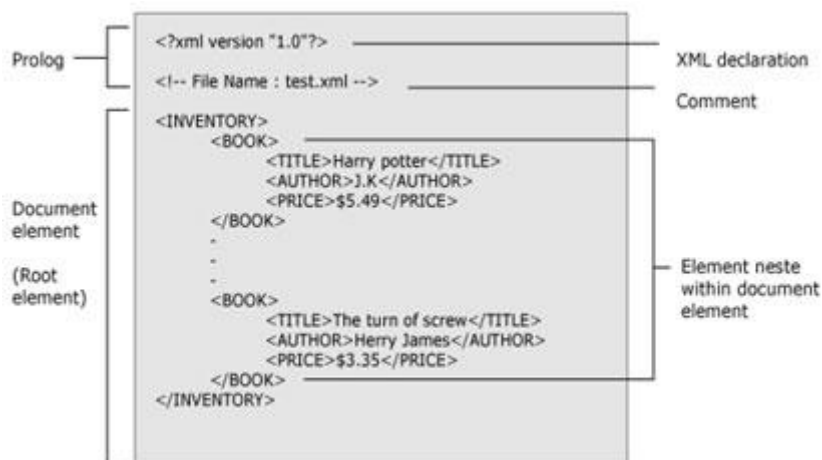
- 1) ส่วนประกอบ Types สำหรับนิยามชนิดของข้อมูลของแต่ละเมสเสจ (Message) ที่ใช้แลกเปลี่ยนระหว่างกัน ด้วยเอกซ์เอ็มแอลสคีมา (XML Schema)
- 2) ส่วนประกอบ Message สำหรับนิยามข้อมูลที่มีการสื่อสารกันในแต่ละโอเปอร์เรชัน (Operation)
- 3) ส่วนประกอบ Operation สำหรับนิยามการดำเนินการที่อยู่ภายในบริการ

- 4) ส่วนประกอบ Port Type สำหรับนิยามกลุ่มของโอเปอเรชันที่อยู่ภายในบริการ
- 5) ส่วนประกอบ Binding สำหรับนิยามที่อยู่ที่ใช้ในการติดต่อกับโอเปอเรชันที่อยู่ภายในบริการ
- 6) ส่วนประกอบ Service สำหรับนิยามรูปแบบโปรโตคอล ที่ใช้ในการติดต่อกับบริการ

สำหรับงานวิจัยนี้ ผู้วิจัยได้นำแนวคิดของส่วนประกอบที่ใช้ในการนิยามคุณสมบัติของบริการในภาษาดัชนีเว็บเอสดีแอล มาใช้กับงานวิจัยการออกแบบส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอ็กซีแอลดีแอล

2.1.7. ภาษาเอกซ์เอ็มแอล (eXtensible Markup Language-XML) [9]

เป็นรูปแบบของข้อความอย่างง่าย ๆ ที่แสดงอยู่ในรูปแบบของโครงสร้าง ซึ่งมาจากรูปแบบมาตรฐานเก่าที่เรียกว่า ภาษาเอสจีเอ็มแอล (SGML) เพื่อให้มีความเหมาะสมกับการใช้งานบนเว็บซึ่งต้องทำงานร่วมกับภาษาเฮชทีเอ็มแอล (HTML) นอกจากนี้ภาษาเอกซ์เอ็มแอลยังมีการใช้งานอย่างแพร่หลาย เหมาะสำหรับเป็นภาษากลางใช้ในการแลกเปลี่ยนสารสนเทศระหว่าง โปรแกรม, คน, คอมพิวเตอร์ หรือภายในองค์กรหรือระหว่างองค์กร ซึ่งทำให้สามารถนำข้อมูลจากหลายๆแหล่งมาทำการประมวลผลและแสดงผลร่วมกันได้ องค์ประกอบของภาษาเอกซ์เอ็มแอลเป็นภาษาที่มีการอธิบายข้อมูลเป็นลำดับชั้น (Hierarchy) อยู่ภายในรูปแบบของป้ายข้อความ (Tag) และมีคุณสมบัติที่สามารถอธิบายตัวเองได้ (Self-describing) และมีความยืดหยุ่นไม่ว่าจะเป็นเอกสารเอกซ์เอ็มแอลใดๆ ก็สามารถใช้เครื่องมือเอกซ์เอ็มแอลตัวใดก็ได้มาอ่านหรือประมวลผลได้เช่นเดียวกัน และมีตัวแปลภาษา (XML Parser) ที่ทำให้รูปแบบของภาษาเอกซ์เอ็มแอลสามารถอ่านออกได้ ทำให้ลดข้อจำกัดของระบบคอมพิวเตอร์ที่มีการทำงานแตกต่างกัน หรือส่วนแสดงผลที่มีความแตกต่างกันได้ แต่ทั้งนี้การสร้างเอกสารเอ็มแอลจะต้องมีโครงสร้างที่ถูกต้องตามหลักไวยากรณ์ของภาษาเอกซ์เอ็มแอลจึงจะสามารถแลกเปลี่ยนข้อมูลระหว่างกันได้ ซึ่งทำให้ต้องมีภาษาที่ใช้คอยควบคุมความถูกต้องของเอกสารไว้ ซึ่งใช้กันอย่างแพร่หลาย เช่น DTDs (Document Type Definitions), XSD (XML Schema Definitions) จากรูปที่ 2.6 เป็นตัวอย่างโครงสร้างของเอกสารเอ็กซีเอ็มแอล ซึ่งจะประกอบไปด้วยอิลิเมนต์ต่างๆที่มีข้อมูลอยู่ภายใน

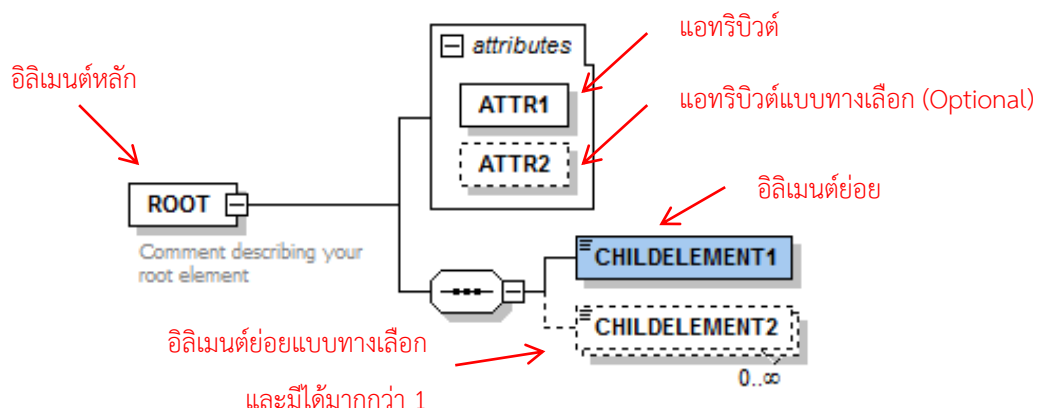


รูปที่ 2.6 โครงสร้างของเอกสารภาษาเอกซ์เอ็มแอล [10]

2.1.8. เอกซ์เอ็มแอลสคีมา (XML Schema)

คือรูปแบบของโครงสร้างที่ใช้ในการกำหนดลักษณะ โครงสร้าง และรายละเอียดของเอกสารเอกซ์เอ็มแอลว่าควรจะต้องมีโครงสร้างเป็นอย่างไร โดยลักษณะโครงสร้างของเอกซ์เอ็มแอลสคีมา จะมีหน้าตาเหมือนกับเอกสารเอกซ์เอ็มแอล ซึ่งจะเป็นตัวกำหนดรายการของส่วนประกอบ อิลิเมนต์ (Element) ในเอกสาร และแอทริบิวต์ (Attribute) ต่างๆที่อยู่ในอิลิเมนต์นั้นๆ ชนิดของข้อมูล ตำแหน่งของอิลิเมนต์ และ เป็นส่วนที่คนและเครื่องมือต่างๆสามารถอ่านออกได้ ถือเป็นคำอธิบายที่เป็นกฎเกณฑ์ของเอกสารเอกซ์เอ็มแอลต่างๆ [11]

สำหรับงานวิจัยนี้จะใช้ภาษาเอกซ์เอ็มแอล ซึ่งเป็นพื้นฐานของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล มาใช้ในการออกแบบสถาปัตยกรรมซอฟต์แวร์ และสร้างเอกซ์เอ็มแอลสคีมาด้วย ภาษาเอกซ์เอสดี (XSD) มาใช้ในการออกแบบโครงสร้างของส่วนต่อขยาย เพื่อใช้ในการควบคุมความถูกต้องของโครงสร้างเอกสารส่วนต่อขยายสถาปัตยกรรมเอกซ์เอตีแอลได้ โดยใช้เครื่องมือที่ชื่อว่า XMLSpy XML Editor [12] มาเป็นเครื่องมือที่ช่วยในการออกแบบ แก้ไข ตรวจสอบความถูกต้องของเอกซ์เอ็มแอลสคีมา และยังสามารถสร้างภาพแบบจำลองโครงสร้างเอกซ์เอ็มแอลสคีมา ซึ่งจะจำลองโครงสร้างที่ประกอบไปด้วย อิลิเมนต์ แอทริบิวต์ และข้อกำหนดต่างๆของเอกสารเอกซ์เอ็มแอลดังในรูปที่ 2.7



รูปที่ 2.7 สร้างภาพแบบจำลองโครงสร้างเอกสารเอกซ์เอ็มแอลสตีมา ด้วยเครื่องมือ XMLSpy XML Editor

2.1.9. เอกซ์ลิงก์ (XLink) [13]

เอกซ์ลิงก์ เป็นองค์ประกอบหนึ่งในภาษาเอกซ์เอ็มแอล ที่จะแทรกลงไป inline ของเอกสารเอกซ์เอ็มแอล เพื่อสร้างและอธิบายการเชื่อมโยงระหว่างเอกสารเอกซ์เอ็มแอลหรือเอกสารอื่นๆ ที่อยู่ภายนอก (External) คล้ายคลึงกับลิงก์ในภาษาเอกซ์เอ็มแอล แต่มีความสามารถที่มากกว่าคือ สามารถทำลิงก์เชื่อมโยงไปยังทรัพยากรได้มากกว่าหนึ่ง และมีความซับซ้อนกว่าเพราะต้องประกาศอิลิเมนต์และแอทริบิวต์ทั้งหมดก่อน โดยเอกซ์ลิงก์สามารถกำหนดลักษณะได้ 2 แบบ คือ เอกซ์ลิงก์แบบพื้นฐาน (Simple) และเอกซ์ลิงก์แบบขยาย (Extended) ในรูปที่ 2.8 ตัวอย่างการสร้างเอกซ์ลิงก์แบบพื้นฐานได้มีการกำหนดค่า แอททริบิวต์ XML:LINK ตัวนี้เป็นค่า simple และ อิลิเมนต์ที่เป็นตัวลิงก์ทุกตัวในเอกซ์ลิงก์จะต้องมี แอททริบิวต์ซึ่งเป็นส่วนที่ใช้ในการเชื่อมโยงไปยังทรัพยากรอื่น คือแอททริบิวต์ HREF ซึ่งทำงานเหมือนกับ แอททริบิวต์ตัวหนึ่งในภาษาเอกซ์เอ็มแอล ส่วนเอกซ์ลิงก์แบบขยาย เป็นการสร้างเอกซ์ลิงก์เพื่อใช้ในการเชื่อมโยงไปยังเชื่อมโยงไปยังทรัพยากรได้มากกว่าหนึ่ง

```
<MYSLINK XML:LINK="simple" HREF="http://google.com">Google Search</MYSLNK>
```

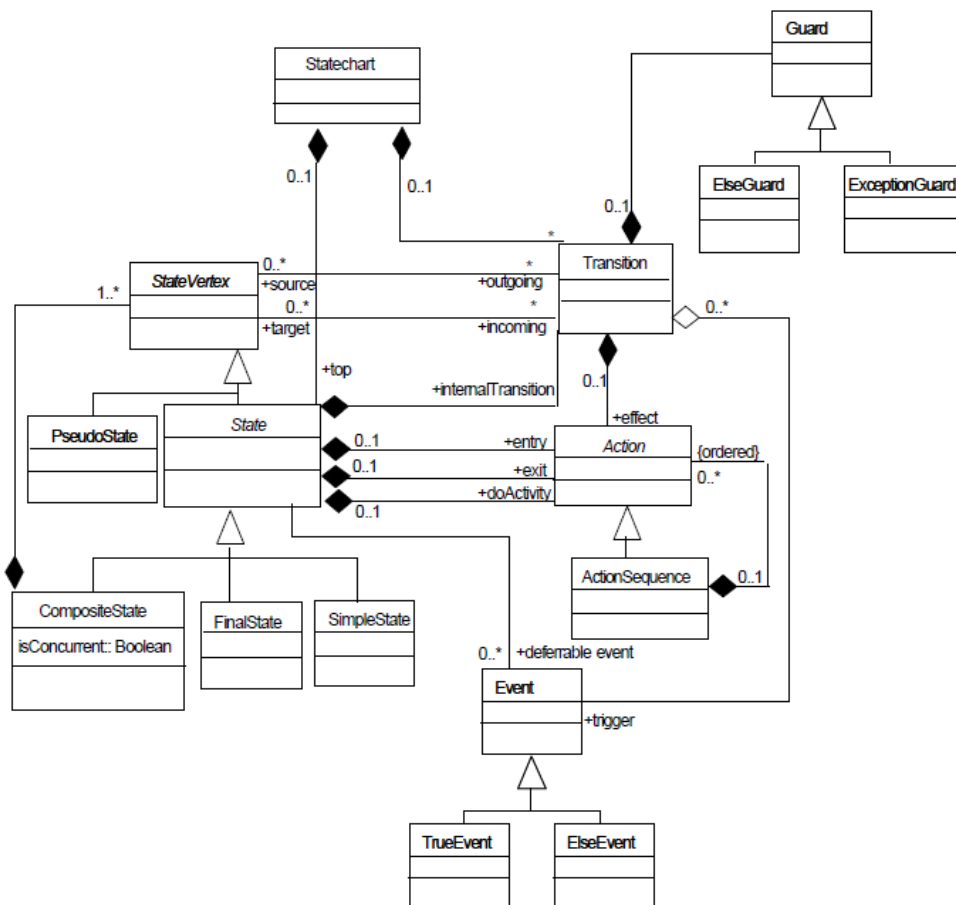
รูปที่ 2.8 ตัวอย่างการสร้างเอกซ์ลิงก์แบบพื้นฐาน

ประโยชน์ของการใช้เอกซ์ลิงก์คือ เพื่อช่วยลดความซับซ้อนของลำดับชั้นภายในเอกสารเอกซ์เอ็มแอลลง และให้ไปเพิ่มเติมการอธิบายไปยังเอกสารอื่นที่เกี่ยวข้องได้ ทำาง่ายต่อการอ่านเอกสารเอกซ์เอ็มแอล

2.2. งานวิจัยที่เกี่ยวข้อง

2.2.1. Extending xADL with Statechart Behavioral Specification [14]

งานวิจัยนี้นำเสนอโดย L. Naslavsky และคณะ ในปี ค.ศ. 2007 โดยกล่าวถึง วิธีการพัฒนา ส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล กับแผนภาพเชิงพฤติกรรม เพื่อให้มีความสามารถในการอธิบายพฤติกรรมของคอมโพเนนต์ได้ และ สนับสนุนเครื่องมือการวิเคราะห์ ทดสอบ โครงสร้าง และพฤติกรรมของสถาปัตยกรรม [14] โดยทำการระบุแนวคิดของโมเดลแผนภาพเชิงพฤติกรรม ในรูปแบบของคลาสไดอะแกรม และสร้างแบบจำลองเป็นรูปแบบภาษาเอกซ์เอ็มแอล ที่ต้องเป็นไปตามกฎเกณฑ์เอกซ์เอ็มแอลสคีมาของแผนภาพเชิงพฤติกรรม แสดงในรูปที่ 2.10 โดยอ้างอิงจากแผนภาพยูเอ็มแอล ตาม Meta Model แสดงในรูปที่ 2.9 จนได้แนวคิดของโมเดลองค์ประกอบของแผนภาพเชิงพฤติกรรม



รูปที่ 2.9 แนวคิดของโมเดลองค์ประกอบของแผนภาพเชิงพฤติกรรม [14]

จากนั้นนำโมเดลแนวคิดที่ได้นำมาสร้างเอกซ์เอ็มแอลสคีมา ของแผนภาพเชิงพฤติกรรมเพื่อนำไปใช้เป็นส่วนต่อขยายคอมโพเนนท์ของภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดดีแอล เพื่อให้โครงสร้างของภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดดีแอล สามารถอธิบายแผนภาพเชิงพฤติกรรม และสามารถนำไปวิเคราะห์จากเครื่องมือที่พัฒนาจากการอ่านโครงสร้างเอกซ์เอ็มแอล ของภาพได้ง่าย

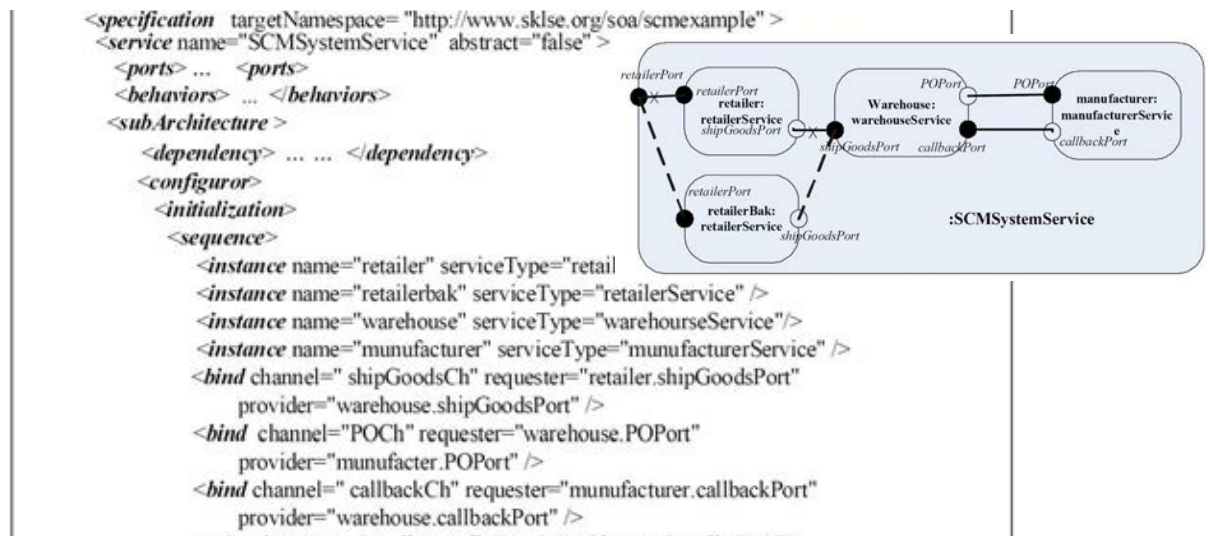
```
<xs:element name = "StateChart">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="transitions" type="TransitionType"
        maxOccurs="unbounded"/>
      <xs:element name="top" type="StateType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

รูปที่ 2.10 เอกซ์เอ็มแอลสคีมา ของส่วนต่อขยายแผนภาพเชิงพฤติกรรมในภาษา xADL [14]

โดยแนวความคิดที่น่าสนใจนำมาใช้ คือ การออกแบบโมเดลแนวคิดของโครงสร้างเพื่อนำมาสร้างนำมาสร้างเอกซ์เอ็มแอลสคีมา ซึ่งทำให้ง่ายต่อการนำไปสร้างเป็นส่วนต่อขยายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดดีแอลต่อไป

2.2.2. A New Architecture Description Language for Service-Oriented Architecture [15]

งานวิจัยนี้นำเสนอโดย X. Jia และคณะ ในปี ค.ศ. 2007 โดยกล่าวถึง ภาษาอธิบายสำหรับสถาปัตยกรรมเชิงบริการใหม่ ชื่อว่า SOADL ซึ่งความสามารถในการอธิบาย โครงสร้าง, ส่วนต่อประสาน, การเชื่อมต่อ, พฤติกรรม, คุณสมบัติในเชิงคุณภาพของสถาปัตยกรรมของบริการ [15] โดยมุ่งเน้นการออกแบบระบบที่มุ่งเน้นการให้บริการให้มีความคล่องตัวทางธุรกิจ และง่ายต่อการนำโครงสร้างของสถาปัตยกรรมที่ออกแบบไปใช้ โดยตัวภาษาใหม่ในงานวิจัยได้นำเอกซ์เอ็มแอล มาเป็นเครื่องมือในการอธิบายสถาปัตยกรรม แสดงในรูปที่ 2.11 ซึ่งเป็นภาษาที่มีโครงสร้างที่ดี มีความสามารถในการขยายความสามารถ และมีเครื่องมือสนับสนุนมากมาย อีกทั้งภาษา SOADL ยังมีความสามารถในการวิเคราะห์ความเข้ากันได้ของอินเตอร์เฟซ ความถูกต้องขององค์ประกอบ และความสามารถในการระบุรายละเอียดของบริการเช่น BPEL เป็นต้น

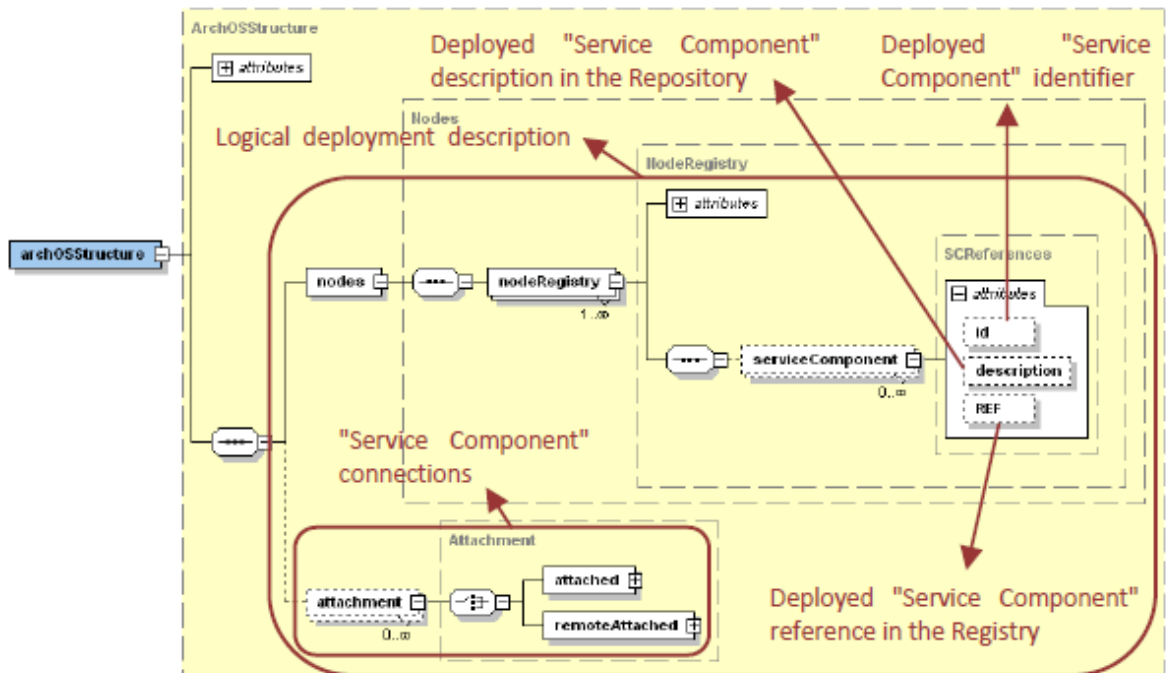


รูปที่ 2.11 โครงสร้างของภาษา SOADL [15]

งานวิจัยนี้เป็นงานวิจัยที่นำเสนอภาษาอธิบายสถาปัตยกรรมรูปแบบหนึ่ง ที่ใช้ในการอธิบายรูปแบบสถาปัตยกรรมเชิงบริการ ซึ่งเป็นแนวคิดหนึ่งที่น่าสนใจในการนำไปใช้ในงานวิจัย การออกแบบส่วนต่อขยายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดไอแอล

2.2.3. An xADL Extension for Managing Dynamic Deployment in Distributed Service Oriented Architectures [16]

งานวิจัยนี้นำเสนอโดย Mohamed Nadhmi และคณะ ในปี ค.ศ. 2010 ได้กล่าวถึง ภาษาอธิบายสถาปัตยกรรมใหม่ที่ชื่อว่า 3DxSoAdl ซึ่งพัฒนามาจากการเพิ่มส่วนต่อขยายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดไอแอล เพื่อใช้สำหรับอธิบายการจัดการการปรับใช้แบบไดนามิกในสถาปัตยกรรมเชิงบริการ ส่วนขยายที่พัฒนาขึ้นจะสามารถอธิบาย การใช้งานและกระบวนการบริหารจัดการสถาปัตยกรรมเชิงบริการ ซึ่งประกอบด้วย 3 ส่วนหลักได้แก่ ส่วนอธิบายบริการที่ถูกเก็บไว้ใช้งาน ส่วนอธิบายการกระจายของบริการ และส่วนอธิบายการจัดการแบบไดนามิกของพัฒนาสถาปัตยกรรม บนพื้นฐานของการนำกลับไปใช้ใหม่ (Redeployment) แสดงรูปที่ 2.12 แนวทางในการพัฒนาส่วนต่อขยายจะมุ่งเน้นในการพัฒนาส่วนต่อขยายที่ส่วนของคอมโพเนนท์ เพื่อให้สามารถอธิบายได้ในลักษณะของคอมโพเนนท์บริการ (Service Component)



รูปที่ 2.12 โครงสร้างเอกซ์เอ็มแอลสคีมาของส่วนต่อขยาย [16]

จากแนวความคิดพัฒนาภาษาอธิบายสถาปัตยกรรมในงานนี้ได้นำเสนอ มีการออกแบบส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมที่สามารถอธิบายสถาปัตยกรรมเชิงบริการได้อย่างครอบคลุม และยืดหยุ่น จึงเป็นตัวอย่างแนวคิดอีกรูปแบบหนึ่งในการพัฒนาส่วนต่อขยายภาษาเอกซ์เอ็มแอลให้มีความสามารถในการอธิบายรูปแบบของสถาปัตยกรรมเชิงบริการได้

2.2.4. Adding Aspects to xADL 2.0 for Software Product Line Architectures [17]

งานวิจัยนี้นำเสนอโดย Lidia Fuentes และ Nadia Gámez โดยกล่าวถึง การพัฒนาส่วนต่อขยายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็มแอล ให้มีความสามารถในการอธิบายคุณลักษณะของสถาปัตยกรรมสายผลิตภัณฑ์ ด้วยการกำหนดตัวแทนของคุณลักษณะตัวแปร Crosscutting และการอ้างอิงในสถาปัตยกรรมสายผลิตภัณฑ์เพื่อมุ่งเน้นในการวิเคราะห์คุณลักษณะ ของแนวความคิดพัฒนาซอฟต์แวร์ในรูปแบบของการออกแบบสถาปัตยกรรมเชิงลักษณะ (Aspect-oriented architecture)

บทที่ 3

การวิเคราะห์และวิธีการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล

ในบทนี้จะกล่าวถึงการวิเคราะห์และวิธีการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล ให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ โดยเริ่มต้นจากแสดงภาพรวมแนวคิดของการพัฒนาส่วนต่อขยาย และต่อด้วยการออกแบบและพัฒนาส่วนต่อขยาย

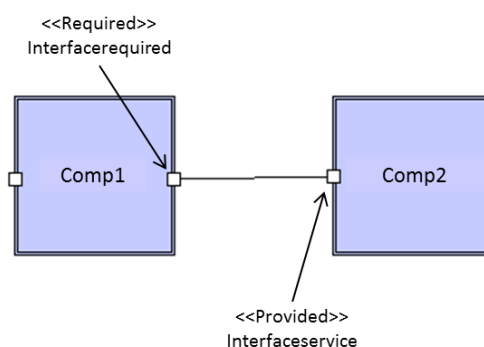
3.1. ภาพรวมและแนวคิดในการพัฒนาส่วนต่อขยาย

ผู้วิจัยได้ใช้ภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ในการพัฒนาส่วนต่อขยาย ซึ่งตัวภาษาดังเดิมนั้นเป็นภาษาที่ได้รับการออกแบบโดยมีจุดมุ่งหมายเพื่อให้สามารถนำไปใช้พัฒนาสำหรับการอธิบายในงานใดงานหนึ่งได้ และพื้นฐานของภาษามาจากภาษาเอกซ์เอ็มแอล ทำให้สามารถนำไปใช้ในการพัฒนาส่วนต่อขยายได้ วิธีการดำเนินวิจัยจึงได้แบ่งออกเป็น 3 ขั้นตอนดังต่อไปนี้

- 1) ทำการรวบรวมแนวคิดของสถาปัตยกรรมเชิงบริการ การออกแบบสถาปัตยกรรมเชิงบริการ และมาตรฐานต่างๆ ที่เกี่ยวข้องกับสถาปัตยกรรมบริการ
- 2) ศึกษาและค้นหาแนวทางการพัฒนาส่วนต่อขยายความสามารถในการอธิบายสถาปัตยกรรมเชิงบริการจากงานวิจัยที่เกี่ยวข้องกับการพัฒนาภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล
- 3) ออกแบบโครงสร้างส่วนต่อขยาย เพื่อให้ได้เอกซ์เอ็มแอลสคีมาของส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอลที่มีความสามารถในการอธิบายโครงสร้างสถาปัตยกรรมเชิงบริการได้

แนวคิดในการออกแบบส่วนขยายนั้น มีการรวบรวมแนวคิดจากภาษาเอสโอเอเอ็มแอล [7] และดับบลิวเอสดีแอล [8] เนื่องจากว่าดับบลิวเอสดีแอล เป็นมาตรฐานที่ใช้ในการอธิบายคุณสมบัติของบริการที่ใช้กันอย่างแพร่หลายในการออกแบบสถาปัตยกรรมอยู่แล้ว ทำให้ผู้ที่นำส่วนต่อขยายที่พัฒนาขึ้นไปใช้ สามารถเข้าใจคุณสมบัติของบริการ และสามารถนำไปใช้งานได้ง่ายขึ้น แล้วยังสามารถนำโครงสร้างส่วนต่อขยายไปสร้างเป็นดับบลิวเอสดีแอลสำหรับใช้งานจริงได้ในอนาคตอีกด้วย ซึ่งมีความแตกต่างกับงานวิจัย [16] และ [15] ที่จะมีการนิยามโครงสร้างการอธิบายคุณสมบัติของสถาปัตยกรรมเชิงบริการใหม่เลย แต่ในงานวิจัยนี้จะมีการออกแบบส่วนต่อขยายการอธิบาย

สถาปัตยกรรมเชิงบริการที่แตกต่างออกไปคือ มีการออกแบบส่วนต่อขยายการอธิบายสถาปัตยกรรมเชิงบริการที่ในอินเทอร์เน็ตเฟส ของคอมโพเนนต์ แสดงในรูปที่ 3.1 เพื่ออธิบายบริการที่มีให้ในแต่ละเซอร์วิสคอมโพเนนต์ ซึ่งจะมีการแสดงเห็นถึงบริการที่เรียกใช้ (Required) คำสั่งที่เรียกใช้ (Operator) และบริการที่ให้บริการ (Provided) โดยอ้างอิงจากดับบลิวเอสดีแอล ด้วยการลดทอนการอธิบายให้คงเหลือส่วนที่สำคัญเอาไว้



รูปที่ 3.1 อธิบายการเชื่อมต่อของบริการในอินเทอร์เน็ตเฟสของคอมโพเนนต์

โครงสร้างภายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 นั้น จะมีการเพิ่มเอกซ์ลิงก์ (XLink) อยู่ภายในอิลิเมนต์อินเทอร์เน็ตเฟส (Interface element) เพื่อเชื่อมโยงการอธิบายคุณสมบัติของบริการอยู่ภายนอกอิลิเมนต์อินเทอร์เน็ตเฟส และทำการอธิบายคุณสมบัติของบริการไว้ในอิลิเมนต์ใหม่ที่ออกแบบไว้ภายนอกโครงสร้างการออกแบบสถาปัตยกรรม ทำให้ไม่กระทบกับโครงสร้างเดิมของภาษาที่มีอยู่เดิม ช่วยให้แก้ไขปรับปรุงโครงสร้าง และการอธิบายคุณสมบัติของบริการเป็นไปได้อย่างสะดวก ซึ่งคุณสมบัติของบริการที่แยกออกมานั้นจะแบ่งออกเป็นสองส่วน ได้แก่ ผู้ให้บริการ และ ผู้ร้องขอบริการ ซึ่งแสดงให้เห็นถึงส่วนร่วมหว่างบริการและอินเทอร์เน็ตเฟสของบริการจากแนวคิดภาษาเอสโอเอเอ็มแอล

3.2. โครงสร้างของส่วนต่อขยาย

- 1) ส่วนของ ผู้ให้บริการ เป็นส่วนที่ใช้ในการนิยามคุณสมบัติหรือความสามารถของผู้ให้บริการ มีการอ้างอิงโครงสร้างมาจากดับบลิวเอสดีแอล โดยส่วนของผู้ให้บริการได้มีการลดทอนการอธิบายในดับบลิวเอสดีแอลที่มีอยู่เดิมให้เหลือส่วนที่สำคัญดังนี้

- Type ใช้ในการอธิบายประเภทของข้อมูล (Data type) และ โครงสร้างข้อมูล (Data Structure) ของข้อมูลเข้า (Input) และ ข้อมูลออก (output) ที่อยู่ภายใน ส่วนของผู้ให้บริการ ซึ่งแสดงให้เห็นถึงข้อมูลของบริการที่มีการรับส่งกันเป็นไปตาม แนวคิดภาษาเอสโอเอเอ็มแอล [7]
- Message ใช้ในการกำหนด ข้อมูลเข้า และข้อมูลออก ของแต่ละคำสั่ง ภายในส่วน ของผู้ให้บริการ
- PortType ภายในจะประกอบไปด้วย กลุ่มของคำสั่งจำนวนหลายๆคำสั่ง ที่ผู้ร้องขอ บริการจะสามารถเรียกใช้งานได้ ซึ่งจะมีการเชื่อมโยงกับ Message เพื่อกำหนด ข้อมูลเข้า ข้อมูลออก และ ประเภทของข้อมูล ต่อไป
- Binding เป็นการนิยามรูปแบบโปรโตคอล (Protocol) และการเข้ารหัสข้อมูล (Encoding) เพื่อใช้ในการเรียกในแต่ละคำสั่ง ซึ่งแสดงให้เห็นถึงรูปแบบการติดต่อ ของบริการเป็นไปตามแนวคิดภาษาเอสโอเอเอ็มแอล [7]
- Service เป็นส่วนที่กำหนดที่อยู่ของผู้ให้บริการ เพื่อให้ผู้ร้องขอบริการสามารถ เรียกใช้งานได้อย่างถูกต้อง

2) ส่วนของ ผู้ร้องขอบริการเป็นส่วนที่ใช้นิยาม หรือระบุคำสั่งที่ต้องการเรียกใช้จากผู้ให้บริการ

- OperationCall ใช้ในการกำหนดคำสั่ง ข้อมูลเข้า และ ข้อมูลออก ที่ผู้ร้องขอ บริการต้องการ ซึ่งคำสั่งจะต้องสอดคล้องกันกับ ส่วนที่ผู้ให้บริการระบุไว้

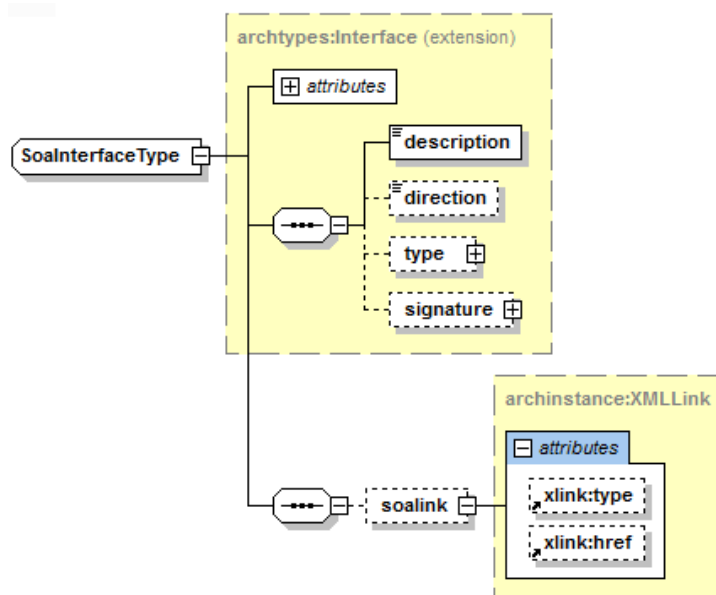
3.3. การพัฒนาส่วนต่อขยาย

จากแนวคิดการออกแบบส่วนต่อขยายในข้างต้น ที่มีการขยายความสามารถในการอธิบาย สถาปัตยกรรมเชิงบริการ ที่อินเทอร์เน็ตเฟสของคอมโพเนนท์ จึงต้องมีการเพิ่มไวยากรณ์ของภาษาอธิบาย สถาปัตยกรรมเอกซ์เอตีแอล 2.0 ที่ใช้ในการเชื่อมโยงการอธิบายคุณสมบัติของบริการอยู่ภายนอกอิลี เมนต์อินเทอร์เน็ตเฟส เพื่อให้ความง่ายต่อการแก้ไข การอ่านทำความเข้าใจ และคงไว้ซึ่ง Semantic ของภาษา อธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 จึงจำเป็นต้องเขียน เอกซ์เอ็มแอลสคีม่า 2 ส่วนหลักๆ ได้แก่

1) การพัฒนาส่วนต่อขยายภายในบรรทัดของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0

การพัฒนาส่วนต่อขยายภายในบรรทัดของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล เป็นส่วนการเชื่อมโยงการอธิบายคุณสมบัติของแต่ละบริการที่อยู่ภายนอกอินเทอร์เน็ตอินเทอร์เน็ต โดยทำการเขียน เอกซ์เอ็มแอลสคีมา เพื่อเพิ่มไวยากรณ์ที่ประกอบด้วยเอกซ์ลิงก์ อยู่ในอินเทอร์เน็ตอินเทอร์เน็ตของไวยากรณ์เดิมของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 เดิม โดยใช้คุณสมบัติการสืบทอดของ แบบจำลองเนื้อหา (Content Model) ของ เอกซ์เอ็มแอลสคีมา โดยสืบทอดจาก Schema: types.xsd ทำให้โครงสร้างที่ขยายเพิ่มเติม สามารถใช้งานได้กับตัวตรวจสอบความถูกต้องของไวยากรณ์เดิม โดยในรูปที่ 3.2 แสดงให้เห็นสคีมาส่วนต่อขยายอินเทอร์เน็ต soalink ด้วยเครื่องมือ XMLSpy สำหรับจำลองโครงสร้างของเอกซ์เอ็มแอลสคีมา และใน

รูปที่ 3.3 เป็นตัวอย่างการขยายเพิ่ม soalink ภายในอินเทอร์เน็ตอินเทอร์เน็ตเดิมของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ซึ่งจะทำหน้าที่เป็นตัวเชื่อมโยงการอธิบายคุณสมบัติของแต่ละบริการในไปภายนอกได้ ด้วยเอกซ์ลิงก์



รูปที่ 3.2 เอกซ์เอ็มแอลสคีมา การเพิ่มอินเทอร์เน็ตอินเทอร์เน็ต soalink ภายในอินเทอร์เน็ตอินเทอร์เน็ตเดิม

```

<!--Interface OrderList -->
<types:interface types:id="interfaceffa80107-92a077c4-49692544-c27602a9"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">OrderList</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <!--SOALink to OrderTakerService.xml#GetOrderService -->
  <soa:soalink xlink:href="OrderTakerService.xml#GetOrderService" xlink:type="simple"
xsi:type="instance:XMLLink"/>
</types:interface>

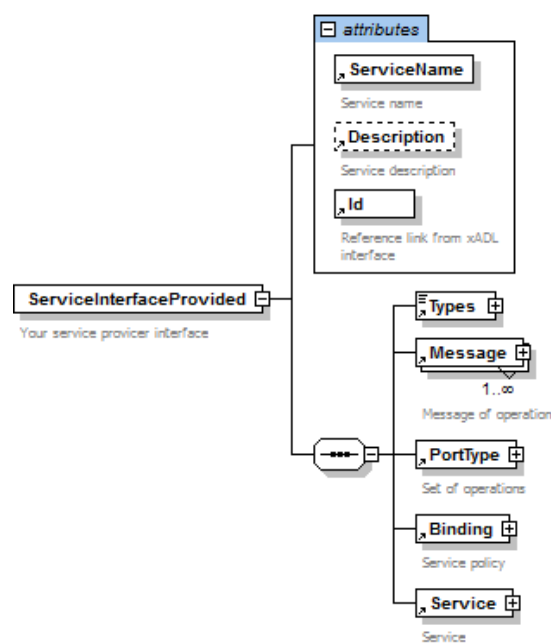
```

รูปที่ 3.3 ตัวอย่างการขยายเพิ่ม soalink ภายในอีลิเมนต์อินเทอร์เน็ตเฟส

2) การพัฒนาส่วนต่อขยายภายนอกเพื่ออธิบายคุณสมบัติของแต่ละบริการ

ในส่วนนี้จะพัฒนาส่วนที่ใช้ในการอธิบายคุณสมบัติของแต่ละบริการที่ได้เชื่อมโยงมาจากเอกซ์ลิงก์ จากภายในอินเทอร์เน็ตเฟสของโครงสร้างภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล ที่ได้พัฒนาส่วนต่อขยายภายในเรียบร้อยแล้ว โดยมีโครงสร้างอยู่ในรูปแบบของภาษาเอกซ์เอ็มแอล ซึ่งจะต้องมีการเขียน เอกซ์เอ็มแอลสคีมา ที่มีไวยากรณ์ที่ใช้ในการอธิบายคุณสมบัติของบริการซึ่งประกอบไปด้วย

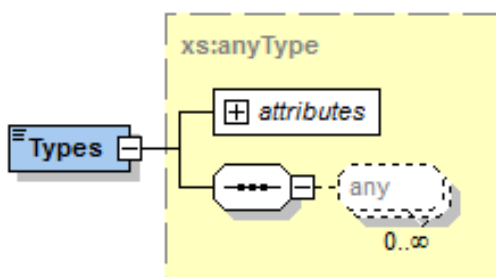
1) ServiceInterfaceProvided ทำหน้าที่นิยามคุณสมบัติของบริการว่าเป็นผู้ให้บริการ ซึ่งภายในจะประกอบไปด้วย Type, Message, PortType, Binding, Service ดังรูปที่ 3.4 แสดงให้เห็นถึงองค์ประกอบภายในที่มีการนิยามคุณสมบัติต่างๆของบริการ



รูปที่ 3.4 เอกซ์เอ็มแอลสคีมา โครงสร้างของ ServiceInterfaceProvided

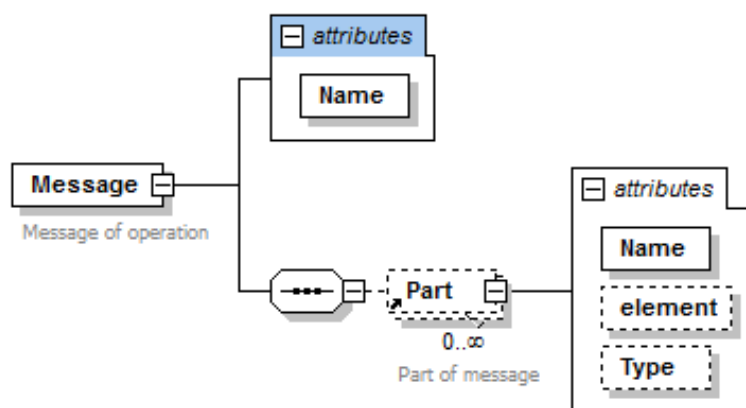
ซึ่งองค์ประกอบภายในของ ServiceInterfaceProvided ประกอบไปด้วยคุณสมบัติของผู้ให้บริการดังนี้

- Type ทำหน้าที่ในการอธิบายประเภทของข้อมูล และ โครงสร้างข้อมูล ทั้งข้อมูลเข้า และข้อมูลออก ที่ใช้ในระหว่างการรับส่งข้อมูลระหว่างกัน ซึ่งภายในสามารถอธิบายประเภทของข้อมูล และ โครงสร้างข้อมูลได้ในรูปแบบของ เอกซ์เอ็มแอลสคีมา ดังในรูปที่ 3.5



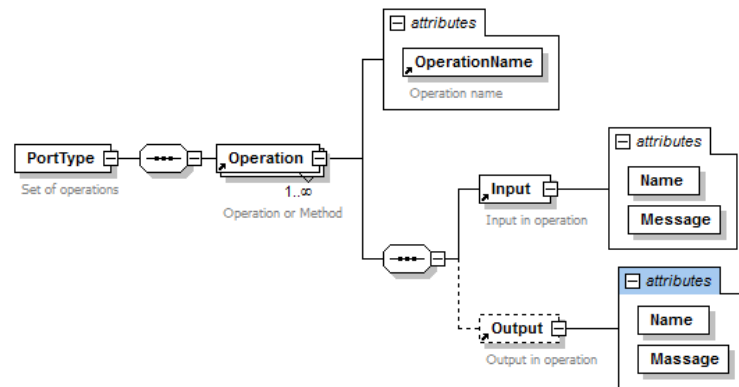
รูปที่ 3.5 แผนภาพ เอกซ์เอ็มแอลสคีมา โครงสร้างของ Type

- Message ทำหน้าที่ในการกำหนด ข้อมูลเข้า และข้อมูลออก ของแต่ละคำสั่ง ภายในส่วนของผู้ให้บริการ ซึ่งสามารถระบุจำนวน และประเภทของข้อมูลเข้าและข้อมูลออกได้ ดังในรูปที่ 3.6



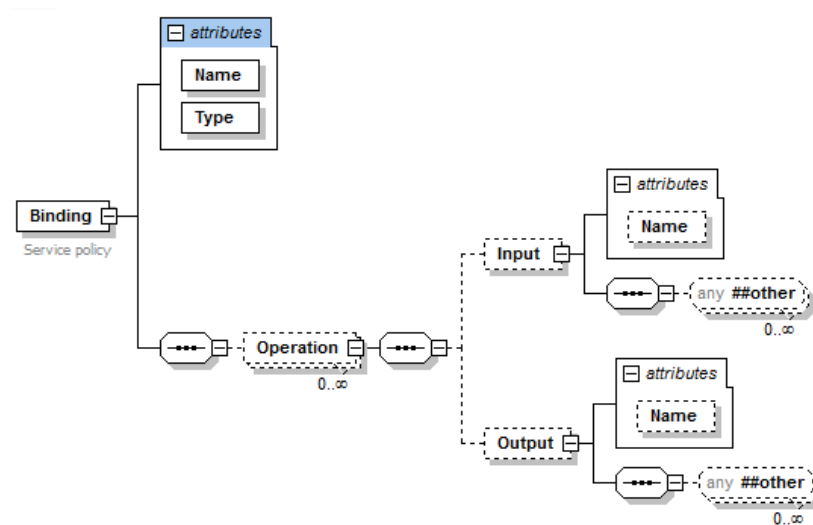
รูปที่ 3.6 เอกซ์เอ็มแอลสคีมา โครงสร้างของ Message

- PortType ทำหน้าที่อธิบายคำสั่งที่มีอยู่ภายในผู้ให้บริการ ซึ่งภายในจะประกอบไปด้วย กลุ่มของคำสั่ง จำนวนหลายคำสั่ง และจะมีการกำหนด ข้อมูลเข้า และข้อมูลออก ซึ่งจะมีการเชื่อมโยงไปยัง Message ดังในรูปที่ 3.7



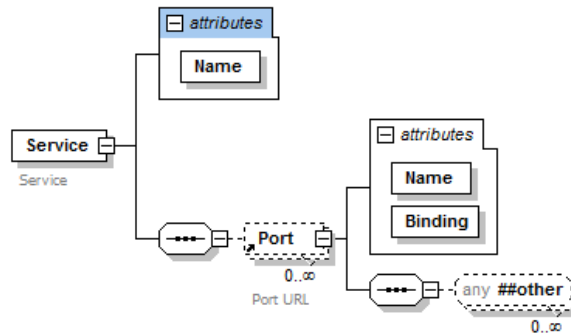
รูปที่ 3.7 แผนภาพ เอกซ์เอ็มแอลสคีมา โครงสร้างของ PortType

- Binding ทำหน้าที่ในการอธิบายรูปแบบโปรโตคอล และการเข้ารหัสข้อมูล เพื่อใช้ในการเรียกในแต่ละคำสั่ง ของผู้ให้บริการ ซึ่งจะสามารถระบุหรือไม่จำเป็นระบุก็ได้ ดังในรูปที่ 3.8



รูปที่ 3.8 เอกซ์เอ็มแอลสคีมา โครงสร้างของ Binding

- Service ทำหน้าที่เป็นอธิบายที่อยู่ของผู้ให้บริการ เพื่อให้ผู้เรียกใช้บริการสามารถเรียกใช้งานได้อย่างถูกต้อง ซึ่งจะสามารถระบุหรือไม่จำเป็นระบุก็ได้ ดังในรูปที่ รูปที่ 3.9



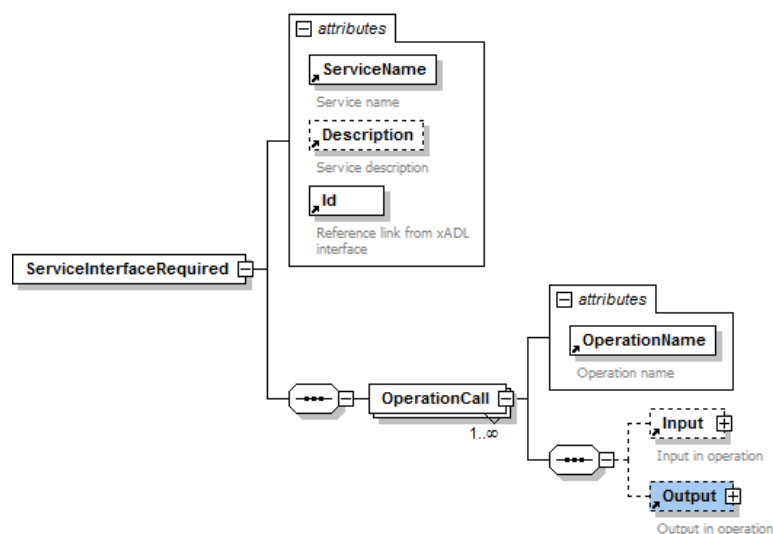
รูปที่ 3.9 เอกซ์เอ็มแอลสคีมา โครงสร้างของ Service

จากองค์ประกอบภายในของ ServiceInterfaceProvided ที่กล่าวมาสามารถเขียนเป็นเอกซ์เอ็มแอลได้ดังรูปที่ 3.10

<code><ServiceInterfaceProvided ServiceName="GetOrderService" Id="GetOrderService"></code>	Type
<pre> <Types> <element name="GetOrderList"> <complexType> <sequence> <element minOccurs="1" maxOccurs="1" name="DD" type="xsd:short"/> <element minOccurs="1" maxOccurs="1" name="MM" type="xsd:short"/> <element minOccurs="1" maxOccurs="1" name="YYYY" type="xsd:short"/> </sequence> </complexType> </element> <element name="GetOrderListResponse"> <complexType> <sequence> <element minOccurs="1" maxOccurs="1" name="OrderListResult" type="xsd:string"/> </sequence> </complexType> </element> </Types> </pre>	Message
<pre> <Message Name="GetOrderListIn"> <Part Name="parameters" Type="xsd:Interger" element="GetOrderList"/> </Message> <Message Name="GetOrderListOut"> <Part Name="parameters" Type="xsd:Interger" element="GetOrderListResponse"/> </Message> </pre>	PortType
<pre> <Port Type> <Operation OperationName="GetOrderList"> <Input Name="GetOrderListIn" Message="GetOrderListIn"/> <Output Message="GetOrderListOut" Name="GetOrderListOut"/> </Operation> </Port Type> </pre>	Binding
<pre> <Binding Name="GetOrderService" Type=""> <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name=" GetOrderList "> <soap:operation soapAction="http://example.com/GetOrderList "/> <input><soap:body use="literal"/></input> <output><soap:body use="literal"/></output> </operation> </Binding > </pre>	Service
<pre> <Service Name="GetOrderService"> <documentation>Get order service</documentation> <port name="GetOrdrePort" binding="tns: GetOrderBinding"> <soap:address location="http://example.com/getorder"/> </port> </Service > </ServiceInterfaceProvided> </pre>	

รูปที่ 3.10 ตัวอย่างเอกซ์เอ็มแอลของ ServiceInterfaceProvided

2) ServiceInterfaceRequired ทำหน้าที่นิยามคุณสมบัติของบริการว่าเป็นผู้ร้องขอ บริการ ซึ่งภายในจะประกอบไปด้วยคำสั่งที่ต้องการเรียกใช้จากผู้ให้บริการ โดยมี OperationCall ที่ทำหน้าที่ในการนิยามคำสั่ง ข้อมูลเข้า และ ข้อมูลออก ที่ผู้ร้องขอบริการต้องการเรียกใช้ ดังในรูปที่ 3.11 ซึ่งคำสั่งจะต้องสอดคล้องกันกับ ส่วนที่ผู้ให้บริการระบุไว้ โดยจะสามารถตรวจสอบความถูกต้อง ในการเรียกใช้ได้จากเครื่องมือ ซึ่งจะกล่าวในบทถัดไป



รูปที่ 3.11 เอกซ์เอ็มแอลสคีมา โครงสร้างของ ServiceInterfaceRequired

จากองค์ประกอบภายในของ ServiceInterfaceRequired ที่กล่าวมาสามารถเขียนเป็นเอกซ์เอ็มแอลได้ดังรูปที่ 3.12

```

<ServiceInterfaceRequired ServiceName="GetShippingStatusRequired"
  Id="GetShippingStatusRequired">
  <OperationCall OperationName="GetShippingStatus">
    <Input Name="GetShippingStatusIn" Message="GetShippingStatusIn"/>
    <Output Name="GetShippingStatusOut" Message="GetShippingStatusOut"/>
  </OperationCall>
</ServiceInterfaceRequired>
</ServiceInterface>
  
```

รูปที่ 3.12 ตัวอย่างเอกซ์เอ็มแอลของ ServiceInterfaceRequired

3.4. การเปรียบเทียบคุณสมบัติของสถาปัตยกรรมเชิงบริการกับส่วนต่อขยายที่พัฒนาขึ้น

จากโครงสร้างส่วนต่อขยายที่พัฒนาขึ้น สามารถนำมาเปรียบเทียบคุณสมบัติของการอธิบายโครงสร้างสถาปัตยกรรมเชิงบริการ จากมาตรฐานที่เป็นที่นิยมใช้กันอยู่เดิม ได้แก่ เอสโอเอเอ็มแอล ดับบลิวเอสดีแอล กับส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอลให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ โดยสามารถเปรียบเทียบคุณสมบัติได้ในตารางที่ 3.1 ดังนี้

ตารางที่ 3.1 ตารางเปรียบเทียบคุณสมบัติของการอธิบายโครงสร้างสถาปัตยกรรมเชิงบริการ

คุณสมบัติสถาปัตยกรรมเชิงบริการ	ส่วนต่อขยายที่พัฒนาขึ้น	อธิบายจากอิลิเมนต์
การมีส่วนร่วมระหว่างบริการ ใช้ในการกำหนดลักษณะของบริการว่า ระหว่างผู้ให้บริการ กับผู้ใช้บริการ ในระบบ เมื่อบริการอยู่ในลักษณะของผู้ให้บริการจะ กำหนดว่าเป็นพอร์ตบริการ แต่เมื่อเป็นผู้ใช้บริการจะกำหนดว่าเป็นพอร์ตร้องขอ	อธิบายได้ในส่วนของการระบุบทบาทของบริการที่มีความสัมพันธ์กันอยู่ในโครงสร้าง ว่าเป็นผู้ร้องขอหรือผู้ให้บริการ	ServiceInterfaceProvided และ ServiceInterfaceRequired
อินเตอร์เฟซของบริการ ใช้ในการอธิบายสิ่งที่ผู้บริการมีให้ และ ข้อตกลงในการร้องขอ หรือ การให้บริการ	สามารถอธิบายได้ในส่วนของการกำหนดคุณสมบัติของบริการว่าประกอบด้วยโอเปอร์เรชันใดบ้าง	portType และ Operation
ข้อมูลของบริการ ใช้ในการอธิบายถึงข้อมูล ที่ข้อความที่แนบไป ในการสื่อสารกันระหว่างผู้ให้บริการและผู้รับบริการ	สามารถอธิบายได้ในการกำหนดรูปแบบของข้อมูลที่ใช้เป็นส่วนเข้าและส่วนออกของข้อมูล	Types และ Message

บทที่ 4

การพัฒนาเครื่องมือ

ในบทนี้จะกล่าวถึงสภาพแวดล้อมในการพัฒนาเครื่องมือ การวิเคราะห์และออกแบบเครื่องมือ โครงสร้างส่วนต่อประสานผู้ใช้ของเครื่องมือในการตรวจสอบและวิเคราะห์โครงสร้างของสถาปัตยกรรมเชิงบริการ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1. สภาพแวดล้อมในการพัฒนาระบบ

1) สภาพแวดล้อมในการพัฒนาเครื่องมือด้านฮาร์ดแวร์

เครื่องคอมพิวเตอร์ตั้งโต๊ะ 1 เครื่อง

- 1) หน่วยประมวลผล Intel i5 Processor ความเร็ว 2.50 กิกะเฮิรซ์ (GHz)
- 2) หน่วยความจำ DDR3 ขนาด 4,096 เมกกะไบต์ (MB)
- 3) ฮาร์ดดิสก์ความเร็ว 5,400 รอบ/วินาที ขนาด 500 กิกะไบต์ (GB)

2) สภาพแวดล้อมในการพัฒนาเครื่องมือด้านซอฟต์แวร์

- 1) ระบบปฏิบัติการวินโดวส์เซเว่น เซอร์วิสแพ็ค 1 (Microsoft Windows 7 Service Pack 1)
- 2) เน็ตบีนส์ไอดีอี เวอร์ชัน 7.2.1 เป็นหลักในการพัฒนาระบบ

4.2. สภาพแวดล้อมในใช้งานเครื่องมือ

1) สภาพแวดล้อมในการพัฒนาเครื่องมือด้านฮาร์ดแวร์

- 1) หน่วยประมวลผล Intel Core2 Duo Processor ความเร็ว 2.50 กิกะเฮิรซ์ (GHz)
- 2) หน่วยความจำ DDR2 ขนาด 2,048 เมกกะไบต์ (MB)
- 3) ฮาร์ดดิสก์ความเร็ว 5,400 รอบ/วินาที ขนาด 250 กิกะไบต์ (GB)

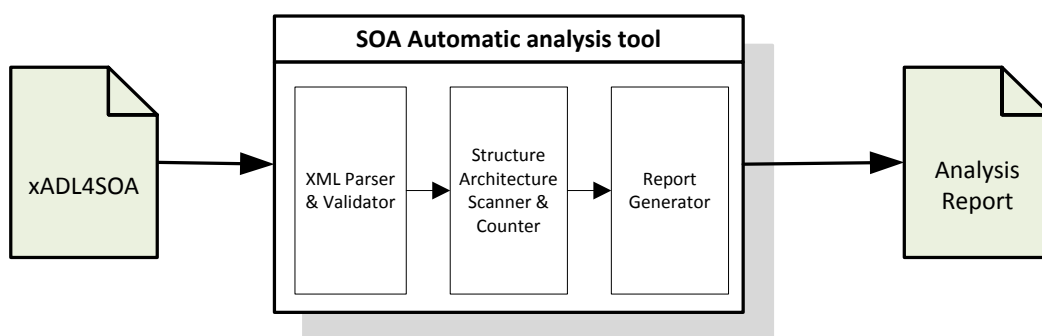
2) สภาพแวดล้อมในการพัฒนาเครื่องมือด้านซอฟต์แวร์

- 1) ระบบปฏิบัติการวินโดวส์เซเว่น เซอร์วิสแพ็ค 1 (Microsoft Windows 7 Service Pack 1)
- 2) จาวา เอสดีเค เวอร์ชัน 6 (Java SDK Standard Edition 6)

4.3. ภาพรวมการทำงานของเครื่องมือ

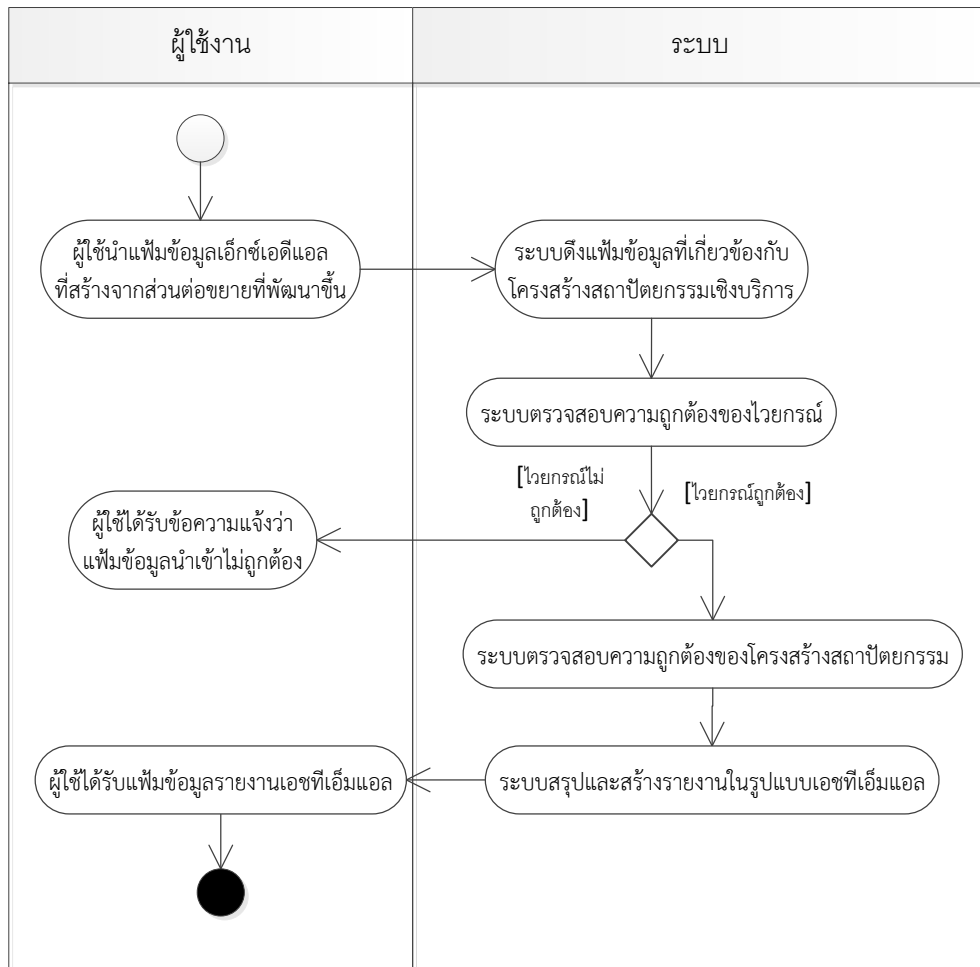
ภาพรวมการทำงานของเครื่องมือที่ใช้ในการประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอ็ดดีแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการได้ จะมีความสามารถในการรายงานผลการวิเคราะห์โครงสร้างของสถาปัตยกรรมเชิงบริการ เพื่อให้สามารถนำไปใช้ในการอธิบายคุณสมบัติของบริการที่อยู่บนโครงสร้างได้ชัดเจนมากยิ่งขึ้น และยังมี ความสามารถในการตรวจสอบความถูกต้องของไวยากรณ์โครงสร้างส่วนต่อขยายโดยการใช้ตัวประมวล ภาษาเอกซ์เอ็มแอล (XML Parser) และส่วนของการวิเคราะห์โครงสร้างของสถาปัตยกรรมเชิงบริการ เพื่อให้ได้รายงานออกมาในรูปแบบของเอกสารเอกซ์เอ็มแอล (HTML) แสดงในรูปที่ 4.1 โดย ส่วนประกอบและขั้นตอนการทำงานของเครื่องมือประกอบด้วย 3 ส่วนดังนี้

- 1) ส่วน XML Parser & Validator เป็นส่วนที่ใช้ในการตรวจสอบความถูกต้องของไวยากรณ์ โครงสร้างของส่วนต่อขยาย ซึ่งจะใช้ไฟล์เอกซ์เอ็มแอลสคีมาของส่วนต่อขยายในการตรวจสอบ ความถูกต้อง กับเอกสารนำเข้าที่เข้ามาว่ามีความถูกต้องและเป็นไปตามโครงสร้างไวยากรณ์ที่ กำหนดไว้ตั้งแต่ต้นหรือไม่ และทำการแปลงเอกสารที่นำเข้ามาจากที่อยู่ในรูปของภาษาเอกซ์เอ็มแอลให้อยู่ในรูปที่สามารถนำไปใช้ในการประมวลผลต่อไปได้
- 2) ส่วน Structure Architecture Scanner & Counter ในส่วนนี้จะทำการรับข้อมูลที่ได้หลังจาก การตรวจสอบไวยากรณ์และแปลงมาแล้ว นำมาประมวลผลโครงสร้างของสถาปัตยกรรมเชิง บริการที่ออกแบบขึ้น
- 3) ส่วน Report Generator เป็นส่วนที่รับข้อมูลที่ได้รับการประมวลผลโครงสร้าง และแปลงออกมา เป็นรายงานในรูปแบบของเอกสารเอกซ์เอ็มแอล เพื่อให้ผู้ใช้งานสามารถอ่านได้ง่ายยิ่งขึ้น



รูปที่ 4.1 ข้อมูลเข้าและข้อมูลออกของเครื่องมือประมวลผลที่พัฒนาขึ้น

จากภาพรวมของเครื่องมือที่กล่าวมา สามารถอธิบายด้วยลำดับของกิจกรรมการทำงานของระบบได้ดังรูปที่ 4.2 โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.2 แผนภาพกิจกรรมการทำงานของระบบ เครื่องมือประมวลผลโครงสร้างสถาปัตยกรรม

- 1) ผู้ใช้งานนำแฟ้มข้อมูลเอ็กซ์เซลที่สร้างขึ้นจากส่วนต่อขยาย อธิบายสถาปัตยกรรมเชิงบริการเข้าสู่ระบบ
- 2) ระบบจะทำการดึงแฟ้มข้อมูลที่เกี่ยวข้องจากการเชื่อมโยงด้วยเอชทีเอ็มแอล ภายในแฟ้มข้อมูลโครงสร้างเอชทีเอ็มแอลมาทำการเก็บไว้ภายในระบบโดยอัตโนมัติ
- 3) ระบบทำการตรวจสอบความถูกต้องทางไวยากรณ์ของแฟ้มเอกสารที่รับเข้ามา ว่าเป็นไปตามไวยากรณ์ที่ถูกต้องหรือไม่ ด้วยการใช้ตัวประมวลผลภาษาเอชทีเอ็มแอล ถ้าไม่ถูกต้องระบบจะทำการแจ้งเตือนไปยังให้ผู้ใช้ทราบว่าแฟ้มข้อมูลที่นำเข้ามาส่วนนั้นมีส่วนใดที่มีไวยากรณ์ไม่ถูกต้อง ถ้าถูกต้องอยู่แล้วระบบจะดำเนินการในขั้นตอนต่อไป

- 4) ระบบทำการตรวจสอบโครงสร้างของสถาปัตยกรรม ว่าภายในคอมโพเนนท์ พบอินเทอร์เน็ตเฟสใดบ้างที่มีการอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ และเป็นผู้ให้บริการ หรือผู้ร้องขอ หากเป็นผู้ร้องขอบริการ ระบบจะทำการตรวจสอบความถูกต้องว่ามีการเชื่อมโยงไปยังผู้ให้บริการถูกต้องหรือไม่ มีการเรียกใช้งานอินเทอร์เน็ตเฟส ข้อมูลเข้าและออกถูกต้องหรือไม่ ซึ่งจะนำไปทำการรายงานผลให้ผู้ใช้ต่อไป
- 5) ระบบทำการสรุปผลนับจำนวนโครงสร้างที่เกี่ยวข้องทั้งหมด และอธิบายรายละเอียดของบริการในแต่ละอินเทอร์เน็ตเฟส ออกมาเป็นรายงานในรูปแบบ HTML และส่งให้ผู้ใช้งาน
- 6) ผู้ใช้งานได้รับแฟ้มข้อมูลรายงาน HTML

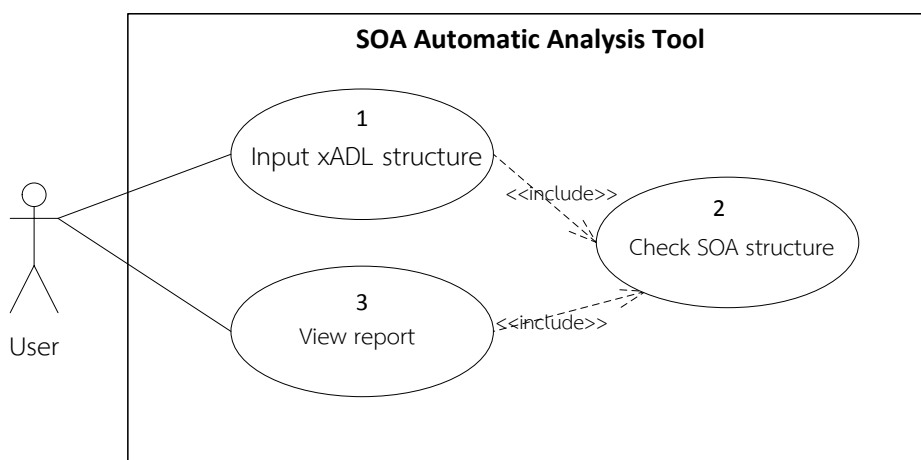
4.4. การวิเคราะห์และออกแบบเครื่องมือ

สำหรับการวิเคราะห์และออกแบบเครื่องมือ นั้น สามารถอธิบายด้วยแผนภาพยูเอ็มแอล ซึ่งเป็นสัญลักษณ์มาตรฐานที่ใช้ในการวิเคราะห์และออกแบบเชิงวัตถุ โดยแผนรูปที่เลือกใช้ได้แก่ แผนภาพยูสเคส (Use case diagram) แผนภาพคลาส (Class diagram) และแผนภาพลำดับ (Sequence diagram)

4.4.1. แผนภาพยูสเคส

จากแผนภาพยูสเคสใช้ในการอธิบายขอบเขตของระบบและฟังก์ชันการทำงานพื้นฐานของระบบในมุมมองของผู้ใช้ แผนภาพยูสเคสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอตีแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการ แสดงได้ดัง

รูปที่ 4.3



รูปที่ 4.3 แผนภาพยูสเคสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรม

จากแผนภาพยูสเคสรูปที่ 4.3 เริ่มต้นผู้ใช้งานนำแฟ้มข้อมูลเอกซ์เอ็ดไอแอลที่สร้างขึ้นจากส่วนต่อขยาย อธิบายสถาปัตยกรรมเชิงบริการ เข้าสู่ระบบ (Input xADL structure) ซึ่งอยู่ในรูปแบบไฟล์ .xml หลังจากนั้นระบบจะทำการดึงแฟ้มข้อมูลที่เกี่ยวข้องกับโครงสร้างสถาปัตยกรรมเชิงบริการ (Find SOA extension files) แล้วระบบจะทำการตรวจสอบความถูกต้องของไวยากรณ์ (Parser XML files) หลังจากนั้นระบบจะทำการตรวจสอบโครงสร้างของสถาปัตยกรรม (Check SOA structure) และสร้างรายงานออกมาในรูปแบบ HTML (Generate report) แล้วส่งไปยังให้ผู้ใช้งานดู (View report) ส่วนรายละเอียดแต่ละยูสเคสแสดงในตารางที่ 4.1 - ตารางที่ 4.3

ตารางที่ 4.1 รายละเอียดยูสเคสนำเข้าแฟ้มข้อมูลเอกซ์เอ็ดไอแอล

ยูสเคส	1: Input xADL structure
แอกเตอร์	ผู้ใช้ (User)
เป้าหมาย	นำเข้าแฟ้มข้อมูลเอกซ์เอ็ดไอแอล
ยูสเคสที่สัมพันธ์	Include: Check SOA structure
เงื่อนไขก่อนหน้า	แฟ้มข้อมูลเอกซ์เอ็ดไอแอลอยู่ในรูปแบบ.xml
ขั้นตอน	1. ผู้ใช้งานเรียกใช้โปรแกรม 2. ระบบสร้างหน้าต่างสำหรับนำเข้าแฟ้มข้อมูล 3. ผู้ใช้เลือกไฟล์นำเข้า
เงื่อนไขภายหลัง	

ตารางที่ 4.2 รายละเอียดยูสเคสตรวจสอบโครงสร้างของสถาปัตยกรรม

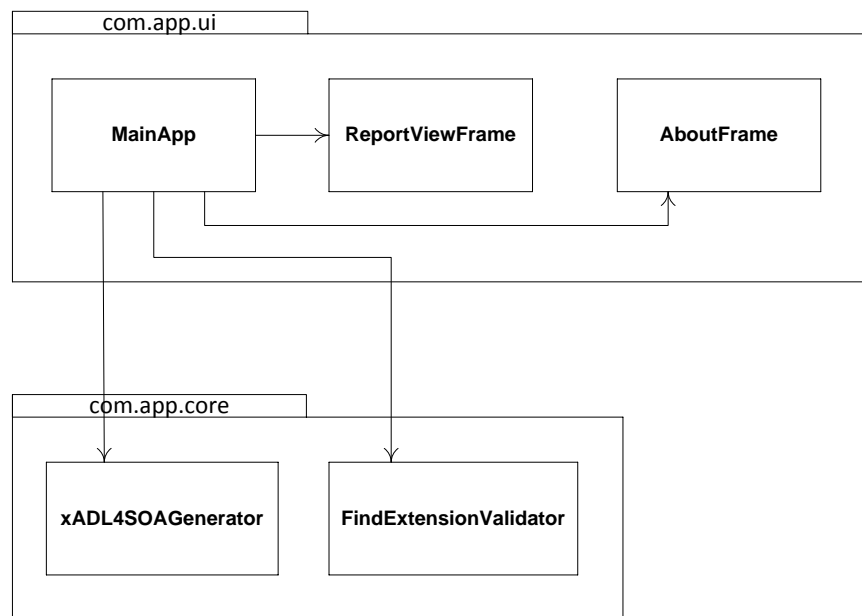
ยูสเคส	2: Check SOA structure
แอกเตอร์	ผู้ใช้ (User)
เป้าหมาย	ตรวจสอบโครงสร้างของสถาปัตยกรรม
ยูสเคสที่สัมพันธ์	Include: Find SOA extension files, Generate report
เงื่อนไขก่อนหน้า	เพิ่มข้อมูลเอกซ์เอตีสแอลอยู่ในรูปแบบ.xml
ขั้นตอน	<ol style="list-style-type: none"> 1. ระบบเรียกใช้ยูสเคส Find SOA extension files 2. นำไฟล์ที่นำเข้ามาจากผู้ใช้ และจากไฟล์เกี่ยวข้อง ที่มีไวยากรณ์ถูกต้อง เข้าทำการตรวจสอบโครงสร้างของสถาปัตยกรรม 3. ค้นหาเพิ่มข้อมูลที่เกี่ยวข้องกับการอธิบายสถาปัตยกรรมเชิงบริการ 4. อ่านข้อมูลโครงสร้าง 5. ระบบเรียกใช้ยูสเคส Generate report
เงื่อนไขภายหลัง	

ตารางที่ 4.3 รายละเอียดยูสเคสดูรายงาน HTML

ยูสเคส	3: View report
แอกเตอร์	ผู้ใช้ (User)
เป้าหมาย	ดูรายงาน HTML
ยูสเคสที่สัมพันธ์	
เงื่อนไขก่อนหน้า	เพิ่มรายงาน HTML
ขั้นตอน	<ol style="list-style-type: none"> 1. ระบบเปิดหน้าต่างแสดงรายงาน 2. ระบบเปิดเพิ่มรายงาน HTML ที่ถูกสร้างขึ้นจากระบบ
เงื่อนไขภายหลัง	

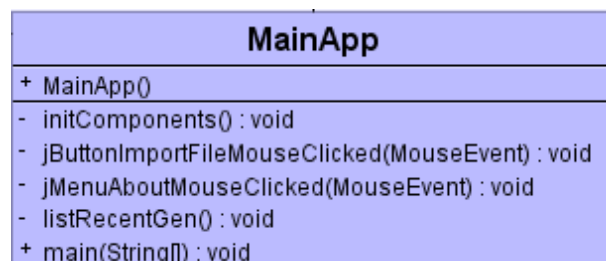
4.4.2. แผนภาพคลาส

แผนภาพคลาสใช้สำหรับแสดงรายละเอียดคลาสและความสัมพันธ์ระหว่างคลาสต่างๆ เพื่อใช้ในการจำลองภาพการออกแบบส่วนที่เป็นโครงสร้างคงที่ของระบบ โดยในรูปที่ 4.4 เป็นแผนภาพคลาสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอ็ดดีแอล ซึ่งในแต่ละคลาสสามารถแบ่งได้ตามแพคเกจ (Package) และมีรายละเอียดดังต่อไปนี้



รูปที่ 4.4 คลาสของเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรม

- 1) คลาส MainApp คือ คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้ในส่วนของการนำเข้าเพิ่มข้อมูล รายละเอียดของคลาสแสดงดังรูปที่ 4.5



รูปที่ 4.5 คลาส MainApp

- 2) คลาส ReportViewFrame คือ คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้ในส่วนของการแสดงผลรายงาน HTML ที่ระบบสร้างขึ้น รายละเอียดของคลาสแสดงดังรูปที่ 4.6

ReportViewFrame	
-	_reportFile : File
+	ReportViewFrame(File)
-	initComponents() : void
-	jButtonExportMouseClicked(MouseEvent) : void
-	showReport() : void

รูปที่ 4.6 คลาส ReportViewFrame

- 3) คลาส AboutFrame คือ คลาสที่ทำหน้าที่ติดต่อกับผู้ใช้ในส่วนของการแสดงข้อมูลเกี่ยวกับโปรแกรมและผู้พัฒนา รายละเอียดของคลาสแสดงดังรูปที่ 4.7

AboutusFrame	
+	AboutusFrame()
-	initComponents() : void
+	main(String[]) : void

รูปที่ 4.7 คลาส AboutFrame

- 4) คลาส FindExtensionValidator คือ คลาสที่ทำหน้าที่การค้นหาเพิ่มข้อมูลที่เกี่ยวข้องกับการอธิบายสถาปัตยกรรมเชิงบริการประมวลผลและตรวจสอบโครงสร้างสถาปัตยกรรม จากเพิ่มข้อมูลเอกซ์เอตีแอล รายละเอียดของคลาสแสดงดังรูปที่ 4.8

FindExtensionValidator	
+	FindExtensionValidator()
+	findExtension(File, String) : Boolean
-	validateSchema(String) : boolean

รูปที่ 4.8 คลาส FindExtensionValidator

- 5) คลาส xADL4SOAGenerator คือ ประมวลผลและตรวจสอบโครงสร้างสถาปัตยกรรม พร้อมทั้งสร้างรายงาน รายละเอียดของคลาสแสดงดังรูปที่ 4.9

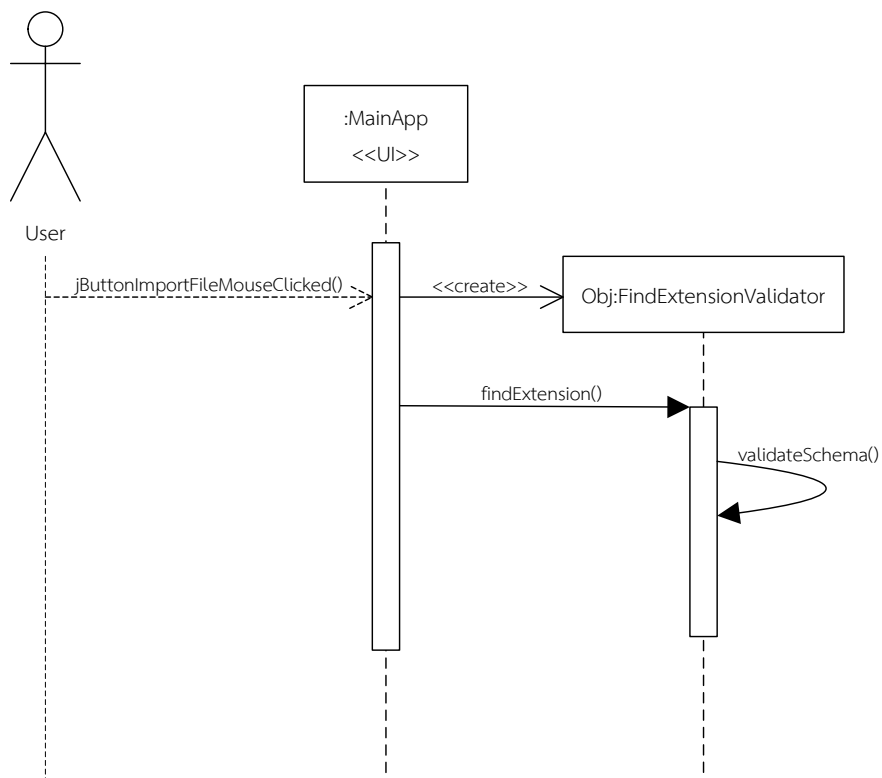
xADL4SOAGenerator
<pre> / arrProvider : String[] / arrRequirer : String[] / doc : Document / strHtmlComponent : String / strHtmlFooter : String / strHtmlHeader : String / strHtmlServiceProvider : String / strHtmlServiceRequirer : String - _fHtml : File - _pathName : String - _xmlFile : File </pre>
<pre> + xADL4SOAGenerator(File, String) </pre>
<pre> - compareServiceInterface(Node, Node) : String - findServiceNodefromFile(File, String) : Node - findSoalInterface(String, String) : Node + generateReport() : File - genProvider(Node) : String - genRequirer(Node) : String - verifyInterface(String, String, Node) : String </pre>

รูปที่ 4.9 คลาส xADL4SOAGenerator

4.4.3. แผนภาพลำดับ

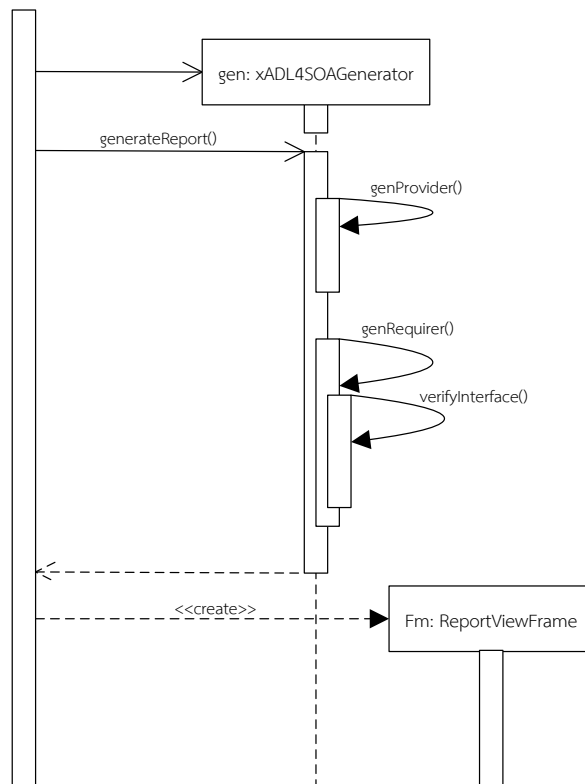
แผนภาพลำดับเป็นแผนรูปที่แสดงปฏิสัมพันธ์ระหว่างวัตถุต่างที่อยู่ภายในระบบ โดยแผนภาพลำดับของเครื่องมือมีดังต่อไปนี้

- 1) แผนภาพลำดับ Input xADL structure เป็นการรับเพิ่มข้อมูลเอกซ์เอดีแอล จากผู้ใช้งานเข้าสู่ระบบ เมื่อผู้ใช้เลือกเพิ่มข้อมูลระบบจะทำการ เรียกใช้อ็อบเจกต์ FindExtensionValidator และเรียกใช้เมทอด findExtension เพื่อทำการค้นหาเพิ่มข้อมูลที่เกี่ยวข้องกับการอธิบายสถาปัตยกรรมเชิงบริการ และทำการตรวจสอบความถูกต้องของไวยากรณ์ หลังจากนั้นจะส่งสถานะการตรวจสอบกลับมาว่าเพิ่มข้อมูลที่เกี่ยวข้องทั้งหมดมีไวยากรณ์ที่ถูกต้องแล้วหรือไม่ ดังรูปที่ 4.10



รูปที่ 4.10 แผนภาพลำดับ Input xADL structure

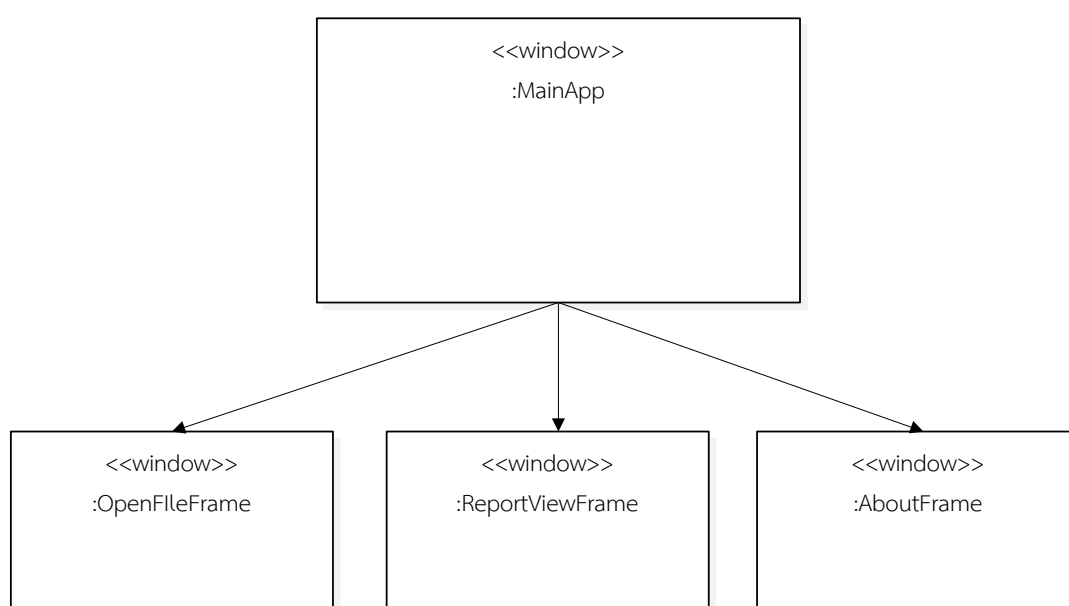
- 2) แผนภาพลำดับ Check SOA structure หลังจากทีระบบทำการค้นหาเพิ่มข้อมูลที่เกี่ยวข้องกับการอธิบายสถาปัตยกรรมเชิงบริการ และทำการตรวจสอบความถูกต้องของไวยากรณ์เป็นที่เรียบร้อยแล้ว หากเพิ่มข้อมูลที่เกี่ยวข้องทั้งหมดมีไวยากรณ์ที่ถูกต้อง ระบบจะทำการสร้างอ็อบเจกต์ xADL4SOAGenerator และเรียกใช้เมทอด generateReport เพื่อทำการตรวจสอบโครงสร้างโดยทำการค้นหาข้อมูลคอมโพเนนท์ อินเทอร์เน็ต และข้อมูลของบริการภายในอินเทอร์เน็ตตามลำดับและทำการเรียกใช้เมทอด genProvided เมื่อพบข้อมูลผู้ให้บริการอยู่ในอินเทอร์เน็ต และเรียกใช้เมทอด genRequired เมื่อพบข้อมูลผู้ร้องขอบริการอยู่ในอินเทอร์เน็ต แล้วทำการตรวจสอบความถูกต้องในการของกสารเรียกใช้งานด้วยเมทอด verifyInterface เมื่อทำการค้นหาข้อมูลเรียบร้อยแล้วระบบจะทำการสร้างไฟล์ HTML และสร้างอ็อบเจกต์ ReportViewFrame เพื่อแสดงหน้าต่างรายงานให้ผู้ใช้งาน ดังรูปที่ 4.11



รูปที่ 4.11 แผนภาพลำดับ Check SOA structure

4.5. โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ

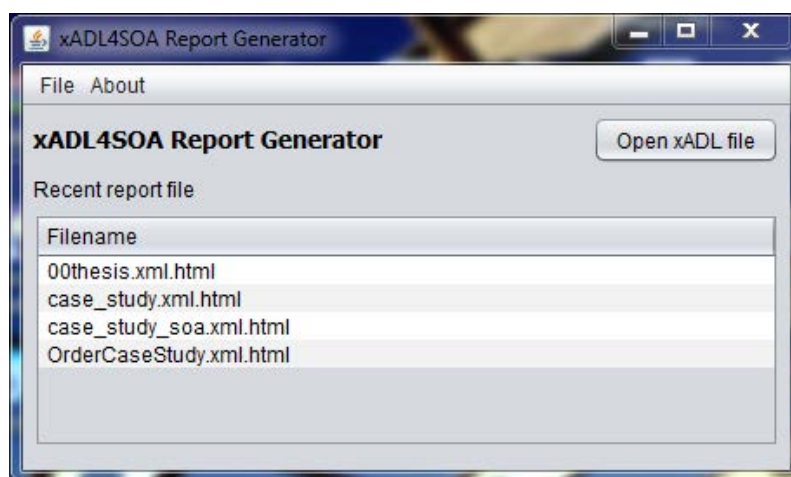
โครงสร้างในส่วนของส่วนต่อประสานผู้ใช้ของเครื่องมือเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอ็ดอีแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการ ด้วยแผนภาพโครงสร้างของจอภาพ ซึ่งเป็นแผนรูปที่ใช้ในการอธิบายหน้าจอที่มีอยู่ภายในระบบ โดยจะแสดงดังรูปที่ 4.12



รูปที่ 4.12 แผนภาพโครงสร้างของจอภาพ เครื่องมือการประมวลผลโครงสร้างของสถาปัตยกรรม

จากรูปที่ 4.12 กรอบ <<window>> แสดงแทนในแต่ละหน้าของส่วนต่อประสานกับผู้ใช้ ประกอบไปด้วย MainApp, OpenFileFrame, ReportViewFrame และ AboutFrame ซึ่งมีรายละเอียดดังต่อไปนี้

- 1) หน้าจอ MainApp เป็นหน้าจอหลักของเครื่องมือเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอ็ดไอแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการ แสดงดังรูปที่ 4.13 โดยสามารถ กดปุ่ม Open xADL file สำหรับนำเข้าแฟ้มข้อมูลเอกซ์เอ็ดไอแอล และสามารถเข้าไปยังหน้าจอ AboutFrame เพื่อดูข้อมูลเกี่ยวกับเครื่องมือ พร้อมทั้งสามารถเลือกรายงานที่ระบบเคยประมวลผลไว้ก่อนได้ที่ Recent report file



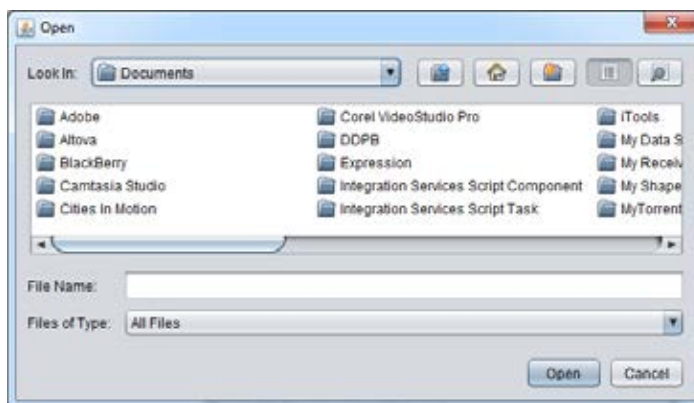
รูปที่ 4.13 หน้าจอหลักของเครื่องมือ

- 2) หน้าจอ AboutFrame ดังรูปที่ 4.14 หน้าจอนี้มาจากการคลิกที่ปุ่ม About ของหน้าจอ MainApp ซึ่งจะแสดงรายละเอียดของเครื่องมือ



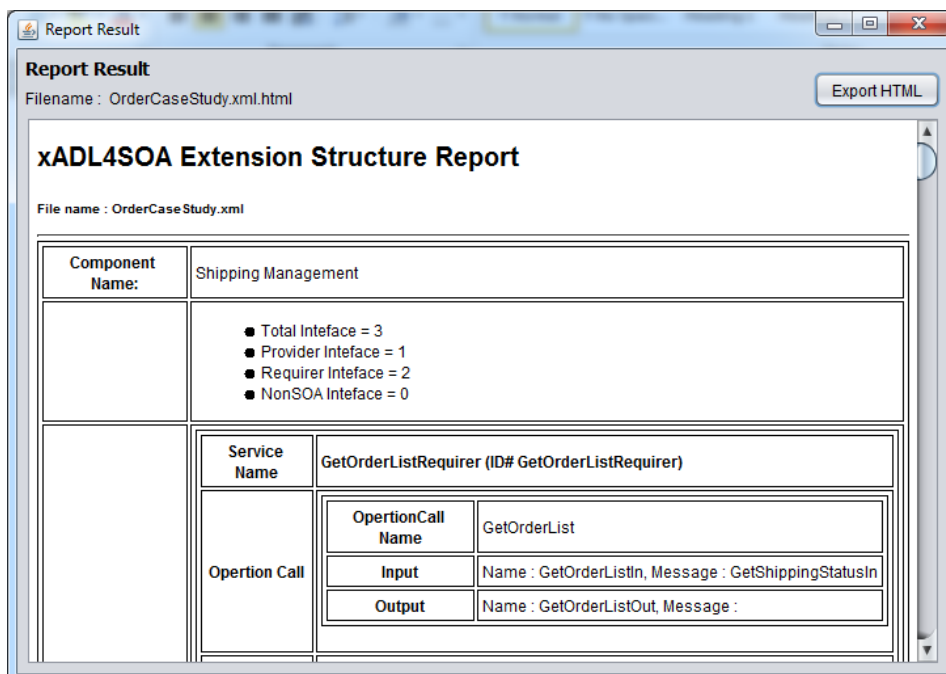
รูปที่ 4.14 หน้าจอข้อมูลเกี่ยวกับเครื่องมือ

- 3) หน้าจอ OpenFileFrame รูปที่ 4.15 หน้าจอนี้มาจากการคลิกปุ่ม Open xADL file ของหน้าจอ MainApp ซึ่งจะแสดงหน้าจอสำหรับการเลือกเพิ่มข้อมูลเข้าสู่ระบบ



รูปที่ 4.15 หน้าจอเลือกเพิ่มข้อมูลเข้าสู่ระบบ

- 4) หน้าจอ ReportViewFrame หน้าจอนี้มีผลจากการประมวลผลหลังจากผู้ใช้นำเพิ่มข้อมูลเข้าสู่ระบบ โดยเมื่อระบบทำการประมวลข้อมูลเสร็จเรียบร้อยแล้วเครื่องมือจะทำการแสดงรายงานผลของโครงสร้างสถาปัตยกรรมดังรูปที่ 4.16



รูปที่ 4.16 หน้าจอรายงานของเครื่องมือ

บทที่ 5

การทดสอบ

ในบทนี้จะกล่าวถึงการทดสอบส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล ให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการ และเครื่องมือที่พัฒนาขึ้น จะทำได้โดยสร้างโครงสร้างสถาปัตยกรรมเชิงบริการทดสอบ เนื้อหาจะประกอบไปด้วยสภาพแวดล้อมที่ใช้ในการทดสอบ ขั้นตอนทดสอบ โครงสร้างของสถาปัตยกรรมเชิงบริการตัวอย่างที่ใช้ในการทดสอบ และสรุปผลการทดสอบ

5.1. สภาพแวดล้อมในการทดสอบ

เป็นสภาพแวดล้อมเดียวกับที่ใช้ในการพัฒนาเครื่องมือในหัวข้อที่ 4.1

5.2. ขั้นตอนการทดสอบ

- 1) ออกแบบโครงสร้างและความสัมพันธ์ของบริการจากรายละเอียดตัวอย่างทดสอบ ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio มาออกแบบโครงสร้างสถาปัตยกรรม
- 2) นำโครงสร้างสถาปัตยกรรมจากตัวอย่างทดสอบที่สร้างขึ้นด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอลมาทำการเพิ่มไวยากรณ์ <soa:soalink> ที่ประกอบด้วยเอกซ์ลิงก์สำหรับเชื่อมโยงการอธิบายคุณสมบัติของบริการไปยังไฟล์ส่วนต่อขยายใหม่ที่จะสร้างขึ้นมา
- 3) สร้างไฟล์ส่วนต่อขยายที่ชื่อไฟล์ตรงกันกับเอกซ์ลิงก์ที่ได้เชื่อมโยงจากไฟล์โครงสร้างโครงสร้างสถาปัตยกรรมไว้ และสร้างส่วนอธิบายคุณสมบัติของบริการตามโครงสร้าง เอกซ์เอ็มแอลสคีมา ที่ได้ออกแบบไว้ โดยสำหรับสำหรับผู้ให้บริการจะเขียนด้วยไวยากรณ์ <ServiceInterfaceProvided> และสำหรับผู้ร้องขอบริการเขียนด้วยไวยากรณ์ <ServiceInterfaceRequired>
- 4) เริ่มทำการตรวจสอบโดยทำการนำแฟ้มข้อมูลโครงสร้างสถาปัตยกรรมเอกซ์เอตีแอล และแฟ้มข้อมูลอธิบายคุณสมบัติของบริการที่เพิ่มเติม เข้าสู่เครื่องมือที่พัฒนาขึ้น เพื่อทำการ

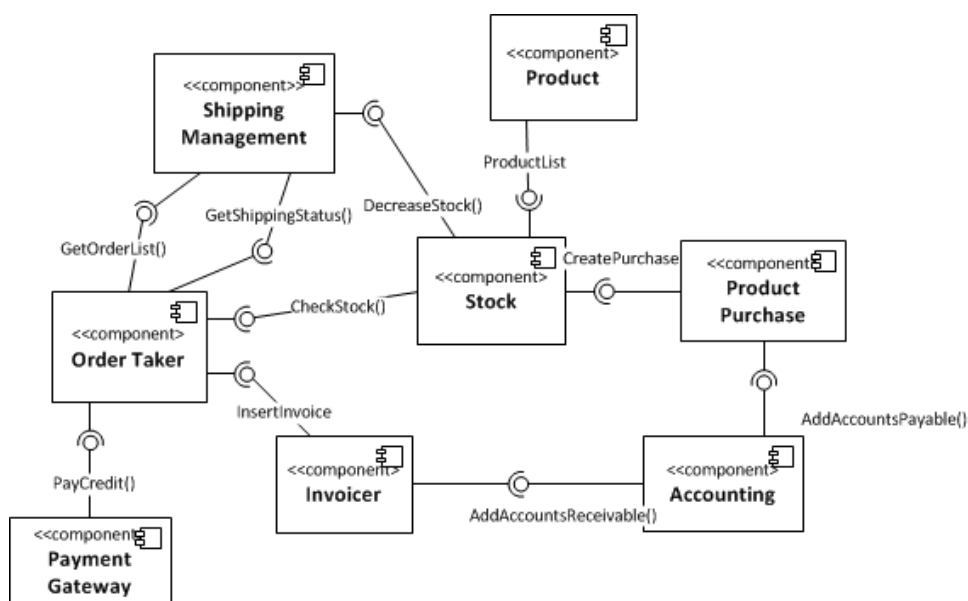
ตรวจสอบความถูกต้องของไวยากรณ์ และความถูกต้องของการเรียกใช้บริการที่เชื่อมโยงอยู่ภายในโครงสร้างที่ออกแบบ

- 5) ทำการทดสอบโดยเปรียบเทียบจากรายละเอียดโครงสร้างของตัวอย่างทดสอบ กับรายงานโครงสร้างจากการนำเพิ่มข้อมูลโครงสร้างที่นำเข้าไปประมวลผลด้วยเครื่องมือที่พัฒนาขึ้น
- 6) รวบรวมและสรุปผลการทดลอง

5.3. ตัวอย่างที่ใช้ในการทดสอบ

งานวิทยานิพนธ์นี้สร้างตัวอย่างที่ใช้ในการทดสอบ คือระบบสั่งซื้อสินค้า โดยตัวอย่างนี้จะอธิบายด้วยแผนภาพยูเอ็มแอล มีรายละเอียดดังต่อไปนี้

ระบบสั่งซื้อสินค้า มีลักษณะการทำงานตามแผนภาพคอมโพเนนต์ที่ไดอะแกรม ในรูปที่ 5.1 เป็นสถาปัตยกรรมแบบผสม โดยส่วนใหญ่จะเป็นโครงสร้างสถาปัตยกรรมเชิงบริการ มีคอมโพเนนต์ต่างๆ แทนระบบย่อยในการทำงาน และมีการเชื่อมโยงการทำงานด้วยเซอร์วิส โครงสร้างทั้งหมดจะประกอบไปด้วยคอมโพเนนต์จำนวน 8 คอมโพเนนต์ อินเทอร์เฟซทั้งหมดจำนวน 20 อินเทอร์เฟซ แบ่งเป็นอินเทอร์เฟซที่เป็นสถาปัตยกรรมเชิงบริการจำนวน 16 อินเทอร์เฟซ และอินเทอร์เฟซแบบอื่นๆจำนวน 4 อินเทอร์เฟซ ซึ่งรายละเอียดโครงสร้างสถาปัตยกรรมทั้งหมดมีดังนี้



รูปที่ 5.1 แผนภาพคอมโพเนนต์ที่ไดอะแกรมระบบสั่งซื้อสินค้า

- 1) คอมโพเนนท์ Order Taker ทำหน้าที่แทนระบบสั่งซื้อสินค้าผ่านทางเว็บไซต์ เพื่อให้ลูกค้าเข้ามาทำการสั่งซื้อสินค้า โดยมีการตรวจสอบข้อมูลจำนวนของสินค้าว่ามีเพียงพอให้สั่งหรือไม่ก่อนทำการสั่งซื้อโดยมีการดึงข้อมูลจากระบบคลังสินค้ามาตรวจสอบ (คอมโพเนนท์ Stock) และสามารถตรวจสอบสถานะการสั่งซื้อได้ จากการตรวจสอบข้อมูลจากระบบจัดส่งสินค้า (คอมโพเนนท์ Shipping Management) ซึ่งคอมโพเนนท์ Order Taker มีอินเตอร์เฟสที่เชื่อมต่อกับระบบอื่นๆจำนวน 5 อินเตอร์เฟสดังนี้
- ก) อินเตอร์เฟส OrderList ทำหน้าที่เป็นเซอร์วิสผู้ให้บริการข้อมูลรายการสั่งซื้อสินค้าของลูกค้า โดยมีคุณสมบัติของบริการดังตารางที่ 5.1

ตารางที่ 5.1 คุณสมบัติของบริการภายในอินเตอร์เฟส ServiceOrderList

ชื่อบริการ	GetOrderService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	GetOrderList
	ข้อมูลเข้า	(Date) OrderDate
	ข้อมูลออก	(Obj) OrderList
	คำอธิบาย	สำหรับการดึงรายการคำสั่งซื้อในแต่ละวัน

- ข) อินเตอร์เฟส ReqShippingStatus ทำหน้าที่ร้องขอบริการสถานะของการส่งสินค้าในแต่ละรายการ โดยมีรายละเอียดการร้องขอดังตารางที่ 5.2

ตารางที่ 5.2 คุณสมบัติของบริการภายในอินเตอร์เฟส ReqShippingStatus

ชื่อบริการ	GetShippingStatusRequired
โอเปอเรชันที่เรียกใช้	GetShippingStatus
ข้อมูลเข้า	(Int) ShippingId
ข้อมูลออก	(Int) ShippingId, (Enum) ShippingStatus
คำอธิบาย	สำหรับเรียกดูสถานะการส่งสินค้าในแต่ละรายการ

- ค) อินเตอร์เฟส ReqPayCredit ทำหน้าที่ร้องขอบริการสถานะของการส่งสินค้าในแต่ละรายการ โดยมีรายละเอียดการร้องขอดังตารางที่ 5.3

ตารางที่ 5.3 คุณสมบัติของบริการภายในอินเทอร์เฟซ ReqPayCredit

ชื่อบริการ	PayCreditRequired
โอเปอเรชันที่เรียกใช้	PayCredit
ข้อมูลเข้า	(Int) OrderId, (String) CreditCode, (Float) Amount
ข้อมูลออก	(Int) ShippingId, (Enum) ShippingStatus
คำอธิบาย	สำหรับเรียกส่งคำสั่งชำระเงินด้วยบัตรเครดิต

- ง) อินเทอร์เฟซ ReqCheckStock ทำหน้าที่ร้องขอบริการสถานะของการส่งสินค้าในแต่ละรายการ โดยมีรายละเอียดการร้องขอดังตารางที่ 5.4

ตารางที่ 5.4 คุณสมบัติของบริการภายในอินเทอร์เฟซ ReqCheckStock

ชื่อบริการ	CheckStockRequired
โอเปอเรชันที่เรียกใช้	CheckStock
ข้อมูลเข้า	(Int) ProductId
ข้อมูลออก	(Int) ProductId, (String) ProductName, (Int) Qty
คำอธิบาย	สำหรับเรียกดูจำนวนสินค้าในแต่ละรายการ

- จ) อินเทอร์เฟซ InsertInvoice ทำหน้าที่เชื่อมต่อกับคอมโพเนนต์ Invoicer เพื่อทำการสร้างใบแจ้งหนี้โดยไม่ได้มีการเชื่อมต่อแบบเซอร์วิสบริการ
- 2) คอมโพเนนต์ Shipping Management ทำหน้าที่แทนระบบจัดการการส่งสินค้า เพื่อให้ทางร้านสามารถจัดการการส่งสินค้าไปยังมือของลูกค้าได้อย่างถูกต้อง โดยมีการตรวจสอบรายการสั่งซื้อในแต่ละวันจากระบบสั่งซื้อสินค้า (คอมโพเนนต์ Order Taker) และทำการส่งคำสั่งลดสินค้าจำนวนในระบบคลังสินค้า (คอมโพเนนต์ Stock) เมื่อสินค้าถูกส่งไปยังลูกค้าเรียบร้อยแล้ว ซึ่งคอมโพเนนต์ Shipping Management มีอินเทอร์เฟซที่เชื่อมต่อกับระบบอื่นๆจำนวน 3 อินเทอร์เฟซดังนี้
- ก) อินเทอร์เฟซ ReqOrderList ทำหน้าที่ร้องขอบริการรายการคำสั่งซื้อในแต่ละวัน โดยมีรายละเอียดการร้องขอดังตารางที่ 5.5

ตารางที่ 5.5 คุณสมบัติของบริการภายในอินเทอร์เฟซ ReqOrderList

ชื่อบริการ	GetOrderListRequired
โอเปอเรชันที่เรียกใช้	GetOrderList
ข้อมูลเข้า	(Date) OrderDate
ข้อมูลออก	(Obj) OrderList
คำอธิบาย	สำหรับเรียกดูรายการคำสั่งซื้อในแต่ละวัน

- ข) อินเทอร์เฟซ ReqDecreaseStock ทำหน้าที่ร้องขอคำสั่งลดจำนวนสินค้าในระบบหลังจากทำการส่งสินค้าไปแล้ว โดยมีรายละเอียดการร้องขอดังตารางที่ 5.6

ตารางที่ 5.6 คุณสมบัติของบริการภายในอินเทอร์เฟซ ReqDecreaseStock

ชื่อบริการ	DecreaseStockRequired
โอเปอเรชันที่เรียกใช้	DecreaseStock
ข้อมูลเข้า	(Int) ProductId, (Int), Qty
ข้อมูลออก	(Int) Status
คำอธิบาย	สำหรับเรียกคำสั่งลดจำนวนสินค้าในระบบคลังสินค้า

- ค) อินเทอร์เฟซ ShippingStatus ทำหน้าที่เป็นเซอร์วิสผู้ให้บริการข้อมูลสถานะการส่งสินค้าในแต่ละรายการ โดยมีคุณสมบัติของบริการดังตารางที่ 5.7

ตารางที่ 5.7 คุณสมบัติของบริการภายในอินเทอร์เฟซ ShippingStatus

ชื่อบริการ	GetShippingStatusService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	GetShippingStatus
	ข้อมูลเข้า	(Int) ShippingId
	ข้อมูลออก	(Int) ShippingId, (Enum) ShippingStatus
	คำอธิบาย	สำหรับการดึงสถานะการส่งสินค้าในแต่ละรายการ

- 3) คอมพิวเตอร์ Stock ทำหน้าที่แทนระบบจัดการคลังสินค้า โดยมีการเชื่อมโยงกับระบบรายการสินค้า (คอมพิวเตอร์ Product) เพื่อดึงรายการสินค้าเข้าสู่ระบบ มีการเชื่อมโยงกับระบบจัดซื้อสินค้า (คอมพิวเตอร์ Product Purchase) เมื่อสินค้าใกล้หมดจากภายในคลังสินค้า ซึ่งคอมพิวเตอร์ Stock มีอินเตอร์เฟซที่เชื่อมต่อกับระบบอื่นๆจำนวน 4 อินเตอร์เฟซดังนี้
- ก) อินเตอร์เฟซ CheckStock ทำหน้าที่เป็นเซิร์ฟวิสผู้ให้บริการข้อมูลจำนวนสินค้าในแต่ละรายการ โดยมีคุณสมบัติของบริการดังตารางที่ 5.8

ตารางที่ 5.8 คุณสมบัติของบริการภายในอินเตอร์เฟซ CheckStock

ชื่อบริการ	CheckStockService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	CheckStock
	ข้อมูลเข้า	(Int) ProductId
	ข้อมูลออก	(Int) ProductId, (String) ProductName, (Int) Qty
	คำอธิบาย	สำหรับการดึงจำนวนสินค้าในแต่ละรายการ

- ข) อินเตอร์เฟซ DecreaseStock ทำหน้าที่เป็นเซิร์ฟวิสผู้ให้บริการคำสั่งการลดจำนวนสินค้า โดยมีคุณสมบัติของบริการดังตารางที่ 5.9

ตารางที่ 5.9 คุณสมบัติของบริการภายในอินเตอร์เฟซ DecreaseStock

ชื่อบริการ	DecreaseStockService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	DecreaseStock
	ข้อมูลเข้า	(Int) ProductId, (Int), Qty
	ข้อมูลออก	(Int) Status
	คำอธิบาย	คำสั่งสำหรับลดจำนวนสินค้าในระบบ

- ค) อินเทอร์เน็ต ReqProductList ทำหน้าที่เชื่อมต่อกับคอมโพเนนท์ Product เพื่อทำดึงข้อมูลรายการสินค้าจากระบบโดยไม่ได้มีการเชื่อมต่อแบบเซอร์วิสบริการ
- ง) อินเทอร์เน็ต ReqCreateProductPurchase ทำหน้าที่เชื่อมต่อกับคอมโพเนนท์ Product Purchase เพื่อทำสร้างรายการคำสั่งจัดซื้อสินค้าโดยไม่ได้มีการเชื่อมต่อแบบเซอร์วิสบริการ
- 4) คอมโพเนนท์ Product ทำหน้าที่แทนระบบจัดการรายการและราคาสินค้า โดยมีการเชื่อมโยงกับระบบคลังสินค้า ซึ่งคอมโพเนนท์ Product Purchase มีอินเทอร์เน็ตที่เชื่อมต่อกับระบบอื่นๆ จำนวน 1 อินเทอร์เน็ตดังนี้
- ก) อินเทอร์เน็ต ProductList ทำหน้าที่เชื่อมต่อกับคอมโพเนนท์ Stock เพื่อให้ข้อมูลรายการสินค้ากับระบบโดยไม่มีมีการเชื่อมต่อแบบเซอร์วิสบริการ
- 5) คอมโพเนนท์ Product Purchase ทำหน้าที่แทนระบบจัดซื้อสินค้า เมื่อสินค้าใกล้หมดจากคลังสินค้า ที่เชื่อมต่อกับระบบบัญชี (คอมโพเนนท์ Accounting) เพื่อทำการบันทึกข้อมูลการรายจ่ายเข้าสู่ระบบต่อไป ซึ่งคอมโพเนนท์ Product Purchase มีอินเทอร์เน็ตที่เชื่อมต่อกับระบบอื่นๆจำนวน 2 อินเทอร์เน็ตดังนี้
- ก) อินเทอร์เน็ต ReqAddAccountPayable ทำหน้าที่ร้องขอคำสั่งบันทึกข้อมูลรายจ่ายจากระบบบัญชี เมื่อมีการจัดซื้อสินค้าสำเร็จ โดยมีรายละเอียดการร้องขอดังตารางที่ 5.10

ตารางที่ 5.10 คุณสมบัติของบริการภายในอินเทอร์เน็ต ReqAddAccountPayable

ชื่อบริการ	AddAccountsPayableRequired
โอเปอเรชันที่เรียกใช้	AddAccountsPayable
ข้อมูลเข้า	(String) Description, (Float), TotalPayable
ข้อมูลออก	(Int) Status
คำอธิบาย	สำหรับเรียกคำสั่งบันทึกข้อมูลรายจ่าย

- ข) อินเทอร์เน็ต CreateProductPurchase ทำหน้าที่เชื่อมต่อกับคอมโพเนนท์ Product เพื่อทำการสร้างรายการคำสั่งจัดซื้อสินค้าโดยไม่ได้มีการเชื่อมต่อแบบเซอร์วิสบริการ

- 6) คอมพิวเตอร์ Accounting ทำหน้าที่แทนระบบบัญชี โดยมีการเชื่อมโยงกับระบบจกซื้อสินค้า (คอมพิวเตอร์ Product Purchase) เพื่อบันทึกข้อมูลรายจ่ายเมื่อมีการสั่งจัดซื้อสินค้า และเชื่อมโยงกับระบบออกใบแจ้งหนี้ (คอมพิวเตอร์ Invoicer) เพื่อทำการบันทึกข้อมูลรายได้ เมื่อรายการใบแจ้งหนี้มีการชำระเงินเป็นที่เรียบร้อยแล้ว ซึ่งคอมพิวเตอร์ Accounting มีอินเตอร์เฟซที่เชื่อมต่อกับระบบอื่นๆจำนวน 2 อินเตอร์เฟซดังนี้
- ก) อินเตอร์เฟซ AddAccountPayable ทำหน้าที่เป็นเซิร์ฟวิสผู้ให้บริการคำสั่งบันทึกข้อมูลรายจ่าย โดยมีคุณสมบัติของบริการดังตารางที่ 5.11

ตารางที่ 5.11 คุณสมบัติของบริการภายในอินเตอร์เฟซ AddAccountPayable

ชื่อบริการ	AddAccountsPayableService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	AddAccountsPayable
	ข้อมูลเข้า	(String) Description, (Float), TotalPayable
	ข้อมูลออก	(Int) Status
	คำอธิบาย	สำหรับเรียกคำสั่งบันทึกข้อมูลรายจ่าย

- ข) อินเตอร์เฟซ AddAccountsReceivable ทำหน้าที่เป็นเซิร์ฟวิสผู้ให้บริการคำสั่งบันทึกข้อมูลรายรับ โดยมีคุณสมบัติของบริการดังตารางที่ 5.12

ตารางที่ 5.12 คุณสมบัติของบริการภายในอินเตอร์เฟซ AddAccountPayable

ชื่อบริการ	AddAccountsReceivableService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	AddAccountsReceivable
	ข้อมูลเข้า	(String) Description, (Float) TotalReceive
	ข้อมูลออก	(Int) Status
	คำอธิบาย	สำหรับเรียกคำสั่งบันทึกข้อมูลรายรับ

- 7) คอมพิวเตอร์ Invoicer ทำหน้าที่แทนระบบจัดการรายการใบแจ้งหนี้ โดยมีการเชื่อมโยงกับระบบสั่งซื้อสินค้า เพื่อรองรับการสร้างใบแจ้งหนี้จากระบบ และเชื่อมโยงกับระบบบัญชี เพื่อทำ

การบันทึกข้อมูลรายได้ เมื่อรายการในใบแจ้งหนี้มีการชำระเงินเป็นที่เรียบร้อยแล้ว ซึ่งคอมพิวเตอร์ Invoicer มีอินเตอร์เฟซที่เชื่อมต่อกับระบบอื่นๆจำนวน 2 อินเตอร์เฟซดังนี้

- ก) อินเตอร์เฟซ ProductList ทำหน้าที่เชื่อมต่อกับคอมพิวเตอร์ Stock เพื่อให้ข้อมูลรายการสินค้ากับระบบโดยไม่มีการเชื่อมต่อแบบเซอร์วิสบริการ
- ข) อินเตอร์เฟซ ReqAddAccountsReceivable ทำหน้าที่ร้องขอคำสั่งบันทึกข้อมูลรายรับจากระบบบัญชี โดยมีรายละเอียดการร้องขอดังตารางที่ 5.13

ตารางที่ 5.13 คุณสมบัติของบริการภายในอินเตอร์เฟซ ReqAddAccountsReceivable

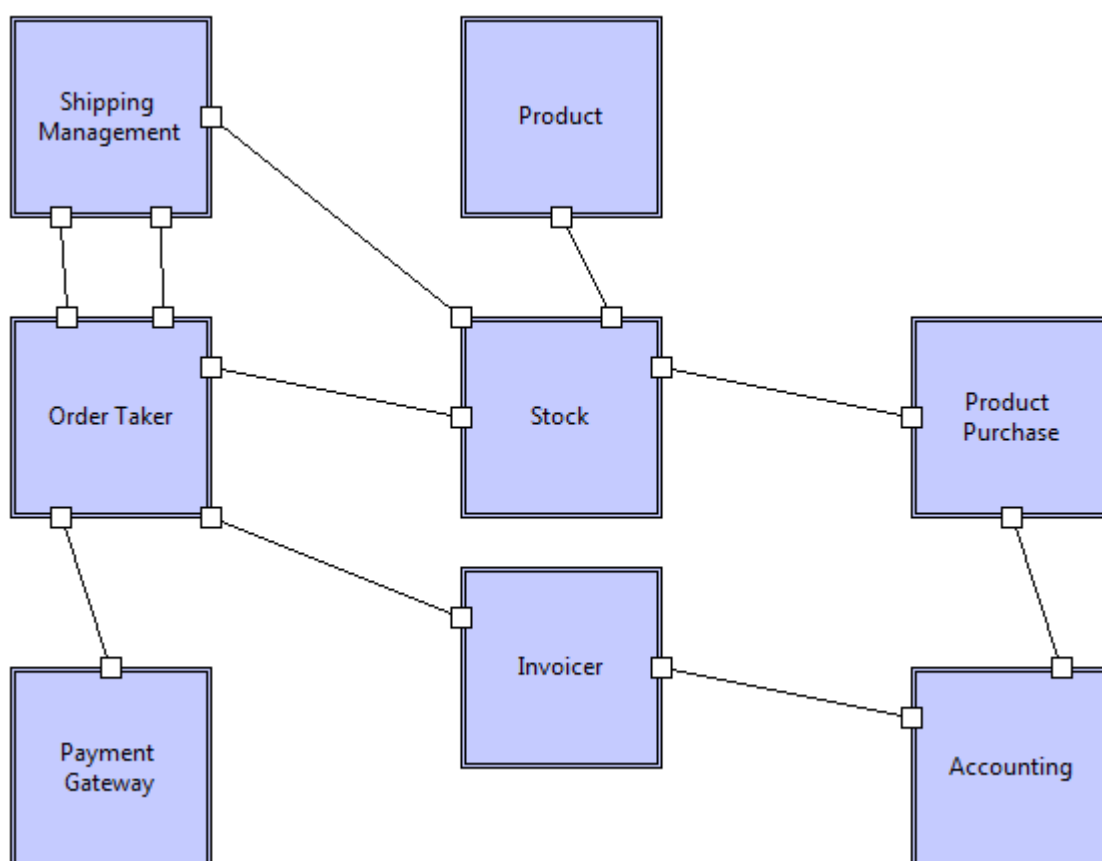
ชื่อบริการ	AddAccountsReceivableRequired
โอเปอเรชันที่เรียกใช้	AddAccountsReceivable
ข้อมูลเข้า	(String) Description, (Float), TotalReceivable
ข้อมูลออก	(Int) Status
คำอธิบาย	สำหรับเรียกคำสั่งบันทึกข้อมูลรายรับ

- 8) คอมพิวเตอร์ Payment Gateway ทำหน้าที่แทนระบบรับชำระเงิน โดยมีการเชื่อมโยงกับระบบสั่งซื้อสินค้า (คอมพิวเตอร์ Order Taker) สำหรับการรับชำระเงินด้วยบัตรเครดิต ซึ่งคอมพิวเตอร์ Payment Gateway มีอินเตอร์เฟซที่เชื่อมต่อกับระบบอื่นๆจำนวน 1 อินเตอร์เฟซดังนี้
- ก) อินเตอร์เฟซ PayCredit ทำหน้าที่เป็นเซอร์วิสผู้ให้บริการข้อมูลจำนวนสินค้าในแต่ละรายการ โดยมีคุณสมบัติของบริการดังตารางที่ 5.14

ตารางที่ 5.14 คุณสมบัติของบริการภายในอินเตอร์เฟซ PayCredit

ชื่อบริการ	PayCreditService	
รายการโอเปอเรชัน	ชื่อโอเปอเรชัน	PayCredit
	ข้อมูลเข้า	(Int) OrderId, (String) CreditCode, (Float) Amount
	ข้อมูลออก	(Int) ShippingId, (Enum) ShippingStatus
	คำอธิบาย	คำสั่งสำหรับชำระเงินด้วยบัตรเครดิต

จากโครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบสั่งซื้อสินค้า สามารถนำมาออกแบบโครงสร้างและความสัมพันธ์ของบริการจากรายละเอียดที่กล่าวมาข้างต้น ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio ได้ดังรูปที่ 5.2



รูปที่ 5.2 โครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบสั่งซื้อสินค้า ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio

หลังจากที่ทำการออกแบบโครงสร้างสถาปัตยกรรมซอฟต์แวร์ของระบบสั่งซื้อสินค้า ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio เป็นที่เรียบร้อยแล้ว ต่อมาจะต้องทำการเพิ่มไวยากรณ์ <soa:soalink> ที่ประกอบด้วยเอกซ์ลิงก์สำหรับเชื่อมโยงการอธิบายคุณสมบัติของบริการไปยังไฟล์ส่วนต่อขยายใหม่ที่จะสร้างขึ้นมา จากรูปที่ 5.3 เป็นตัวอย่างการเพิ่มเติมไวยากรณ์ภายในอินเตอร์เฟซของคอมโพเนนต์ Order Taker


```

<types:component types:id="componentffa80107-929e369c-5c15710c-c276004d"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Order Taker</types:description>

  <!--Interface OrderList -->
  <types:interface types:id="interfaceffa80107-92a077c4-49692544-c27602a9"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">OrderList</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <!--SOALink to OrderTakerService.xml#GetOrderService -->
  <soa:soalink xlink:href="OrderTakerService.xml#GetOrderService" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>

  <!--Interface ReqShippingStatus -->
  <types:interface types:id="interfaceffa80107-92a1d333-08930f4f-c276033b"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqShippingStatus</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <!--SOALink to OrderTakerService.xml#GetShippingStatusRequired -->
  <soa:soalink xlink:href="OrderTakerService.xml#GetShippingStatusRequired"
xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:interface>

  <!--Interface ReqInsertInvoice -->
  <types:interface types:id="interfaceffa80107-92aa131a-72aa45a0-c2760596"
xsi:type="types:Interface">
  <types:description xsi:type="instance:Description">ReqInsertInvoice</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  </types:interface>

  <!--Interface ReqPayCredit -->
  <types:interface types:id="interfaceffa80107-92ab03d1-72c832cb-c27605e7"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqPayCredit</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <!--SOALink to OrderTakerService.xml#PayCreditRequired -->
  <soa:soalink xlink:href="OrderTakerService.xml#PayCreditRequired" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>

  <!--Interface ReqCheckStock -->
  <types:interface types:id="interfaceffa80107-92ac0fba-74b3ecd3-c2760637"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqCheckStock</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <!--SOALink to OrderTakerService.xml#CheckStockRequired -->
  <soa:soalink xlink:href="OrderTakerService.xml#CheckStockRequired" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>
</types:component>

```

รูปที่ 5.3 เพิ่มเติมไวยกรณ์ soalink ภายในอินเตอร์เฟซของคอมโพเนนท์ Order Taker

เมื่อได้เพิ่มเติมไวยากรณ์ภายในโครงสร้างสถาปัตยกรรมซอฟต์แวร์ภาษาอธิบายสถาปัตยกรรม เอกซ์เอตีแอลเป็นที่เรียบร้อยแล้ว จะต้องทำการเขียนคุณสมบัติของบริการไปยังไฟล์ส่วนต่อขยายใหม่ ตามเอกซ์ลิงก์ที่ได้รับไว้ ในรูปที่ 5.4 เป็นตัวอย่างการเขียนคุณสมบัติของบริการ GetOrderService และ GetShippingStatusRequired จากการเชื่อมโยงเอกซ์ลิงก์ภายในคอมโพเนนท์ Order Taker

```

<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xad14soa/
GetOrderService
  <ServiceInterfaceProvided ServiceName="GetOrderService" Id="GetOrderService">
    <Types>
      <element name="GetOrderList">
        <complexType> GetOrderService
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="DD" type="xsd:short"/>
            <element minOccurs="1" maxOccurs="1" name="MM" type="xsd:short"/>
            <element minOccurs="1" maxOccurs="1" name="YYYY" type="xsd:short"/>
          </sequence>
        </complexType>
      </element>
      <element name="GetOrderListResponse">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="OrderListResult" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
    </Types>
    <Message Name="GetOrderListIn">
      <Part Name="parameters" Type="xsd:Interger" element="GetOrderList"/>
    </Message>
    <Message Name="GetOrderListOut">
      <Part Name="parameters" Type="xsd:Interger" element="GetOrderListResponse"/>
    </Message>
    <PortType>
      <Operation OperationName="GetOrderList">
        <Input Name="GetOrderListIn" Message="GetOrderListIn"/>
        <Output Message="GetOrderListOut" Name="GetOrderListOut"/>
      </Operation>
    </PortType>
    <Binding Name="GetOrderService" Type=""/>
    <Service Name="GetOrderService"/>
  </ServiceInterfaceProvided>
  <ServiceInterfaceRequired ServiceName="GetShippingStatusRequired"
  Id="GetShippingStatusRequired">
    <OperationCall OperationName="GetShippingStatus">
      <Input Name="GetShippingStatusIn" Message="GetShippingStatusIn"/>
      <Output Name="GetShippingStatusOut" Message="GetShippingStatusOut"/>
    </OperationCall>
  </ServiceInterfaceRequired>
</ServiceInterface>
GetShippingStatusRequirer

```

รูปที่ 5.4 รายละเอียดคุณสมบัติของบริการ GetOrderService

เมื่อทำการเขียนคุณสมบัติในโครงสร้างทั้งหมดเป็นที่เรียบร้อยแล้ว จะนำเพิ่มข้อมูล
โครงสร้างสถาปัตยกรรมเอกซ์เอตีแอล และเพิ่มข้อมูลอธิบายคุณสมบัติของบริการที่เพิ่มเติม เข้าสู่
เครื่องมือที่พัฒนาขึ้น จะได้รายงานผลลัพธ์รายละเอียดโครงสร้างและอธิบายคุณสมบัติทั้งหมดของ
บริการ ออกมาดังภาคผนวก ก ในรูปที่ 5.5 แสดงตัวอย่างรายงาน ของคอมโพเนนท์ OrderTaker

Component Name:	Order Taker																																																		
	<ul style="list-style-type: none"> • Total Interface = 5 • Provided Interface = 1 • Required Interface = 3 • NonSOA Interface = 1 																																																		
Interface :	<table border="1"> <tr> <td>Service Name</td> <td colspan="2">GetOrderService (ID# GetOrderService)</td> </tr> <tr> <td>Type</td> <td colspan="2">xxx</td> </tr> <tr> <td>Message</td> <td colspan="2">Message Name : GetOrderListIn</td> </tr> <tr> <td rowspan="2">Part</td> <td>Part Name</td> <td>Type</td> </tr> <tr> <td>parameters</td> <td>xsd:Interger</td> </tr> <tr> <td>Message</td> <td colspan="2">Message Name : GetOrderListOut</td> </tr> <tr> <td rowspan="2">Part</td> <td>Part Name</td> <td>Type</td> </tr> <tr> <td>parameters</td> <td>xsd:Interger</td> </tr> <tr> <td rowspan="3">PortType</td> <td>Opertion Name</td> <td>GetOrderList</td> </tr> <tr> <td>Input</td> <td>Name : GetOrderListIn, Message : GetOrderListIn</td> </tr> <tr> <td>Output</td> <td>Name : GetOrderListOut, Message :</td> </tr> <tr> <td>Binding</td> <td colspan="2"></td> </tr> <tr> <td>Sevice</td> <td colspan="2"></td> </tr> </table> <table border="1"> <tr> <td>Service Name</td> <td colspan="2">GetShippingStatusRequired (ID# GetShippingStatusRequired)</td> </tr> <tr> <td rowspan="3">Opertion Call</td> <td>OpertionCall Name</td> <td>GetShippingStatus</td> </tr> <tr> <td>Input</td> <td>Name : GetShippingStatusIn, Message : GetShippingStatusIn</td> </tr> <tr> <td>Output</td> <td>Name : GetShippingStatusOut, Message :</td> </tr> <tr> <td>Verify</td> <td colspan="2">/ Pass</td> </tr> </table>			Service Name	GetOrderService (ID# GetOrderService)		Type	xxx		Message	Message Name : GetOrderListIn		Part	Part Name	Type	parameters	xsd:Interger	Message	Message Name : GetOrderListOut		Part	Part Name	Type	parameters	xsd:Interger	PortType	Opertion Name	GetOrderList	Input	Name : GetOrderListIn, Message : GetOrderListIn	Output	Name : GetOrderListOut, Message :	Binding			Sevice			Service Name	GetShippingStatusRequired (ID# GetShippingStatusRequired)		Opertion Call	OpertionCall Name	GetShippingStatus	Input	Name : GetShippingStatusIn, Message : GetShippingStatusIn	Output	Name : GetShippingStatusOut, Message :	Verify	/ Pass	
Service Name	GetOrderService (ID# GetOrderService)																																																		
Type	xxx																																																		
Message	Message Name : GetOrderListIn																																																		
Part	Part Name	Type																																																	
	parameters	xsd:Interger																																																	
Message	Message Name : GetOrderListOut																																																		
Part	Part Name	Type																																																	
	parameters	xsd:Interger																																																	
PortType	Opertion Name	GetOrderList																																																	
	Input	Name : GetOrderListIn, Message : GetOrderListIn																																																	
	Output	Name : GetOrderListOut, Message :																																																	
Binding																																																			
Sevice																																																			
Service Name	GetShippingStatusRequired (ID# GetShippingStatusRequired)																																																		
Opertion Call	OpertionCall Name	GetShippingStatus																																																	
	Input	Name : GetShippingStatusIn, Message : GetShippingStatusIn																																																	
	Output	Name : GetShippingStatusOut, Message :																																																	
Verify	/ Pass																																																		

รูปที่ 5.5 ตัวอย่างรายงาน คอมโพเนนท์ OrderTaker

5.4. สรุปผลการทดสอบ

หลังจากการทดสอบโครงสร้างของส่วนต่อขยายสถาปัตยกรรมบริการด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล และเครื่องมือที่พัฒนาขึ้น โดยทำการเปรียบเทียบกับการออกแบบโครงสร้างของระบบสั่งซื้อสินค้า ที่เป็นตัวอย่างในการทดสอบ ผลการทดสอบพบว่า โครงสร้างของส่วนต่อขยายขยายสถาปัตยกรรมบริการด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอดีแอล มีไวยากรณ์ของภาษาถูกต้องเป็นไปตามที่ได้ออกแบบไว้ตาม เอกซ์เอ็มแอลสคิม่า ที่สร้างขึ้น นอกจากนี้เครื่องมือที่พัฒนายังสามารถตรวจสอบความถูกต้องในการร้องขอบริการในกายอินเตอร์เฟซว่ามีการร้องขอตรงกับบริการที่มีอยู่ ได้อย่างถูกต้อง

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1. สรุปผลการวิจัย

วิทยานิพนธ์นี้ได้นำเสนอวิธีการพัฒนาส่วนต่อขยายภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล ให้สามารถอธิบายโครงสร้างของสถาปัตยกรรมเชิงบริการได้ การออกแบบส่วนต่อขยายแบ่งได้ออกเป็น 2 ส่วนหลักๆ ได้แก่ การพัฒนาส่วนต่อขยายภายในภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 เป็นส่วนเชื่อมโยงการอธิบายคุณสมบัติของแต่ละบริการที่อยู่ภายนอกอินเทอร์เฟซ โดยทำการเขียนเอกซ์เอ็มแอลสคิม่า เพื่อเพิ่มไวยากรณ์ที่ประกอบด้วยเอกซ์ลิงก์ (XLink) อยู่ในอินเทอร์เฟซของไวยากรณ์เดิมของภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 เดิม จำนวน 1 อินเทอร์เฟซ ประกอบไปด้วยอินเทอร์เฟซที่มีชื่อว่า soalink และการพัฒนาส่วนต่อขยายภายนอกเพื่ออธิบายคุณสมบัติของแต่ละบริการเป็นส่วนที่ใช้ในการอธิบายคุณสมบัติของแต่ละบริการที่ได้เชื่อมโยงมาจากเอกซ์ลิงก์ จากภายในอินเทอร์เฟซของโครงสร้างภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล 2.0 ที่ได้พัฒนาส่วนต่อขยายภายในเรียบร้อยแล้ว จำนวน 2 อินเทอร์เฟซหลัก ได้แก่ อินเทอร์เฟซ ServiceInterfaceProvided สำหรับข้อมูลอธิบายคุณสมบัติของผู้ให้บริการ และ อินเทอร์เฟซ ServiceInterfaceRequired สำหรับอธิบายคุณสมบัติของผู้ร้องขอบริการ ซึ่งจะต้องมีการเขียนเอกซ์เอ็มแอลสคิม่า ที่มีไวยากรณ์ที่ใช้ในการอธิบายคุณสมบัติของบริการซึ่งประกอบไปด้วย การอธิบายบริการที่มีให้ในแต่ละเซอวิสเซอคอมโพเนนท์ ซึ่งจะมีการแสดงเห็นถึงบริการที่เรียกร้องขอ คำสั่งที่เรียกใช้ และบริการที่ให้บริการ ซึ่งการออกแบบมีการอ้างอิงแนวคิดจากภาษาเอสไอเอเอ็มแอลและดับบลิวเอสดีแอล โดยครอบคลุมคุณสมบัติ การมีส่วนร่วมระหว่างบริการ ว่าเป็นบทบาทผู้ร้องขอหรือผู้ให้บริการ อินเทอร์เฟซของบริการจะสามารถอธิบายได้ในส่วนของการกำหนดคุณสมบัติด้วยโอเปอเรชันของบริการ ข้อมูลของบริการสามารถอธิบายได้ในการกำหนดรูปแบบของข้อมูลที่ใช้เป็นส่วนเข้าและส่วนออกของข้อมูล และทดสอบเครื่องมือประมวลผลโครงสร้างของสถาปัตยกรรมที่ใช้ภาษาเอกซ์เอตีแอลที่ได้รับการเพิ่มส่วนต่อขยายให้สามารถอธิบายรูปแบบสถาปัตยกรรมเชิงบริการ โดยในวิทยานิพนธ์นี้ ได้สร้างงานตัวอย่างระบบสั่งซื้อสินค้าขึ้นมาทดสอบ แล้วใช้เครื่องมือทำการทดสอบ ผลที่ได้คือโครงสร้างของส่วนต่อขยายสถาปัตยกรรมบริการด้วยภาษาอธิบาย

สถาปัตยกรรมเอกซ์เอตีแอล มีไวยากรณ์ของภาษาถูกต้องเป็นไปตามที่ได้ออกแบบไว้ตามเอกซ์เอ็มแอล สคีม่า ที่สร้างขึ้น นอกจากนี้เครื่องมือที่พัฒนายังสามารถตรวจสอบความถูกต้องในการร้องขอบริการ ในภายในเตอร์เฟซว่ามีการร้องขอตรงกับบริการที่มีอยู่ระบบ ได้อย่างถูกต้อง

6.2. ข้อจำกัดของงานวิจัย

- 1) วิทยานิพนธ์นี้ ใช้ภาษาอธิบายสถาปัตยกรรมโครงสร้างเอกซ์เอตีแอล 2.0
- 2) ออกแบบส่วนต่อขยายสถาปัตยกรรมเชิงบริการเครื่องมือ ArchStudio เวอร์ชัน 4 ในการ ออกแบบโครงสร้าง
- 3) ส่วนส่วนต่อขยายที่พัฒนาขึ้นออกแบบ เอกซ์เอ็มแอลสคีม่า ตามมาตรฐาน XML เวอร์ชัน 1.0

6.3. ข้อเสนอแนะ

- 1) วิทยานิพนธ์นี้ได้ออกแบบส่วนต่อขยายสถาปัตยกรรมเชิงบริการ ด้วยภาษาอธิบาย สถาปัตยกรรมเอกซ์เอตีแอล ในส่วนของโครงสร้างทางสถาปัตยกรรมเชิงบริการเท่านั้น ดังนั้น ควรปรับปรุงให้มีการเพิ่มเติมความสามารถในการอธิบายการทำงานของสถาปัตยกรรมเชิง บริการ ในรูปแบบของแบบจำลองเชิงพฤติกรรม (Behavioral Model) ของสถาปัตยกรรมเชิง บริการ จากส่วนต่อขยายที่พัฒนาขึ้น เพื่ออธิบายลำดับการไหลของข้อมูลในแต่ละคอม โปเนนท์ภายในโครงสร้างสถาปัตยกรรม จากแต่ละกิจกรรมระบบ
- 2) การตรวจสอบการร้องขอบริการภายในอินเตอร์เฟซควรมีการตรวจสอบความถูกต้อง นอกเหนือจากการตรวจสอบความเหมือนระหว่างชื่อโอเปอเรชัน ข้อมูลเข้า ข้อมูลออก ที่ เรียกใช้ กับที่มีให้บริการ ควรปรับปรุงให้สามารถตรวจสอบชนิดข้อมูล ข้อมูลเข้า ข้อมูลออก ที่มีชนิดข้อมูลที่มีความใกล้เคียงกันได้
- 3) ควรมีการพัฒนาเครื่องมือที่จะช่วยในสร้างไวยากรณ์เพื่อใช้ในการอธิบายคุณสมบัติของบริการ ได้โดยอัตโนมัติ เพื่อช่วยให้ผู้ที่นำไปใช้งานจริงสามารถนำไปใช้งานได้ง่ายขึ้น

6.4. ประโยชน์ที่ได้รับ

- 1) ทำให้ได้ภาษาอธิบายสถาปัตยกรรมเอกซ์เอ็ดดีแอล 2.0 ที่มีความสามารถในการอธิบายโครงสร้างของสถาปัตยกรรมบริการ ซึ่งจะช่วยให้นักออกแบบสถาปัตยกรรมซอฟต์แวร์ ลดจำนวนภาษาที่ใช้ในการออกแบบหรืออธิบายสถาปัตยกรรมที่มีรูปแบบที่หลากหลายให้ลดน้อยลง
- 2) เครื่องมือที่ได้จากวิทยานิพนธ์ สามารถช่วยในผู้ที่ออกแบบสถาปัตยกรรมซอฟต์แวร์ที่ใช้ส่วนต่อขยายสถาปัตยกรรมเชิงบริการที่สร้างขึ้นสามารถตรวจสอบความถูกต้องทั้งของไวยากรณ์และการเชื่อมต่อของบริการภายในโครงสร้างที่ออกแบบได้ง่ายขึ้น

รายการอ้างอิง

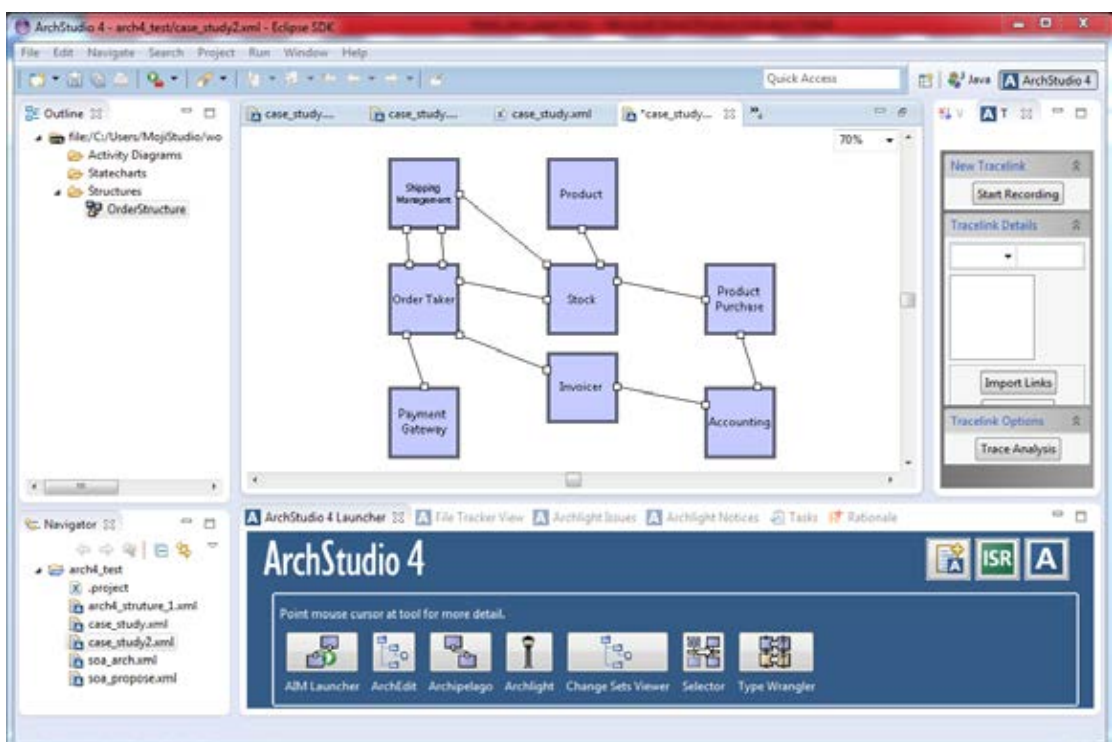
- [1] P. K. Surya, "xADL – A Better way to Describe Architecture." [Online]. Available from:<http://people.cs.clemson.edu/~johnmc/courses/cpsc875/resources/xadl.pdf>. [Accessed: 22-Aug-2012].
- [2] "SoaML 1.0.1." [Online]. Available from:
<http://www.omg.org/spec/SoaML/1.0.1/>. [Accessed: 29-Aug-2012].
- [3] A. Piromruen, "Compatibility Checking of Software Component in C2 Architectural Style on Protocol Level," Master's Thesis, Thammasat University, 2547.
- [4] J. Harkki, "Architecture Description Languages," 2004. [Online]. Available from:
<http://www.cs.tut.fi/~kk/webstuff/ArchitectureDescriptionLanguages.pdf>. [Accessed: 22-Aug-2012].
- [5] "xADL 2.0 - A Highly Extensible Architecture Description Language for Software and Systems." [Online]. Available from:
<http://www.isr.uci.edu/projects/archstudio-4/www/xarchuci/>. [Accessed: 08-Aug-2012].
- [6] ทวีติย์ เสนิงวงศ์ ณ อยุธยา and ยรรยง เต็งอำนาจ, "'ประกาศการณ์เอสโอเอ' ศูนย์เชี่ยวชาญเฉพาะทางด้านวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย," 2549.
- [7] A. Sadovykh, P. Desfray, B. Elvesaeter, A.-J. Berre, and E. Landre, "Enterprise architecture modeling with SoaML using BMM and BPMN - MDA approach in practice," in *Software Engineering Conference (CEE-SECR), 2010 6th Central and Eastern European*, 2010, pp. 79 –85.
- [8] "Web Service Definition Language (WSDL)." [Online]. Available from:
http://www.w3.org/TR/wsd1#_introduction. [Accessed: 10-Sep-2013].
- [9] "XML Technology - W3C." [Online]. Available from:
<http://www.w3.org/standards/xml/>. [Accessed: 28-Aug-2012].

- [10] “XML.” [Online]. Available from:
<http://multimedia.udru.ac.th/homeecs1/WorkProjeet/xml.html>. [Accessed: 25-Sep-2012].
- [11] “Associating Schemas with XML documents 1.0 (Second Edition).” [Online]. Available from: <http://www.w3.org/TR/2011/NOTE-xml-model-20110811/>. [Accessed: 28-Aug-2012].
- [12] “XMLSpy XML Editor.” [Online]. Available from:
<http://www.altova.com/xmlspy.html>. [Accessed: 02-Oct-2013].
- [13] “XML Linking Language (XLink) Version 1.1.” [Online]. Available from:
<http://www.w3.org/TR/xlink11/>. [Accessed: 02-Oct-2013].
- [14] L. Naslavsky, L. Xu, M. Dias, D. J. Richardson, and H. Ziv, “Extending xADL with Statechart Behavioral Specification,” presented at the Workshop on Architecting Dependable Systems, Edimburgh, Scotland, UK.
- [15] X. Jia, S. Ying, H. Cao, and D. Xie, “A New Architecture Description Language for Service-Oriented Architec,” in *Sixth International Conference on Grid and Cooperative Computing, 2007. GCC 2007, 2007*, pp. 96 –103.
- [16] M. N. Miladi, I. Krichen, M. Jmaiel, and K. Drira, “An xADL extension for managing dynamic deployment in distributed service oriented architectures,” in *Proceedings of the Third IPM international conference on Fundamentals of Software Engineering*, Berlin, Heidelberg, 2010, pp. 439–446.
- [17] L. Fuentes and N. Gámez, “Adding Aspects to xADL 2.0 for Software Product Line Architectures.” [Online]. Available from: http://www.sse.uni-due.de/vamos/2007/files/vamos07_0033_paper_9.pdf. [Accessed: 22-Aug-2012].

ภาคผนวก

ผลลัพธ์จากตัวอย่างทดสอบระบบสั่งซื้อสินค้า

ผลลัพธ์ออกแบบโครงสร้างและความสัมพันธ์ของบริการจากรายละเอียดตัวอย่างทดสอบ
ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล โดยใช้เครื่องมือ ArchStudio



รูปที่ ก-1 โครงสร้างสถาปัตยกรรมของตัวอย่างทดสอบ ด้วยภาษาอธิบายสถาปัตยกรรมเอกซ์เอตีแอล
โดยใช้เครื่องมือ ArchStudio

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<instance:xArch xmlns:instance="http://www.ics.uci.edu/pub/arch/xArch/instance.xsd"
xmlns:hints3="http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd"
xmlns:types="http://www.ics.uci.edu/pub/arch/xArch/types.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:soa="http://www.mojistudio.com/xadl4soa/soalink"
xsi:schemaLocation="http://www.ics.uci.edu/pub/arch/xArch/hints3.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/hints3.xsd
http://www.ics.uci.edu/pub/arch/xArch/instance.xsd
http://www.isr.uci.edu/projects/xarchuci/core/instance.xsd
http://www.ics.uci.edu/pub/arch/xArch/statecharts.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/statecharts.xsd
http://www.ics.uci.edu/pub/arch/xArch/activitydiagrams.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/activitydiagrams.xsd
http://www.ics.uci.edu/pub/arch/xArch/types.xsd http://www.isr.uci.edu/projects/xarchuci/ext/types.xsd
http://www.ics.uci.edu/pub/arch/xArch/changesets.xsd
http://www.isr.uci.edu/projects/xarchuci/ext/changesets.xsd http://www.mojistudio.com/xadl4soa/soalink
http://www.mojistudio.com/demo/xadl4soa/soalink.xsd">

```

```

  <types:archStructure types:id="archStructureffa80107-929d...>
  xsi:type="types:ArchStructure">

```

Component Shipping Management

```

  <types:description xsi:type="instance:Description">Order</types:description>
  <types:component types:id="componentffa80107-929da55d-ed4029a3-c2760005"
  xsi:type="types:Component">
    <types:description xsi:type="instance:Description">Shipping Management</types:description>
    <types:interface types:id="interfaceffa80107-92a0be98-e443b263-c27602e3"
  xsi:type="soa:SoaInterfaceType">
      <types:description xsi:type="instance:Description">ReqOrderList</types:description>
      <types:direction xsi:type="instance:Direction">none</types:direction>
      <soa:soalink xlink:href="ShippingManagementService.xml#GetOrderListRequired"
  xlink:type="simple" xsi:type="instance:XMLLink"/>
    </types:interface>
    <types:interface types:id="interfaceffa80107-92a23bac-a67e45e8-c276035c"
  xsi:type="soa:SoaInterfaceType">
      <types:description xsi:type="instance:Description">ShippingStatus</types:description>
      <types:direction xsi:type="instance:Direction">none</types:direction>
      <soa:soalink xlink:href="ShippingManagementService.xml#GetShipingStatusService"
  xlink:type="simple" xsi:type="instance:XMLLink"/>
    </types:interface>
    <types:interface types:id="interfaceffa80107-92a37953-c3adc011-c2760397"
  xsi:type="soa:SoaInterfaceType">
      <types:description xsi:type="instance:Description">ReqDecreaseStock</types:description>
      <types:direction xsi:type="instance:Direction">none</types:direction>
      <soa:soalink xlink:href="ShippingManagementService.xml#DecreaseStockRequired"
  xlink:type="simple" xsi:type="instance:XMLLink"/>
    </types:interface>
  </types:component>

```

Component Product

```

  <types:component types:id="componentffa80107-929e010d-971bb8f2-c2760029"
  xsi:type="types:Component">
    <types:description xsi:type="instance:Description">Product</types:description>
    <types:interface types:id="interfaceffa80107-92a5388e-566c331e-c276040e"
  xsi:type="types:Interface">
      <types:description xsi:type="instance:Description">[New Interface]</types:description>
      <types:direction xsi:type="instance:Direction">none</types:direction>
    </types:interface>
  </types:component>

```

Component OrderTaker

```

  <types:component types:id="componentffa80107-929e369c-5c15710c-c276004d"
  xsi:type="types:Component">
    <types:description xsi:type="instance:Description">Order Taker</types:description>
    <types:interface types:id="interfaceffa80107-92a077c4-49692544-c27602a9"
  xsi:type="soa:SoaInterfaceType">
      <types:description xsi:type="instance:Description">OrderList</types:description>
      <types:direction xsi:type="instance:Direction">none</types:direction>
      <soa:soalink xlink:href="OrderTakerService.xml#GetOrderService" xlink:type="simple"

```

Component OrderTaker

```

xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92a1d333-08930f4f-c276033b"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqShippingStatus</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="OrderTakerService.xml#GetShippingStatusRequired"
xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92aa131a-72aa45a0-c2760596"
xsi:type="types:Interface">
  <types:description xsi:type="instance:Description">ReqInsertInvoice</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92ab03d1-72c832cb-c27605e7"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqPayCredit</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="OrderTakerService.xml#PayCreditRequired" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92ac0fba-74b3ecd3-c2760637"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">ReqCheckStock</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="OrderTakerService.xml#CheckStockRequired" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>
  </types:component>

```

Component Stock

```

<types:component types:id="componentffa80107-929e6b54-763583ae-c2760071"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Stock</types:description>
  <types:interface types:id="interfaceffa80107-92a44c9c-55725949-c27603bb"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">DecreaseStock</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="StockService.xml#DecreaseStockService" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92a51a7b-3291b8d0-c27603ef"

```

```

xsi:type="types:Interface">
  <types:description xsi:type="instance:Description">[New Interface]</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92a62212-fb429a5e-c276043d"
xsi:type="types:Interface">
  <types:description xsi:type="instance:Description">[New Interface]</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92abd54b-8c5c4152-c2760617"
xsi:type="soa:SoaInterfaceType">
  <types:description xsi:type="instance:Description">CheckStock</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="StockService.xml#CheckStockService"
xsi:type="instance:XMLLink"/>
  </types:interface>

```

Component ProductPurchase

```

</types:component>
<types:component types:id="componentffa80107-929e9325-4cb8cadc-c2760095"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Product Purchase</types:description>
  <types:interface types:id="interfaceffa80107-92a63d83-f827c0de-c276045d"
xsi:type="types:Interface">
  <types:description xsi:type="instance:Description">[New Interface]</types:description>

```

Component ProductPurchase (Cont)

```

</types:direction xsi:type="instance:Direction">none</types:direction>
</types:interface>
<types:interface types:id="interfaceffa80107-92a8481a-8429086f-c27604ce"
xsi:type="soa:SoaInterfaceType">
  <types:description
xsi:type="instance:Description">ReqAddAccountPayable</types:description>
  <types:direction xsi:type="instance:Direction">none</types:direction>
  <soa:soalink xlink:href="ProductPurchaseService.xml#AddAccountsPayableRequired"
xlink:type="simple" xsi:type="instance:XMLLink"/>
</types:interface>
</types:component>

```

Component Payment Gateway

```

<types:component types:id="componentffa80107-929f361c-fe6e704d-c2760124"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Payment Gateway</types:description>
  <types:interface types:id="interfaceffa80107-92aac11-38cbea5c-c27605c6"
xsi:type="soa:SoaInterfaceType">
    <types:description xsi:type="instance:Description">PayCredit</types:description>
    <types:direction xsi:type="instance:Direction">none</types:direction>
    <soa:soalink xlink:href="PaymentGatewayService.xml#PayCreditService" xlink:type="simple"
xsi:type="instance:XMLLink"/>
  </types:interface>
</types:component>

```

Component Invoicer

```

<types:component types:id="componentffa80107-929f6ab4-e4a81866-c2760148"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Invoicer</types:description>
  <types:interface types:id="interfaceffa80107-92a9379a-3f417a90-c2760540"
xsi:type="soa:SoaInterfaceType">
    <types:description
xsi:type="instance:Description">ReqAddAccountsReceivable</types:description>
    <types:direction xsi:type="instance:Direction">none</types:direction>
    <soa:soalink xlink:href="InvoicerService.xml#AddAccountsReceivableRequired"
xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92a9de79-ff51ea47-c2760576"
xsi:type="types:Interface">
    <types:description xsi:type="instance:Description">InsertInvoice</types:description>
    <types:direction xsi:type="instance:Direction">none</types:direction>
  </types:interface>
</types:component>

```

Component Accounting

```

<types:component types:id="componentffa80107-929fa3ec-09f14fd6-c276016c"
xsi:type="types:Component">
  <types:description xsi:type="instance:Description">Accounting</types:description>
  <types:interface types:id="interfaceffa80107-92a7ccb3-8e1129e2-c27604ae"
xsi:type="soa:SoaInterfaceType">
    <types:description xsi:type="instance:Description">AddAccountPayable</types:description>
    <types:direction xsi:type="instance:Direction">none</types:direction>
    <soa:soalink xlink:href="AccountingService.xml#AddAccountsPayableService"
xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:interface>
  <types:interface types:id="interfaceffa80107-92a8d541-0f1de0bd-c2760520"
xsi:type="soa:SoaInterfaceType">
    <types:description
xsi:type="instance:Description">AddAccountsReceivable</types:description>
    <types:direction xsi:type="instance:Direction">none</types:direction>
    <soa:soalink xlink:href="AccountingService.xml#AddAccountsReceivableService"
xlink:type="simple" xsi:type="instance:XMLLink"/>
  </types:interface>
</types:component>

```

```

<types:link types:id="linkffa80107-92a0f7dc-e1e5145f-c27602eb" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a0be98-e443b263-c27602e3"
xsi:type="instance:XMLLink" xlink:type="simple"/>
  </types:point>
</types:link>

```

```

</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a077c4-49692544-c27602a9"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a274d4-2df53d32-c2760364" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a23bac-a67e45e8-c276035c"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a1d333-08930f4f-c276033b"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a47ef3-e97406ea-c27603c5" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a37953-c3adc011-c2760397"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a44c9c-55725949-c27603bb"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a5c24a-ff90b0d5-c2760415" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a5388e-566c331e-c276040e"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a51a7b-3291b8d0-c27603ef"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a6519a-c58ecf84-c2760464" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a62212-fb429a5e-c276043d"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a63d83-f827c0de-c276045d"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a8a143-c5dca6f7-c27604f1" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a8481a-8429086f-c27604ce"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a7ccb3-8e1129e2-c27604ae"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92a981fa-06107a76-c2760547" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>

```

```

<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a9379a-3f417a90-c2760540"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
<types:point xsi:type="instance:Point">
  <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a8d541-0f1de0bd-c2760520"
xsi:type="instance:XMLLink" xlink:type="simple"/>
</types:point>
</types:link>
<types:link types:id="linkffa80107-92aa74b9-01c2d287-c276059f" xsi:type="types:Link">
  <types:description xsi:type="instance:Description">[New Link]</types:description>
  <types:point xsi:type="instance:Point">
    <instance:anchorOnInterface xlink:href="#interfaceffa80107-92aa131a-72aa45a0-c2760596"
xsi:type="instance:XMLLink" xlink:type="simple"/>
    </types:point>
    <types:point xsi:type="instance:Point">
      <instance:anchorOnInterface xlink:href="#interfaceffa80107-92a9de79-ff51ea47-c2760576"
xsi:type="instance:XMLLink" xlink:type="simple"/>
      </types:point>
    </types:link>
    <types:link types:id="linkffa80107-92ab46da-6f9b41cb-c27605ee" xsi:type="types:Link">
      <types:description xsi:type="instance:Description">[New Link]</types:description>
      <types:point xsi:type="instance:Point">
        <instance:anchorOnInterface xlink:href="#interfaceffa80107-92ab03d1-72c832cb-c27605e7"
xsi:type="instance:XMLLink" xlink:type="simple"/>
        </types:point>
        <types:point xsi:type="instance:Point">
          <instance:anchorOnInterface xlink:href="#interfaceffa80107-92aaac11-38cbea5c-c27605c6"
xsi:type="instance:XMLLink" xlink:type="simple"/>
          </types:point>
        </types:link>
        <types:link types:id="linkffa80107-92ac4c38-466ef613-c276063e" xsi:type="types:Link">
          <types:description xsi:type="instance:Description">[New Link]</types:description>
          <types:point xsi:type="instance:Point">
            <instance:anchorOnInterface xlink:href="#interfaceffa80107-92ac0fba-74b3ecd3-c2760637"
xsi:type="instance:XMLLink" xlink:type="simple"/>
            </types:point>
            <types:point xsi:type="instance:Point">
              <instance:anchorOnInterface xlink:href="#interfaceffa80107-92abd54b-8c5c4152-c2760617"
xsi:type="instance:XMLLink" xlink:type="simple"/>
              </types:point>
            </types:link>
          </types:archStructure>

```

รูปที่ ก-2 รายละเอียดโครงสร้างสถาปัตยกรรมระบบสั่งซื้อสินค้า

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xsd14soa/XADE4SOA\Schema.xsd"
  >
  <ServiceInterfaceProvided ServiceName="GetOrderService" Id="GetOrderService">
    <Types>
      <element name="GetOrderList">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="DD" type="xsd:short"/>
            <element minOccurs="1" maxOccurs="1" name="MM" type="xsd:short"/>
            <element minOccurs="1" maxOccurs="1" name="YYYY" type="xsd:short"/>
          </sequence>
        </complexType>
      </element>
      <element name="GetOrderListResponse">
        <complexType>
          <sequence>
            <element minOccurs="1" maxOccurs="1" name="OrderListResult" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
    </Types>
    <Message Name="GetOrderListIn">
      <Part Name="parameters" Type="xsd:Interger" element="GetOrderList"/>
    </Message>
    <Message Name="GetOrderListOut">
      <Part Name="parameters" Type="xsd:Interger" element="GetOrderListResponse"/>
    </Message>
    <PortType>
      <Operation OperationName="GetOrderList">
        <Input Name="GetOrderListIn" Message="GetOrderListIn"/>
        <Output Message="GetOrderListOut" Name="GetOrderListOut"/>
      </Operation>
    </PortType>
    <Binding Name="GetOrderService" Type=""/>
    <Service Name="GetOrderService"/>
  </ServiceInterfaceProvided>
  <ServiceInterfaceRequired ServiceName="GetShippingStatusRequired"
    Id="GetShippingStatusRequired">
    <OperationCall OperationName="GetShippingStatus">
      <Input Name="GetShippingStatusIn" Message="GetShippingStatusIn"/>
      <Output Name="GetShippingStatusOut" Message="GetShippingStatusOut"/>
    </OperationCall>
  </ServiceInterfaceRequired>
  <ServiceInterfaceRequired ServiceName="CheckStockRequired" Id="CheckStockRequired">
    <OperationCall OperationName="CheckStock">
      <Input Name="CheckStockIn" Message="CheckStockIn"/>
      <Output Name="CheckStockOut" Message="CheckStockIn"/>
    </OperationCall>
  </ServiceInterfaceRequired>
  <ServiceInterfaceRequired ServiceName="PayCreditRequired" Id="PayCreditRequired">
    <OperationCall OperationName="PayCredit">
      <Input Name="PayCreditIn" Message="PayCreditIn"/>
      <Output Name="PayCreditOut" Message="PayCreditOut"/>
    </OperationCall>
  </ServiceInterfaceRequired>
</ServiceInterface>

```

รูปที่ ก-3 รายละเอียดบริการภายในคอมพิวเตอร์ OrderTaker


```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/dome/xsd/4soa/4soa.xsd"
<ServiceInterfaceProvided ServiceName="GetShipingStatusService" Id="GetShipingStatusService">
  <Types>
    <element name="GetShipingStatus">
      <complexType>
        <sequence>
          <element minOccurs="1" maxOccurs="1" name="ShippingId" type="xsd:interger"/>
        </sequence>
      </complexType>
    </element>
    <element name="GetShipingStatusResponse">
      <complexType>
        <sequence>
          <element minOccurs="1" maxOccurs="1" name="ShippingId" type="xsd:interger"/>
          <element minOccurs="1" maxOccurs="1" name="Status" type="xsd:interger"/>
        </sequence>
      </complexType>
    </element>
  </Types>
  <Message Name="GetShipingStatusIn">
    <Part Name="parameters" element="GetShipingStatus"/>
  </Message>
  <Message Name="GetShipingStatusOut">
    <Part Name="parameters" element="GetShipingStatusResponse"/>
  </Message>
  <PortType>
    <Operation OperationName="GetShipingStatus">
      <Input Name="GetShipingStatusIn" Message="GetShipingStatusIn"/>
      <Output Message="GetShipingStatusOut" Name="GetShipingStatusOut"/>
    </Operation>
  </PortType>
  <Binding Name="GetShipingStatusService" Type=""/>
  <Service Name="GetShipingStatusService"/>
</ServiceInterfaceProvided>
</ServiceInterfaceRequired>
<ServiceInterfaceRequired ServiceName="GetOrderListRequired" Id="GetOrderListRequired">
  <OperationCall OperationName="GetOrderList">
    <Input Name="GetOrderListIn" Message="GetShipingStatusIn"/>
    <Output Name="GetOrderListOut" Message="GetShipingStatusOut"/>
  </OperationCall>
</ServiceInterfaceRequired>
<ServiceInterfaceRequired ServiceName="DecreaseStockRequired" Id="DecreaseStockRequired">
  <OperationCall OperationName="DecreaseStock">
    <Input Name="DecreaseStockIn" Message="DecreaseStockIn"/>
    <Output Name="DecreaseStockOut" Message="DecreaseStockOut"/>
  </OperationCall>
</ServiceInterfaceRequired>
</ServiceInterface>

```

รูปที่ ก-4 รายละเอียดบริการภายในคอมพิวเตอร์ Shipping Management

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xadl4soa/XADL4SOASchema.xsd"
  <ServiceInterfaceRequired ServiceName="AddAccountsReceivableRequired"
  Id="AddAccountsReceivableRequired">
    <OperationCall OperationName="AddAccountsReceivable">
      <Input Name="AddAccountsReceivableIn" Message="AddAccountsReceivableIn"/>
      <Output Name="AddAccountsReceivableOut" Message="AddAccountsReceivableOut"/>
    </OperationCall>
  </ServiceInterfaceRequired>
</ServiceInterface>

```

รูปที่ ก-5 รายละเอียดบริการภายในคอมพิวเตอร์ Invoicer

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xadl4soa/
  <ServiceInterfaceRequired ServiceName="AddAccountsPayableRequired"
  Id="AddAccountsPayableRequired">
    <OperationCall OperationName="AddAccountsPayable">
      <Input Name="AddAccountsPayableIn" Message="AddAccountsPayableIn"/>
      <Output Name="AddAccountsPayableOut" Message="AddAccountsPayableOut"/>
    </OperationCall>
  </ServiceInterfaceRequired>
</ServiceInterface>

```

รูปที่ ก-6 รายละเอียดบริการภายในคอมพิวเตอร์ ProductPurchase

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xadl4soa/"
PayCreditService
<ServiceInterfaceProvided ServiceName="PayCreditService" Id="PayCreditService">
  <Types>
    <element name="PayCreditService">
      <complexType>
        <sequence>
          <element name="OrderId" type="xsd:integer"/>
          <element name="CreditCode" type="xsd:string"/>
          <element name="Amount" type="xsd:float"/>
        </sequence>
      </complexType>
    </element>
    <element name="PayCreditServiceResponse">
      <complexType>
        <sequence>
          <element name="PaymentCode" type="xsd:integer"/>
          <element name="Amount" type="xsd:float"/>
          <element name="Status" type="xsd:integer"/>
        </sequence>
      </complexType>
    </element>
  </Types>
  <Message Name="PayCreditServiceIn">
    <Part Name="parameters" element="PayCreditService"/>
  </Message>
  <Message Name="PayCreditServiceOut">
    <Part Name="parameters" element="PayCreditServiceResponse"/>
  </Message>
  <PortType>
    <Operation OperationName="PayCredit">
      <Input Name="PayCreditServiceIn" Message="PayCreditServiceIn"/>
      <Output Message="PayCreditServiceOut" Name="PayCreditServiceIn"/>
    </Operation>
  </PortType>
  <Binding Name="PayCreditService" Type=""/>
  <Service Name="PayCreditService"/>
</ServiceInterfaceProvided>
</ServiceInterface>

```

รูปที่ ก-7 รายละเอียดบริการภายในคอมโพเนนท์ Payment Gateway

```

<?xml version="1.0" encoding="UTF-8"?>
<ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xsd/4soa/
<ServiceInterfaceProvided ServiceName="CheckStockService" Id="CheckStockService">
  <Types>
    <element name="CheckStockStatus">
      <complexType>
        <sequence>
          <element minOccurs="1" maxOccurs="1" name="ProductId" type="xsd:integer"/>
        </sequence>
      </complexType>
    </element>
    <element name="CheckStockResponse">
      <complexType>
        <sequence>
          <element name="ProductId" type="xsd:integer"/>
          <element name="ProductName" type="xsd:string"/>
          <element name="Qty" type="xsd:integer"/>
        </sequence>
      </complexType>
    </element>
  </Types>
  <Message Name="CheckStockIn">
    <Part Name="parameters" element="CheckStock"/>
  </Message>
  <Message Name="CheckStockOut">
    <Part Name="parameters" element="CheckStockResponse"/>
  </Message>
  <PortType>
    <Operation OperationName="CheckStock">
      <Input Name="CheckStockIn" Message="CheckStockIn"/>
      <Output Message="CheckStockOut" Name="CheckStockOut"/>
    </Operation>
  </PortType>
  <Binding Name="CheckStockService" Type=""/>
  <Service Name="CheckStockService"/>
</ServiceInterfaceProvided>

```

CheckStockService

DecreaseStockService

```

<ServiceInterfaceProvided ServiceName="DecreaseStockService" Id="DecreaseStockService">
  <Types>
    <element name="DecreaseStock">
      <complexType>
        <sequence>
          <element name="ProductId" type="xsd:integer"/>
          <element name="Qty" type="xsd:integer"/>
        </sequence>
      </complexType>
    </element>
    <element name="DecreaseStockResponse">
      <complexType>
        <sequence>
          <element name="Status" type="xsd:integer"/>
        </sequence>
      </complexType>
    </element>
  </Types>
  <Message Name="DecreaseStockIn">
    <Part Name="parameters" element="DecreaseStock"/>
  </Message>

```

<pre> <Message Name="DecreaseStockOut"> <Part Name="parameters" element="DecreaseStockResponse"/> </Message> <PortType> <Operation OperationName="DecreaseStock"> <Input Name="DecreaseStockIn" Message="DecreaseStockIn"/> <Output Message="DecreaseStockOut" Name="DecreaseStockOut"/> </Operation> </PortType> <Binding Name="DecreaseStockService" Type=""/> <Service Name="DecreaseStockService"/> </ServiceInterfaceProvided> </ServiceInterface> </pre>	DecreaseStockService (Cont)
--	-----------------------------

รูปที่ ก-8 รายละเอียดบริการภายในคอมพิวเตอร์ Stock

<pre> <?xml version="1.0" encoding="UTF-8"?> <ServiceInterface xmlns:xsi="http://www.w3.org/2001/XMLSchema- xsi:noNamespaceSchemaLocation="http://mojistudio.com/demo/xac <ServiceInterfaceProvided ServiceName="AddAccountsReceivableService" Id="AddAccountsReceivableService"> <Types> <element name="AddAccountsReceivable"> <complexType> <sequence> <element name="Description" type="xsd:string"/> <element name="TotalReceive" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsReceivableResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types> <Message Name="AddAccountsReceivableIn"> <Part Name="parameters" element="AddAccountsReceivable"/> </Message> <Message Name="AddAccountsReceivableOut"> <Part Name="parameters" element="AddAccountsReceivableResponse"/> </Message> <PortType> <Operation OperationName="AddAccountsReceivable"> <Input Name="AddAccountsReceivableIn" Message="AddAccountsReceivableIn"/> <Output Message="AddAccountsReceivableOut" Name="AddAccountsReceivableOut"/> </Operation> </PortType> <Binding Name="AddAccountsReceivableService" Type=""/> <Service Name="AddAccountsReceivableService"/> </ServiceInterfaceProvided> </pre>	AddAccountsReceivableService
<pre> <ServiceInterfaceProvided ServiceName="AddAccountsPayableService" Id="AddAccountsPayableService"> <Types> <element name="AddAccountsPayable"> </pre>	AddAccountsPayableService

<pre><complexType> <sequence></pre>	AddAccountsPayableService (cont)
<pre> <element name="Description" type="xsd:string"/> <element name="TotalPayable" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsPayableResponse"> <complexType> <sequence> <element name="Status" type="xsd:integer"/> </sequence> </complexType> </element> </Types> <Message Name="AddAccountsPayableIn"> <Part Name="parameters" element="AddAccountsPayable"/> </Message> <Message Name="AddAccountsPayableOut"> <Part Name="parameters" element="AddAccountsPayableResponse"/> </Message> <PortType> <Operation OperationName="AddAccountsPayable"> <Input Name="AddAccountsPayableIn" Message="AddAccountsPayableIn"/> <Output Message="AddAccountsPayableOut" Name="AddAccountsPayableOut"/> </Operation> </PortType> <Binding Name="AddAccountsPayableService" Type=""/> <Service Name="AddAccountsPayableService"/> </ServiceInterfaceProvided> </pre>	
<pre></ServiceInterface></pre>	

รูปที่ ก-9 รายละเอียดบริการภายในคอมพิวเตอร์ Accounting

xADL4SOA Extension Structure Report

File name : OrderCaseStudy.xml

Component Name:	Shipping Management		
	<ul style="list-style-type: none"> • Total Interface = 3 • Provided Interface = 1 • Required Interface = 2 • NonSOA Interface = 0 		
Interface :	Service Name	GetOrderListRequired (ID# GetOrderListRequired)	
	Operation Call	Operation Call Name	GetOrderList
		Input	Name : GetOrderListIn, Message : GetShippingStatusIn
		Output	Name : GetOrderListOut, Message :
	Verify	/ Pass	
	Service Name	GetShipingStatusService (ID# GetShipingStatusService)	
	Type	<pre><Types> <element name="GetShipingStatus"> <complexType> <sequence> <element maxOccurs="1" minOccurs="1" name="ShippingId" type="xsd:interger"/> </sequence> </complexType> </element> <element name="GetShipingStatusResponse"> <complexType> <sequence> <element maxOccurs="1" minOccurs="1" name="ShippingId" type="xsd:interger"/> <element maxOccurs="1" minOccurs="1" name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre>	
	Message	Message Name : GetShipingStatusIn	
	Part	Part Name	Type
		parameters	
	Message	Message Name : GetShipingStatusOut	
	Part	Part Name	Type
		parameters	
	PortType	Operation Name	GetShipingStatus
		Input	Name : GetShipingStatusIn, Message : GetShipingStatusIn
		Output	Name : GetShipingStatusOut, Message :
	Binding Service		
	Service Name	DecreaseStockRequired (ID# DecreaseStockRequired)	
	Operation Call	Operation Call Name	DecreaseStock
		Input	Name : DecreaseStockIn, Message : DecreaseStockIn
	Output	Name : DecreaseStockOut, Message :	

		Verify	/ Pass
Component Name:	Product		
	<ul style="list-style-type: none"> • Total Interface = 1 • Provided Interface = 0 • Required Interface = 0 • NonSOA Interface = 1 		
Interface :			
Component Name:	Order Taker		
	<ul style="list-style-type: none"> • Total Interface = 5 • Provided Interface = 1 • Required Interface = 3 • NonSOA Interface = 1 		
Interface :	Service Name	GetOrderService (ID# GetOrderService)	
	Type	<pre><Types> <element name="GetOrderList"> <complexType> <sequence> <element maxOccurs="1" minOccurs="1" name="DD" type="xsd:short"/> <element maxOccurs="1" minOccurs="1" name="MM" type="xsd:short"/> <element maxOccurs="1" minOccurs="1" name="YYYY" type="xsd:short"/> </sequence> </complexType> </element> <element name="GetOrderListResponse"> <complexType> <sequence> <element maxOccurs="1" minOccurs="1" name="OrderListResult" type="xsd:string"/> </sequence> </complexType> </element> </Types></pre>	
	Message	Message Name : GetOrderListIn	
	Part	Part Name	Type
		parameters	xsd:Integer
	Message	Message Name : GetOrderListOut	
	Part	Part Name	Type
		parameters	xsd:Integer
	PortType	Operation Name	GetOrderList
		Input	Name : GetOrderListIn, Message : GetOrderListIn
		Output	Name : GetOrderListOut, Message :
	Binding		
	Service		
	Service Name	GetShippingStatusRequired (ID# GetShippingStatusRequired)	
	Operation Call	OperationCall Name	GetShippingStatus
		Input	Name : GetShippingStatusIn, Message : GetShippingStatusIn
		Output	Name : GetShippingStatusOut, Message :
	Verify	X Operation is not match	
	Service Name	PayCreditRequired (ID# PayCreditRequired)	
	Operation Call	OperationCall Name	PayCredit
		Input	Name : PayCreditIn, Message : PayCreditIn
		Output	Name : PayCreditOut, Message :

	Verify	X Output is not match	
	Service Name	CheckStockRequired (ID# CheckStockRequired)	
	Opertion Call	OpertionCall Name	CheckStock
		Input	Name : CheckStockIn, Message : CheckStockIn
		Output	Name : CheckStockOut, Message :
	Verify	/ Pass	
Component Name:	Stock		
	<ul style="list-style-type: none"> • Total Inteface = 4 • Provided Inteface = 2 • Required Inteface = 0 • NonSOA Inteface = 2 		
Interface :	Service Name	DecreaseStockService (ID# DecreaseStockService)	
	Type	<pre><Types> <element name="DecreaseStock"> <complexType> <sequence> <element name="ProductId" type="xsd:interger"/> <element name="Qty" type="xsd:interger"/> </sequence> </complexType> </element> <element name="DecreaseStockResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre>	
	Message	Message Name : DecreaseStockIn	
	Part	Part Name	Type
		parameters	
	Message	Message Name : DecreaseStockOut	
	Part	Part Name	Type
		parameters	
	PortType	Opertion Name	DecreaseStock
		Input	Name : DecreaseStockIn, Message : DecreaseStockIn
		Output	Name : DecreaseStockOut, Message :
	Binding Service		
	Service Name	CheckStockService (ID# CheckStockService)	
	Type	<pre><Types> <element name="CheckStockStatus"> <complexType> <sequence> <element maxOccurs="1" minOccurs="1" name="ProductId" type="xsd:interger"/> </sequence> </complexType> </element> <element name="CheckStockResponse"> <complexType> <sequence> <element name="ProductId" type="xsd:interger"/> <element name="ProductName" type="xsd:string"/> </sequence> </complexType> </element> </Types></pre>	

		<element name="Qty" type="xsd:interger"/> </sequence> </complexType> </element> </Types>	
Message	Message Name : CheckStockIn		
Part	Part Name	Type	
	parameters		
Message	Message Name : CheckStockOut		
Part	Part Name	Type	
	parameters		
PortType	Opertion Name	CheckStock	
	Input	Name : CheckStockIn, Message : CheckStockIn	
	Output	Name : CheckStockOut, Message :	
Binding			
Sevice			
Component Name:	Product Purchase		
	<ul style="list-style-type: none"> • Total Inteface = 2 • Provided Inteface = 0 • Required Inteface = 1 • NonSOA Inteface = 1 		
Interface :	Service Name	AddAccountsPayableRequired (ID# AddAccountsPayableRequired)	
	Opertion Call	Opertion Call Name	AddAccountsPayable
		Input	Name : AddAccountsPayableIn, Message : AddAccountsPayableIn
		Output	Name : AddAccountsPayableOut, Message :
	Verify	/ Pass	
Component Name:	Payment Gateway		
	<ul style="list-style-type: none"> • Total Inteface = 1 • Provided Inteface = 1 • Required Inteface = 0 • NonSOA Inteface = 0 		
Interface :	Service Name	PayCreditService (ID# PayCreditService)	
	Type	<Types> <element name="PayCreditService"> <complexType> <sequence> <element name="OrderId" type="xsd:interger"/> <element name="CreditCode" type="xsd:string"/> <element name="Amount" type="xsd:float"/> </sequence> </complexType> </element> <element name="PayCreditServiceResponse"> <complexType> <sequence> <element name="PaymentCode" type="xsd:interger"/> <element name="Amount" type="xsd:float"/> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types>	
	Message	Message Name : PayCreditServiceIn	
	Part	Part Name	Type
		parameters	

	<table border="1"> <tr> <td>Message</td> <td colspan="3">Message Name : PayCreditServiceOut</td> </tr> <tr> <td rowspan="2">Part</td> <td>Part Name</td> <td>Type</td> <td></td> </tr> <tr> <td>parameters</td> <td></td> <td></td> </tr> <tr> <td rowspan="3">PortType</td> <td>Opertion Name</td> <td colspan="2">PayCredit</td> </tr> <tr> <td>Input</td> <td colspan="2">Name : PayCreditServiceIn, Message : PayCreditServiceIn</td> </tr> <tr> <td>Output</td> <td colspan="2">Name : PayCreditServiceIn, Message :</td> </tr> <tr> <td>Binding Service</td> <td colspan="3"></td> </tr> </table>	Message	Message Name : PayCreditServiceOut			Part	Part Name	Type		parameters			PortType	Opertion Name	PayCredit		Input	Name : PayCreditServiceIn, Message : PayCreditServiceIn		Output	Name : PayCreditServiceIn, Message :		Binding Service																						
Message	Message Name : PayCreditServiceOut																																												
Part	Part Name	Type																																											
	parameters																																												
PortType	Opertion Name	PayCredit																																											
	Input	Name : PayCreditServiceIn, Message : PayCreditServiceIn																																											
	Output	Name : PayCreditServiceIn, Message :																																											
Binding Service																																													
Component Name:	Invoicer																																												
	<ul style="list-style-type: none"> Total Inteface = 2 Provided Inteface = 0 Required Inteface = 1 NonSOA Inteface = 1 																																												
Interface :	<table border="1"> <tr> <td>Service Name</td> <td colspan="3">AddAccountsReceivableRequired (ID# AddAccountsReceivableRequired)</td> </tr> <tr> <td rowspan="3">Opertion Call</td> <td>OpertionCall Name</td> <td colspan="2">AddAccountsReceivable</td> </tr> <tr> <td>Input</td> <td colspan="2">Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn</td> </tr> <tr> <td>Output</td> <td colspan="2">Name : AddAccountsReceivableOut, Message :</td> </tr> <tr> <td>Verify</td> <td colspan="3">/ Pass</td> </tr> </table>	Service Name	AddAccountsReceivableRequired (ID# AddAccountsReceivableRequired)			Opertion Call	OpertionCall Name	AddAccountsReceivable		Input	Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn		Output	Name : AddAccountsReceivableOut, Message :		Verify	/ Pass																												
Service Name	AddAccountsReceivableRequired (ID# AddAccountsReceivableRequired)																																												
Opertion Call	OpertionCall Name	AddAccountsReceivable																																											
	Input	Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn																																											
	Output	Name : AddAccountsReceivableOut, Message :																																											
Verify	/ Pass																																												
Component Name:	Accounting																																												
	<ul style="list-style-type: none"> Total Inteface = 2 Provided Inteface = 2 Required Inteface = 0 NonSOA Inteface = 0 																																												
Interface :	<table border="1"> <tr> <td>Service Name</td> <td colspan="3">AddAccountsPayableService (ID# AddAccountsPayableService)</td> </tr> <tr> <td>Type</td> <td colspan="3"> <pre><Types> <element name="AddAccountsPayable"> <complexType> <sequence> <element name="Description" type="xsd:string"/> <element name="TotalPayable" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsPayableResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre> </td> </tr> <tr> <td>Message</td> <td colspan="3">Message Name : AddAccountsPayableIn</td> </tr> <tr> <td rowspan="2">Part</td> <td>Part Name</td> <td>Type</td> <td></td> </tr> <tr> <td>parameters</td> <td></td> <td></td> </tr> <tr> <td>Message</td> <td colspan="3">Message Name : AddAccountsPayableOut</td> </tr> <tr> <td rowspan="2">Part</td> <td>Part Name</td> <td>Type</td> <td></td> </tr> <tr> <td>parameters</td> <td></td> <td></td> </tr> <tr> <td rowspan="3">PortType</td> <td>Opertion Name</td> <td colspan="2">AddAccountsPayable</td> </tr> <tr> <td>Input</td> <td colspan="2">Name : AddAccountsPayableIn, Message : AddAccountsPayableIn</td> </tr> <tr> <td>Output</td> <td colspan="2">Name : AddAccountsPayableOut, Message :</td> </tr> <tr> <td>Binding</td> <td colspan="3"></td> </tr> </table>	Service Name	AddAccountsPayableService (ID# AddAccountsPayableService)			Type	<pre><Types> <element name="AddAccountsPayable"> <complexType> <sequence> <element name="Description" type="xsd:string"/> <element name="TotalPayable" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsPayableResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre>			Message	Message Name : AddAccountsPayableIn			Part	Part Name	Type		parameters			Message	Message Name : AddAccountsPayableOut			Part	Part Name	Type		parameters			PortType	Opertion Name	AddAccountsPayable		Input	Name : AddAccountsPayableIn, Message : AddAccountsPayableIn		Output	Name : AddAccountsPayableOut, Message :		Binding			
Service Name	AddAccountsPayableService (ID# AddAccountsPayableService)																																												
Type	<pre><Types> <element name="AddAccountsPayable"> <complexType> <sequence> <element name="Description" type="xsd:string"/> <element name="TotalPayable" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsPayableResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre>																																												
Message	Message Name : AddAccountsPayableIn																																												
Part	Part Name	Type																																											
	parameters																																												
Message	Message Name : AddAccountsPayableOut																																												
Part	Part Name	Type																																											
	parameters																																												
PortType	Opertion Name	AddAccountsPayable																																											
	Input	Name : AddAccountsPayableIn, Message : AddAccountsPayableIn																																											
	Output	Name : AddAccountsPayableOut, Message :																																											
Binding																																													

Service							
Service Name	AddAccountsReceivableService (ID# AddAccountsReceivableService)						
Type	<pre><Types> <element name="AddAccountsReceivable"> <complexType> <sequence> <element name="Description" type="xsd:string"/> <element name="TotalReceive" type="xsd:float"/> </sequence> </complexType> </element> <element name="AddAccountsReceivableResponse"> <complexType> <sequence> <element name="Status" type="xsd:interger"/> </sequence> </complexType> </element> </Types></pre>						
Message	Message Name : AddAccountsReceivableIn						
Part	<table border="1"> <thead> <tr> <th>Part Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>parameters</td> <td></td> </tr> </tbody> </table>	Part Name	Type	parameters			
Part Name	Type						
parameters							
Message	Message Name : AddAccountsReceivableOut						
Part	<table border="1"> <thead> <tr> <th>Part Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>parameters</td> <td></td> </tr> </tbody> </table>	Part Name	Type	parameters			
Part Name	Type						
parameters							
PortType	<table border="1"> <thead> <tr> <th>Opertion Name</th> <th>AddAccountsReceivable</th> </tr> </thead> <tbody> <tr> <td>Input</td> <td>Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn</td> </tr> <tr> <td>Output</td> <td>Name : AddAccountsReceivableOut, Message :</td> </tr> </tbody> </table>	Opertion Name	AddAccountsReceivable	Input	Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn	Output	Name : AddAccountsReceivableOut, Message :
Opertion Name	AddAccountsReceivable						
Input	Name : AddAccountsReceivableIn, Message : AddAccountsReceivableIn						
Output	Name : AddAccountsReceivableOut, Message :						
Binding Service							

รูปที่ ก-10 รายงานผลลัพธ์จากการประมวลผลของเครื่องมือ

ประวัติผู้เขียนวิทยานิพนธ์

นายภาสกร ปานนอก เกิดเมื่อวันพฤหัสบดี ที่ 31 เดือน สิงหาคม พุทธศักราช 2532 ที่ จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาบัณฑิต ในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ จากคณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา 2553 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554