กรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง

นายชนินทร เรืองอุดมสกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2554

SERVICE BASED FRAMEWORK FOR SUPPLY CHAIN TRACKING IN TRANSPORTATION

Chanintorn Ruangudomsakul

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Computer Engineering

Suranaree University of Technology

Academic Year 2011

กรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

_____

(ผศ. ดร.พิชโยทัย มหัทธนาภิวัฒน์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(ผศ. ดร.ปรเมศวร์ ห่อแก้ว)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี

_____

(ศ. ดร.ชูกิจ ลิมปีจำนงค์)
รองอธิการบดีฝ่ายวิชาการ

_____

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

ชนินทร เรืองอุดมสกุล : กรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง (SERVICE BASED FRAMEWORK FOR SUPPLY CHAIN TRACKING IN TRANSPORTATION) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. พิชโยทัย มหัทธนาภิวัฒน์, 110 หน้า.

งานวิจัยนี้มุ่งเน้น การพัฒนากรอบงานเชิงบริการของระบบติดตามการขนส่ง โดยมีวัตถุประสงค์เพื่อประยุกต์ใช้ในการพัฒนาระบบติดตามห่วงโซ่อุปทานสำหรับการขนส่ง ซึ่งเป็นส่วนหนึ่งของระบบโลจิสติกส์ โดยจะกรอบงานจะถูกพัฒนาขึ้นโดยภาษาจาวา โดยการออกแบบให้สามารถรองรับการพัฒนาระบบภายใต้สถาปัตยกรรมเชิงบริการ(Service oriented architecture) หรือ SOA ซึ่งมีข้อดีในหลายๆ ด้าน และเป็นรูปแบบสถาปัตยกรรมที่กำลังได้รับความสนใจอย่างมากในปัจจุบัน

เนื่องจากปัจจุบันการขนส่งสินค้าในประเทศไทยซึ่งเป็นส่วนหนึ่งของกระบวนการทางด้านโลจิสติกส์ ประสบปัญหาในหลายๆด้านเช่น การติดตามและควบคุมการขนส่งเป็นไปได้ด้วยความยากลำบาก กระบวนการขนส่งขาดประสิทธิภาพทำให้เกิดปัญหาความล่าช้า ความผิดพลาดในการขนถ่ายสินค้า

กรอบงานนี้ได้มีการพัฒนาขึ้นโดยการรวบรวมกระบวนการของการขนส่งและเทคโนโลยีระบุตำแหน่งด้วยดาวเทียมหรือ GPS (Global Position System) ซึ่งกำลังเข้ามามีบทบาทในการพัฒนาระบบโลจิสติกส์ในปัจจุบันและเริ่มมีการใช้งานกันอย่างแพร่หลายในปัจจุบัน ซึ่งการที่ได้มีการพัฒนากรอบงานนี้ จะสามารถลดขั้นตอน ข้อผิดพลาดและระยะเวลาในการพัฒนาระบบโลจิสติกส์ ส่งผลให้ต้นทุนในการพัฒนาระบบลดลง รวมทั้งการที่ออกแบบมาให้เป็นกรอบงานเชิงบริการก็ยังมีข้อดีในด้านความสามารถในการทำงานเชื่อมต่อกับระบบอื่นๆ ทำได้ง่ายขึ้นอีกด้วย

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2554 ลายมือชื่ออาจารย์ที่ปรึกษา_____

CHANINTORN RUANGUDOMSAKUL : SERVICE BASED FRAMEWORK FOR SUPPLY CHAIN TRACKING IN TRANSPORTATION. THESIS ADVISOR : ASST. PROF. PICHAYOTAI MAHATTHANAPIWAT, Ph. D., 110 PP.

FRAMEWORK/SUPPLY CHAIN/GPS/SOFTWARE ENGINEERING

This research focus on development of service based framework for transportation tracking system. The objective is for implement in development of supply chain tracking for transportation system which is part of logistic system. The framework develops by java language platform and design to support system development based on service oriented architecture (SOA). Now SOA is of most interesting software architecture and has many benefits.

Now, Transportations in Thailand have many problems. Such as difficult to track and monitoring transportation, lag of performance in transportation process, Error in Transportation process, etc.

The framework develops for integration of transportation process and Global Position System (GPS). This framework will reduce step, cost, time and error in development of logistic system. In addition to this framework design using service based concept, it also has the advantage of connectivity to other systems.

มหาวิทยาลัยเทคโนโลยีสุรนารี

School of Computer Engineering Student's Signature_____

Academic Year 2011 Advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วยดี ผู้วิจัยขอขอบพระคุณ คณะบุคคลดังต่อไปนี้ที่ได้ร่วมเป็นส่วนหนึ่ง ที่ทำให้วิทยานิพนธ์สำเร็จได้ทั้งในด้านวิชาการ ด้านดำเนินงานวิจัย และในด้านอื่นๆ

ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์

รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพ หัวหน้าสาขาวิชา พร้อมด้วยคณาจารประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ทุกท่าน

ขอขอบคุณ คุณธนพร ชื้อชวาลกุล สำหรับเรื่องการดำเนินการด้านการทดสอบระบบ

ขอขอบคุณ คุณมัสติกา พิมพ์สุตะ สำหรับคำปรึกษาและความช่วยเหลือด้านต่างๆ

ขอขอบคุณ บริษัท ศรีสะเกษทีทีมอเตอร์ จำกัด สำหรับความอนุเคราะห์ในการทดสอบระบบ

และสุดท้ายนี้ขอขอบคุณ คุณพ่อไพศาล และคุณแม่สุบิน เรืองอุดมสกุล ที่ให้การเลี้ยงดูอบรมสั่งสอน และส่งเสริมให้ศึกษาเล่าเรียนเป็นอย่างดีมาโดยตลอด จนทำให้ผู้วิจัยสามารถประสบความสำเร็จได้

ชนินทร เรืองอุดมสกุล

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ซ
สารบัญรูป ..... ฌ
บทที่
1 บทนำ 1
1.1 ความสำคัญและที่มาของงานวิจัย 1
1.2 วัตถุประสงค์ของการวิจัย 1
1.3 ข้อตกลงเบื้องต้น 2
1.4 ขอบเขตของงานวิจัย 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
2 **ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง** 4
2.1 Service Oriented Architecture (SOA) 4
2.1.1 ความหมายของSOA 4
2.1.2 ความหมายของบริการ (Service) 5
2.1.3 รูปแบบการทำงานของ SOA 5
2.1.4 องค์ประกอบสำคัญในการพัฒนาเซอร์วิส 6
2.1.5 หลักการของ Service-Oriented 7
2.1.6 หลักการที่เป็นพื้นฐานของ SOA 10
2.1.7 Service Layer และการวางแผนในการพัฒนา SOA 11
2.1.8 วิธีการวางแผนการพัฒนาระบบด้วย SOA 12
2.2 เว็บเซอร์วิส (Web Service) 18

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ (ต่อ)

หน้า

2.2.1 ความเป็นมาของเว็บเซอร์วิส 18

2.2.2 ความหมายของเว็บเซอร์วิส 18

2.2.3 เทคโนโลยีที่ใช้ในการพัฒนาเว็บเซอร์วิส 19

2.3 ระบบบริหารจัดการงานขนส่ง 28

2.3.1 การขนส่งกับโครงสร้างพื้นฐาน 28

2.3.2 เทคโนโลยีที่เกี่ยวกับการขนส่ง 29

2.3.3 ข้อเท็จจริงว่าด้วยระบบจัดการงานขนส่ง 31

2.4 GPS Tracking System 32

2.4.1 ประโยชน์ของ GPS Tracking System 32

2.5 ห่วงโซ่อุปทาน (Supply Chain) 33

2.6 ระบบขนส่งอัจฉริยะ (Intelligent Transport System หรือ ITS) 34

2.6.1 GIS and ITS 34

2.6.2 GIS ในงานการติดตามยานพาหนะ 34

2.7 การพัฒนาเฟรมเวิร์ค (Framework Development) 37

2.7.1 การจำแนกประเภทของเฟรมเวิร์ค 38

2.8 OpenGTS (Open GPS Tracking System) 39

2.9 A Framework for Developing Distributed Location Based Applications 41

2.9.1 สถาปัตยกรรมของเฟรมเวิร์ค 42

3 วิธีดำเนินงานวิจัย 45

3.1 ระเบียบวิธีวิจัย 45

3.2 การกำหนดปัญหา (Problem Definition) 47

3.3 การศึกษาความเป็นไปได้ (Feasibility Study) 47

3.4 การวิเคราะห์ระบบ (System Analysis) 49

3.5 การออกแบบระบบ (System Design) 51

3.5.1 ออกแบบสถาปัตยกรรมของเฟรมเวิร์ค(Framework Architecture Design) 51

3.5.2 Package Design 54

3.6 การพัฒนาระบบ (Implementation) 63

สารบัญ (ต่อ)

หน้า

3.6.1 จัดเตรียมสิ่งแวดล้อมสำหรับการพัฒนาระบบและตั้งค่าเริ่มต้น..... 63
3.6.2 เขียนโปรแกรม (Coding) 66
4 การทดสอบและสรุปผล 69
4.1 สภาพแวดล้อมที่ใช้ในการทดสอบ 69
4.1.1 ฮาร์ดแวร์ (Hardware) 69
4.1.2 ซอฟต์แวร์ (Software) 69
4.1.3 ส่วนประกอบอื่นๆ (Component) 69
4.2 การเตรียมการทดสอบ 70
4.2.1 อิมพลีเมนต์ TableAdapter Class 70
4.2.2 อิมพลีเมนต์ IDBViewMapper Interface 71
4.2.3 อิมพลีเมนต์ Abstract Service Class 72
4.2.4 Create Web Service 72
4.3 ขั้นตอนในการทดสอบ 74
4.3.1 การการทดสอบส่วนอธิบายบริการ โดยการทำ .NET Service Reference 75
4.3.2 การทดสอบความถูกต้องของการทำงานของ Service Operation ด้วยการทำ Unit Testing 82
4.3.3 การนำ SBFSTT Framework มาสร้างระบบตัวอย่าง 87
4.4 อภิปรายผล 97
5 บทสรุป 98
5.1 สรุปผลการวิจัย 99
5.2 การประยุกต์งานวิจัย 100
5.3 แนวคิดในการพัฒนาต่อ 100
รายการอ้างอิง 101
ภาคผนวก
ภาคผนวก ก. บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา 103
ประวัติผู้เขียน 110

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตารางที่ **หน้า**

2.1 อธิบาย Element ของ SOAP Message 22

2.2 อธิบาย fault Element 23

2.3 อธิบาย Fault code Sub Element 24

2.4 อธิบาย Element ของ WSDL 25

3.1 กระบวนการทั่วไปในการขนส่งสินค้า 49

4.1 แสดงเซอร์วิส และรายละเอียดของเซอร์วิสของ SBFSTT Framework 74

4.2 แสดงข้อมูล ItemInfo สำหรับกำหนดให้ตัวแปรใน Unit Test 82

4.3 แสดงข้อมูล CargoInfo สำหรับกำหนดให้ตัวแปรใน Unit Test 82

4.4 แสดงข้อมูล SenderInfo สำหรับกำหนดให้ตัวแปรใน Unit Test 82

4.5 แสดงข้อมูล ReceiverInfo สำหรับกำหนดให้ตัวแปรใน Unit Test 82

4.6 แสดงข้อมูล Cargo Item สำหรับกำหนดให้ตัวแปรใน Unit Test 82

4.7 แสดงข้อมูล ShipmentInfo สำหรับกำหนดให้ตัวแปรใน Unit Test 83

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป

รูปที่		หน้า
2.1	แสดงภาพของ SOA Model	6
2.2	แสดงภาพของ Business Logic และ Application Logic	8
2.3	แสดงภาพของ Service Interface Layer	9
2.4	แสดงภาพของเซอร์วิสที่ห่อหุ้ม Application Logic ไว้ภายใน	9
2.5	แสดงขั้นตอนการพัฒนาระบบด้วย SOA	12
2.6	แสดงขั้นตอนของ Service-Oriented Analysis	13
2.7	แสดงขั้นตอนของ Service-Oriented Design	14
2.8	แสดงชั้น (Layer) ของ Software Technology	16
2.9	แสดงชั้น (Layer) ที่ใช้สำหรับพัฒนา SOA	16
2.10	แผนภาพแสดงการทำงานของ SOAP Protocol	20
2.11	แสดงโครงสร้างของเอกสาร SOAP	21
2.12	Web Service with SOAP, UDDI และ WSDL	27
2.13	แผนที่แสดงเส้นทางภายในรถช่วยแนะนำการเดินทางแก่ผู้ขับขี่	35
2.14	GIS ใช้ร่วมกับ GPS ในการติดตามรถประจำทาง และ รถฉุกเฉิน	36
2.15	การติดตามตำแหน่งรถบรรทุกและแสดงผลบน GIS	37
2.16	OpenGTS Architecture	39
2.17	OpenGTS Web UI	40
2.18	แสดงสถาปัตยกรรมของเฟรมเวิร์ค	42
2.19	แสดง Use case Diagram ของ mobile device	43
3.1	OpenGTS Architecture Overview	51
3.2	แสดงสถาปัตยกรรมของกรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง	52
3.3	แสดงคลาสที่มีอยู่ใน package org.opengts.custom.dbtools	54
3.4	แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db	55
3.5	แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db.table	56

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.6	แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db.views	57
3.7	แสดงคลาสที่มีอยู่ใน package org.opengts.custom.services	57
3.8	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น BusinessDataImportService	58
3.9	แสดง Operation ของ BusinessDataImportService	59
3.10	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น PackageManagementService	59
3.11	แสดง Operation ของ PackageManagementService	60
3.12	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น ShipmentManagementService	60
3.13	แสดง Operation ของ ShipmentManagementService	61
3.14	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น GeoZoneInfoService	61
3.15	แสดง Operation ของ GeoZoneInfoService	62
3.16	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น GPSEventInfoService	62
3.17	แสดง Operation ของ GPSEventInfoService	62
3.18	แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น UtilService	63
3.19	แสดง Operation ของ UtilService	63
3.20	แสดง OpenGTS Project หลังจากตั้งค่าให้ทำงานบน Eclipse IDE	65
3.21	แสดงหน้าจอและผลการทำงานของ OpenGTS	66
3.22	แสดงรูปแบบการอิมพลีเมนต์ Table Class	67
4.1	แสดงขั้นตอนการอิมพลีเมนต์ SBFSTT Framework	70
4.2	แสดงการอิมพลีเมนต์ TableAdapter class	70
4.3	แสดงการอิมพลีเมนต์ table adapter class	71
4.4	แสดงการสร้างเว็บเซอร์วิสซึ่งมีการประกาศ Service Endpoint Interface เพื่อเป็น API สำหรับเข้าถึง Method ในเซอร์วิส	75
4.5	แสดงการกำหนดมาตรฐานให้ใช้ Soap Binding เป็น SOAP 1.2	76
4.6	แสดงหน้าจอของการทำ Service Reference ด้วย visual studio 2008 Professional	80

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.7	แสดง Object Browser ของ Visual Studio 2008 ซึ่งแสดงให้เห็น Service method ที่ได้ทำการ Reference จากเซอร์วิส	80
4.8	แสดง C# Class library Project ซึ่งทำ Service Reference เรียบร้อยแล้ว	83
4.9	แสดงตัวอย่างหน้าจอแสดงผลการทดสอบเว็บเซอร์วิสด้วย Visual Studio 2008 Unit Testing	86
4.10	แสดงสถาปัตยกรรมของระบบ Cargo Transport Management Web Application ซึ่งเป็นระบบตัวอย่างที่สร้างจากการอิมพลีเมนต์ SBFSTT Framework	87
4.11	แสดงหน้าแรกของตัวอย่างโปรแกรมซึ่งพัฒนาขึ้นจากเว็บเซอร์วิสที่สร้างขึ้นจาก SBFSTT Framework	88
4.12	แสดงหน้าจอกรอกข้อมูลซึ่งเรียกใช้เว็บเซอร์วิสจาก SBFSTT Framework	88
4.13	แสดงการกำหนดจุดสำคัญ (Geo Zone) ด้วยระบบ OpenGTS	89
4.14	แสดงตัวอย่างหน้าจอแสดงผลในส่วนของจุดสำคัญ (Geo Zone) อาศัยการดึงข้อมูลจากเซอร์วิสของ SBFSTT Framework ซึ่งสามารถนำไปใช้กำหนดจุดผ่านของการขนส่งสินค้าได้	89
4.15	แสดงตัวอย่างหน้าจอแสดงผลของ Device ซึ่งข้อมูลที่ได้เกิดจากการดึงข้อมูลโดยผ่านเว็บเซอร์วิสของ SBFSTT Framework	90
4.16	แสดงหน้าจอโปรแกรมซึ่งติดตั้งบน Smart Phone ที่จะคอยรายงานตำแหน่งพิกัด GPS มายังเซิฟเวอร์	90
4.17	แสดงหน้าจอ Track map ในส่วนของ GPS Tracking Service Manager and Monitor ซึ่งผู้วิจัยเลือกดัดแปลงระบบของ OpentGTS มาใช้โดยไม่ต้องทำการพัฒนาขึ้นใหม่	91
4.18	แสดงหน้าจอโปรแกรมแสดงผลในส่วนของการดูประวัติเส้นทางการขนส่งสินค้า	92
4.19	แสดง hook method ซึ่งสามารถทำการ override ได้	93

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

โลจีสติกส์ มีต้นกำเนิดจากแผนการจัดส่งอาวุธและอาหารในการส่งกำลังของกองทัพทหารสหรัฐในสงครามโลก ในปัจจุบันได้แปรเปลี่ยนมาสู่ภาคธุรกิจ โดยโลจีสติกส์เป็นส่วนหนึ่งของกระบวนการของการบริหารงานห่วงโซ่อุปทาน ซึ่งรวมถึงเรื่องของการวางแผน การดำเนินการ การควบคุมการไหลเวียน การจัดเก็บวัสดุสินค้า การบริการ และสารสนเทศ อย่างมีประสิทธิภาพและประสิทธิผล จากแหล่งจุดกำเนิดของวัตถุดิบ จนถึงจุดบริโภคหรือจุดการใช้งาน เพื่อตอบสความต้องการของลูกค้า โดยเป้าหมายของระบบโลจิสติกส์คือ การบริหารจัดการข้อมูล คน เครื่องจักร สินค้า เพื่อการจัดส่งสินค้าไปยังสถานที่ที่ถูกต้องในเวลาที่เหมาะสมและมีค่าใช้จ่ายประหยัดที่สุด

เนื่องจากปัจจุบันการขนส่งสินค้าในประเทศไทยซึ่งเป็นส่วนหนึ่งของกระบวนกทางด้านโลจีสติกส์ ประสบปัญหาในหลายๆด้านเช่น การติดตามและควบคุมการขนส่งเป็นไปได้ด้วยความยากลำบาก กระบวนการขนส่งขาดประสิทธิภาพทำให้เกิดปัญหาความล่าช้า ความผิดพลาดในการขนถ่ายสินค้า

ซึ่งปัญหาเหล่านี้ปัจจุบันได้มีการประยุกต์นำเทคโนโลยีต่างๆ มาใช้ในการปรับปกระบวนการทั้งในส่วนของเทคโนโลยีการระบุตำแหน่งด้วยดาวเทียมหรือ GPS (Global Position System), บาร์โค๊ดมาตรฐานต่างๆ, แถบแม่เหล็ก รวมทั้งเทคโนโลยีการระบุด้วยคลื่นวิทยุหรือ RFID (Radio Frequency Identification) แต่เนื่องด้วยการพัฒนาระบบเหล่านี้มีต้นทุนในการพัฒนาที่สูงส่งผลให้ภาคธุรกิจในประเทศไทยซึ่งมีต้นทุนไม่สูงไม่สามารถนำไปประยุกต์ใช้ได้ รวมทั้งการพัฒนาอาจต้องใช้ระยะเวลาที่นานซึ่งจะส่งผลให้เกิดความเสียเปรียบในการแข่งขันทางธุรกิจ จึงเป็นที่มาของโครงการวิจัยนี้

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษากระบวนการในการออกแบบเฟรมเวิร์ค
2. เพื่อออกแบบและพัฒนาเฟรมเวิร์คของระบบการติดตามการขนส่งสินค้าโดยประยุกต์ใช้เทคโนโลยีการตรวจหาพิกัดโดยการใช้ดาวเทียม

1.3 ข้อตกลงเบื้องต้น

1. กรอบงาน (Framework) ที่พัฒนาขึ้นจะพัฒนาด้วยเทคโนโลยีภาษา JAVA

2. สิ่งแวดล้อมที่จะใช้ในการพัฒนาได้แก่ ระบบปฏิบัติการ Windows 7 professional 64 bit เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาคือ Lenovo Y560, CPU Intel Core i7 740 1.73 Ghz. หน่วยความจำ 4 Gb.

3. เครื่องมือที่ใช้ในการพัฒนาได้แก่ Eclipse Helios, OpenGTS (Open GPS Tracking System) V.2.3.7, Apache Tomcat 6, Apache CXF 2.x, ภาษที่ใช้พัฒนา คือภาษา JAVA

1.4 ขอบเขตของงานวิจัย

งานวิจัยนี้มุ่งเน้นการออกแบบพัฒนาเฟรมเวิร์คด้วยภาษา JAVA เพื่อเป็นตัวแบบของการพัฒนาระบบการติดตามการขนส่งสินค้าโดยประยุกต์ใช้เทคโนโลยีการตรวจหาพิกัดโดยการใช้ดาวเทียมหรือ GPS (Global Position System) ร่วมกับเทคโนโลยีอื่นๆ ซึ่งจะส่งผลให้การพัฒนาระบบโลจีสติกส์ในส่วนของการติดตามและควบคุมการขนส่งเป็นไปได้อย่างสะดวกรวดเร็วมากขึ้น โดยจะมีการพัฒนาองค์ประกอบต่างๆ ดังนี้

1. เฟรมเวิร์คสำหรับการพัฒนาระบบติดตามการขนส่งสินค้าโดยประยุกต์ใช้เทคโนโลยีการระบุตำแหน่งด้วยดาวเทียม ซึ่งจะถูกพัฒนาขึ้นด้วยภาษาจาวา โดยมีเซอร์วิสดังนี้

1.1 Business data import service มีหน้าที่จัดการในส่วนการนำเข้าข้อมูลการจัดส่งสินค้าหรือสิ่งของ ทั้งในส่วนการนำเข้าข้อมูลโดยตรงหรือนำเข้าผ่านการเชื่อมต่อกับระบบอื่น

1.2 Packaging management service มีหน้าที่จัดการในส่วนการจัดระเบียบของหีบห่อในการจัดส่งทั้งในส่วนของการจัดกลุ่มและการระบุตัวสิ่งของเช่นการกำหนดรหัสการจัดส่งด้วยบาร์โค๊ดหรือ RFID

1.3 GPS - vehicle association service มีหน้าที่จัดการในส่วนของการติดตั้งระบบระบุตำแหน่งของพาหนะขนส่งด้วยอุปกรณ์ตรวจหาพิกัดโดยการใช้ดาวเทียม

1.4 Shipment management service มีหน้าที่จัดการในส่วนของการนำสินค้าหรือสิ่งของขึ้นสู่พาหนะจัดส่ง

1.5 GeoZone defining มีหน้าที่ในการกำหนดตำแหน่งพิกัดและข้อมูลเบื้องต้นของสถานีขนถ่ายสินค้า

1.6 GPS position capture มีหน้าที่รับตำแหน่งพิกัดดาวเทียมที่ถูกส่งมาจากอุปกรณ์ GPS ที่ถูกติดตั้งในพาหนะขนส่ง

1.7 Transport position monitoring moduleมีหน้าที่แสดงตำแหน่งและข้อมูลสถานะของการขนส่งปัจจุบันของสินค้า

1.8 Information query module มีหน้าที่แสดงข้อมูลประวัติการขนส่งหรือข้อมูลอื่นๆ เพื่อการติดตามและตรวจสอบย้อนหลังในการจัดส่งสินค้าหรือสิ่งของ

2. ต้นแบบของระบบที่พัฒนาขึ้นจากเฟรมเวิร์คเพื่อเป็นแบบอย่างแนวคิดในการพัฒนาระบบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เฟรมเวิร์คในการพัฒนาระบบการติดตามการขนส่งสินค้าที่มีประสิทธิภาพ ทันสมัย และครอบคลุม ซึ่งจะช่วยลดเวลา ต้นทุนและความซับซ้อนในการพัฒนาระบบโลจีสติกส์ในส่วนของการติดตามการขนส่งสินค้า ซึ่งสามารถสรุปได้ดังนี้

1. ลดขั้นตอนและระยะเวลาในการพัฒนาระบบโลจีสติกส์
2. ต้นทุนในการพัฒนาระบบการติดตามการขนส่งสินค้าลดลง
3. ลดความซับซ้อนในการพัฒนาระบบ
4. สามารถเชื่อมต่อกับระบบอื่นเพื่อนำข้อมูลไปใช้ประโยชน์ต่อได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 Service Oriented Architecture (SOA)

(สุธี พงศาสสกุลชัย, 2550)

แนวคิดเชิงบริการ (Service Orientation) คือแบบแผนในการออกแบบและพัฒนาซอฟต์แวร์ที่อยู่บนพื้นฐานของบริการ ซึ่งเป็นแนวคิดที่ตอบสนองความต้อง รทางธุรกิจในอนาคต โดยการทำงานอยู่บนมาตรฐานเปิดที่ได้รับการยอมรับ แนวคิดดังกล่าวเ ประยุกต์ใช้เพื่อสนับสนุนการออกแบบคุณลักษณะต่างๆ ของบริการ เพื่อให้เ สถาปัตยกรรมเชิงบริการ (Service Oriented Architecture : SOA) ซึ่งแนวคิดเชิงบริการจะมององค์ประกอบต่างๆ ของซอฟต์แวร์เป็นบริการ (Service) ซึ่งจะมีส่วนที่คล้ายคลึงกับแนวคิดเชิงวัตถุ (Object Orientation) ที่มององค์ประกอบต่างๆ เป็นวัตถุ (Object)

SOA วิวัฒนาการมาจากเทคโนโลยีการประมวลผลแบบกระจาย (Distributed Computing) ซึ่ง SOA มีจุดเด่นคือ การไม่ยึดติดกับภาษาที่ใช้ในการพัฒนาซอฟต์แวร์ มีความยืดหยุ่นสู (flexibility) สามารถทำงานในระบบที่แตกต่างกันได้ (Platform Independent) แต่เนื่องจาก SOA เป็นเพียงหลักการและวิธีการวิเคราะห์และออกแบบซอฟต์แวร์เพื่อนำ อิมพลีเมนต์ ร่วมกับเทคโนโลยีต่างๆ เช่น DCOM, CORBA หรือ RPC เป็นต้น ซึ่งการนำเทคโนโลยีดังกล่าวมาอิมพลีเมนต์ต้องเสียค่าใช้จ่ายที่ค่อนข้างสูง และสามารถรองรับได้เพียงแค่บางแพลตฟอร์มจึงไม่เหมาะที่จะนำมาอิมพลีเมนต์ ส่งผลให้ปัจจุบันนิยมนำเอาเทคโนโลยีเว็บเซอร์วิส (Web Service) มาใช้ในการอิมพลีเมนต์ SOA เนื่องจากเป็นเทคโนโลยีที่มีประสิทธิภาพ มีความยืดหยุ่นสูงและไม่ขึ้นกับแพลตฟอร์มและตอบสนองความต้องการด้านธุรกิจได้เป็นอย่างดี

2.1.1 ความหมายของ SOA

สถาบัน OASIS (Organization for the Advancement of Structured Information Standards) ได้ให้ความหมายของ SOA ไว้ดังนี้

“สถาปัตยกรรมเชิงบริการ (Service Oriented Architecture: SOA)” คือแบบจำลอง การจัดระเบียบและการใช้ประโยชน์จากบริการที่ถูกแจกจ่ายโดยผู้เป็นเจ้าของบริการที่หลากหลาย ซึ่งบริการที่ถูกแจกจ่ายนั้นจะอยู่ในรูปแบบที่พร้อมในการนำเสนอสู่ภายนอก เพื่อให้ผู้ที่ต้องการใช้สามารถค้นหา ทำงานร่วม และเรียกใช้บริการเหล่านั้นเพื่อนำไปสู่ผลลัพธ์ที่ต้องการได้

Tomas Erl ซึ่งเป็นนักวิชาการทางด้านคอมพิวเตอร์แบะระบบบริหารจัดการ ให้ให้คำจำกัดความของ SOA ไว้ดังนี้

“สถาปัตยกรรมเชิงบริการ (Service Oriented Architecture : SOA)” คือแบบจำลอง การแบ่งส่วนการทำงานซึ่งสามารถทำได้โดยอัตโนมัติออกเป็นหน่วยย่อยของแต่ละลอจิก ซึ่งประกอบไปด้วย ลอจิกการทำงานทางธุรกิจแบบอัตโนมัติ ที่สามารถที่จะถูกกระจายไปใช้งานข้ามระบบได้” (Thomas Erl, 2006)

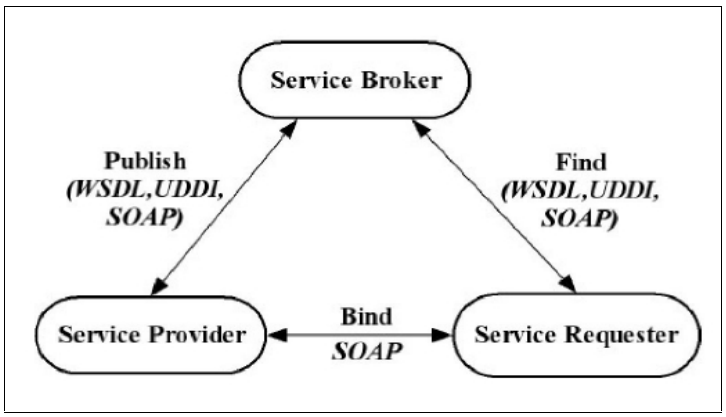
2.1.2 ความหมายของบริการ (Service)

บริการหรือเซอร์วิส (Service) คือฟังก์ชั่นหรือการทำงานที่ถูกสร้างขึ้น ในลักษณะของการให้บริการ ซึ่งอยู่ในรูปแบบของ Interface ที่มีการกำหนดข้อตกลงในการเรียกใช้บริการไว้ โดยมีการห่อหุ้ม (Encapsulate) ลอจิก (logic) ที่อยู่ภายในไว้ มีการกำหนดขอบข่ายการทำงานไว้อย่างชัดเจน โดยที่เซอร์วิสจะทำงานตามขอบข่ายการทำงานดังกล่าวได้ด้วยตัวของมันเอง เนื่องจาฟังก์ชั่นการทำงานจะถูกบรรจุไว้ภายใน

2.1.3 รูปแบบการทำงานของ SOA

SOA ถูกแบ่งเป็นประเภทต่างๆ ตามเทคโนโลยีที่อิมพลีเมนต์เพื่อสร้างเซอร์วิสโ แบบจำลองของ SOA (SOA Model) เป็นจุดเริ่มต้นของมาตรฐานที่สำคัญของ เว็บเซอร์วิส จุดเริ่มต้นของ SOA เกิดจาก 3 ส่วนประกอบสำคัญคือ ผู้ให้บริการ (Service Provider) , ผู้ร้องขอบริการ (Service Requester) , และหน่วยสืบค้นบริการ (Service Registry) โดยมีรายละเอียดดังนี้

- ผู้ให้บริการ (Service Provider) คือ โปรแกรมประยุกต์ที่จัดเตรียมบริการไว้ให้โปรแกรมประยุกต์อื่นเรียกใช้
- ผู้ร้องขอบริการ (Service Requester) คือ โปรแกรมประยุกต์ที่ขอบริการจากโปรแกรมประยุกต์ที่ประกาศ (Publish)
- หน่วยสืบค้นบริการ (Service Registry/Service Broker) คือฐานข้อมูลกลางที่รวบรวมบริการต่างๆ ที่ผู้ให้บริการประกาศไว้



รูปที่ 2.1 แสดงภาพของ SOA Model

SOA มีรูปแบบการทำงานคือ ผู้ให้บริการ (Service Provider) จะประกาศ (Publish) บริการ (Service) ของตนเองไปยังหน่วยสืบค้นบริการ (Service Registry/Service broker) เพื่อให้ผู้ขอใช้บริการ (Service Requester) ค้นหาบริการผ่าน Service Registry เมื่อพบบริการที่ตนเองต้องการแล้วจึงเรียกใช้ (Bind/Invoke) บริการดังกล่าวผ่าน Service Provider ต่อไป

2.1.4 องค์ประกอบสำคัญในการพัฒนาเซอร์วิส

SOA สามารถนำไปประยุกต์ได้กับหลายเทคโนโลยี แต่เว็บเซอร์วิสเป็นเทคโนโลยีสามารถนำเอาหลักการของเทคโนโลยีเชิงบริการ (Service – Oriented) มาใช้ได้เป็นรูปธรรมมากที่สุด ดังนั้นองค์ประกอบในการพัฒนาเซอร์วิสที่จะกล่าวถึงต่อไปนี้ จะใช้เว็บเซอร์วิส เป็นตัวแทนในการอธิบายโครงร่างการทำงานขององค์ประกอบต่างๆ ของเซอร์วิสได้แก่ Service Description (WSDL) และ Message (SOAP)

- Web Service Framework

โครงร่างการทำงาน หรือกรอบการทำงาน (Framework) คือการรวบรวมความคิดต่างๆ ที่ประกอบด้วย กลุ่มของสถาปัตยกรรม (Architecture) เทคโนโลยี(Technology) แนวคิด (Concept) และแบบจำลอง (Model) ต่างๆ เพื่อใช้เป็นข้อกำหนดในการพัฒนาสิ่งใดสิ่งหนึ่ง โดยเฟรมเวิร์คของเว็บเซอร์วิสจะมีลักษณะดังนี้

- ส่วนประกอบหลักของเว็บเซอร์วิส ซึ่งจะประกอบด้วย (1) Web Service (2) Service Description และ (3) Message

- กำหนดมาตรฐานโดย Standard Organization โดยไม่ขึ้นอยู่กับผู้ผลิตซอฟต์แวร์ใดๆ แต่สามารถนำไปอิมพลีเมนต์บนแพลตฟอร์มของผู้ผลิตซอฟต์แวร์ต่างๆ แล้วนำมาทำงานร่วมกันได้

- ข้อตกลงในการติดต่อสื่อสารถูกกำหนดโดย Service Description ที่อยู่ในรูปแบบของ WSDL

- Messaging Framework ที่ใช้ในการติดต่อสื่อสารอยู่ในรูปแบบของ SOAP

- ใช้สถาปัตยกรรมของ UDDI การลงทะเบียนและค้นหา Service Description

เฟรมเวิร์คของเว็บเซอร์วิส เป็นแนวคิดที่ถูกหลอมรวม หรืออยู่แนวทางเดียว กับหลักการขแนวคิดเชิงบริการ (Service-Orientation)

- แนวคิดการพัฒนาเซอร์วิส

Service อาจมีบทบาทที่แตกต่างกันขึ้นอยู่กับลักษณะการแลกเปลี่ยน Message และสภาพแวดล้อมที่ถูกเรียกใช้ โดยอาจเป็นทั้งผู้ให้บริการ (Service Provider) ผู้ร้องขอบริการ (Service Requester) หรือสื่อกลางบริการ (Intermediary) นอกจากนี้เซอร์วิสยังสามารถแบ่งได้หลายประเภทขึ้นอยู่กับวิธีการนำไปใช้ และลักษณะของ Application Logic ที่มีอยู่ภายใน แต่มี 3 ชนิดที่สำคัญคือ

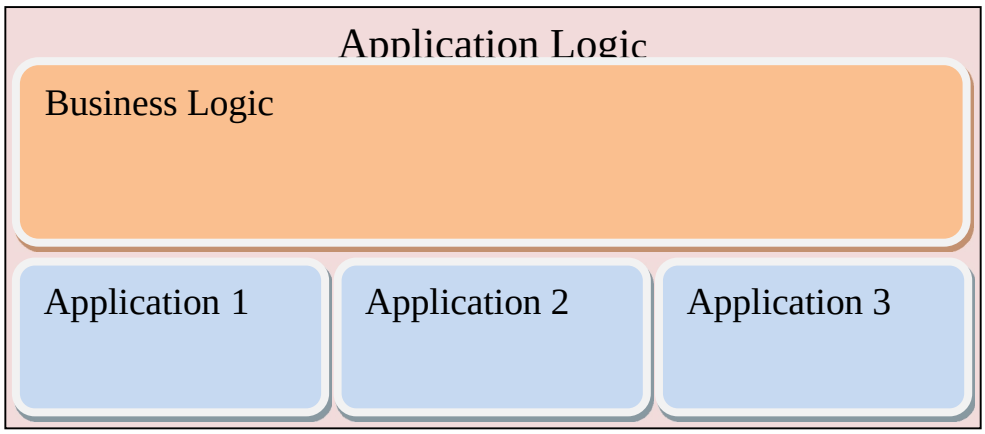
1. Business Service Model
2. Unity Service Model
3. Controller Service Model

การติดต่อสื่อสารระหว่างเซอร์วิสจะอาศัยเอกส Service Description เพื่อบอกรายละเอียดและวิธีเรียกใช้เซอร์วิส (Web Service จะใช้ WSDL ในการสร้าง Service Description) หากเซอร์วิสใดต้องการเรียกใช้บริการจากเซอร์วิสอื่น จะต้องค้นหา Service Description ผ่านหน่วยสืบค้นบริการ(Service Registry) เพื่อให้ผู้ร้องขอบริการเข้ามาค้นหา และทุกการติดต่อสื่อสารระหว่างเซอร์วิสจะอยู่ในแบบของ Message และใช้โปรโตคอลที่ติดต่อสื่อสารข้ามแพลตฟอร์มกันได้ (Web Service ใช้ SOAP เป็นโปรโตคอลขนส่ง)

2.1.5 หลักการของ Service-Oriented

ลอจิกต่างๆ ที่ถูกรวบรวมไว้ภายในระบบขององค์กร หรือ Enterprise Logic จะต้องเปลี่ยนรูปแบบการทำงานไปตามความต้องการที่เกิดขึ้นจากภายใน และภายนอกองค์กร โดย Enterprise Logic แบ่งได้เป็น 2 ประเภทคือ

- Business Logic
- Application Logic

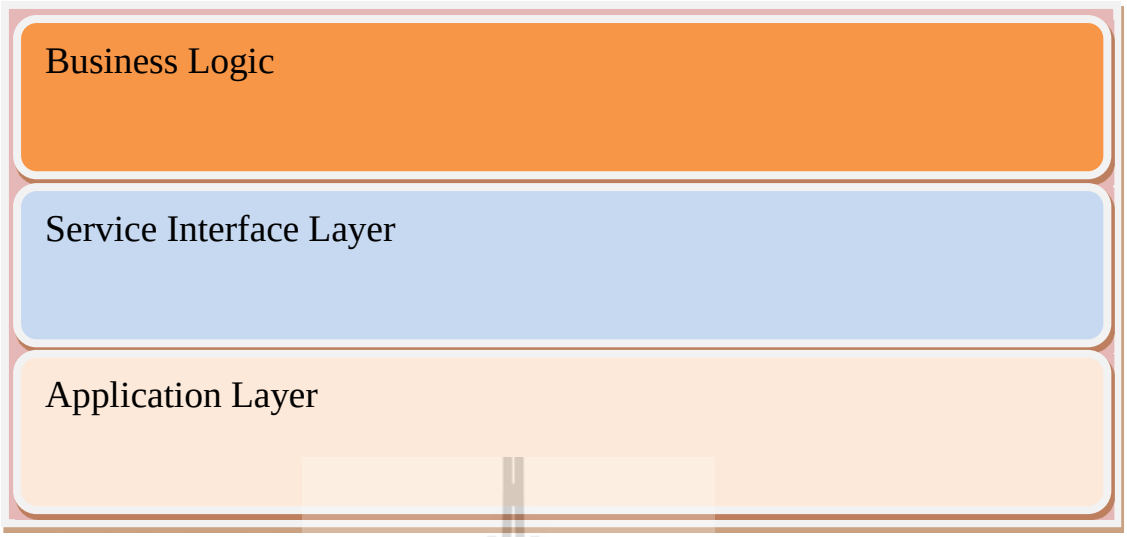


รูปที่ 2.2 แสดงภาพของ Business Logic และ Application Logic

Business Logic คือ ส่วนที่แสดงรายละเอียดการอิมพลีเมนต์ที่กำหนด ตามความต้องการขององค์กร โดยแสดงโครงสร้างทั่วไปของกระบวนการต่างๆ ที่ทำให้เห็นความต้ (Requirement) ชัดเจนยิ่งขึ้น โดยต้องคำนึงถึงข้อจำกัดต่างๆ และอิทธิพลจากภายนอกองค์กรด้วย

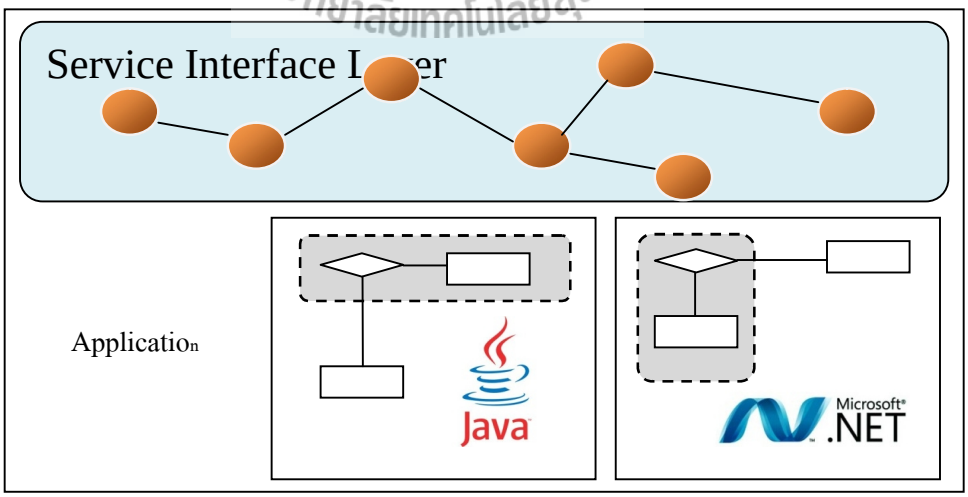
Application Logic คือ การนำ Business Logic ต่างๆ ไปอิมพลีเมนต์บนเทคโนโลยีที่ต่างกัน โดยแสดงให้เห็นขั้นตอนการทำงานด้วยวิธีการพัฒนาระบบที่เหมาะสม ภายใต้ขอบเ ของโครงสร้างพื้นฐานด้าน IT ความสามารถทางด้านเทคนิค ผู้ผลิตซอฟต์แวร์ และข้อจำกัดทางด้านความปลอดภัย

การนำ Service-Oriented มาประยุกต์ใช้งานกับ Enterprise Logic เรียกว่าเซอร์วิสโดย เซอร์วิสจะถูกเก็บไว้ภายในระบบขององค์กร โดยถูกวางไว้ระหว่างชั้น (Layer) ของ Application Logic และ Business Logic ทำให้เซอร์วิสสามารถห่อหุ้มทั้ง Application Logic และ Business Logic



รูปที่ 2.3 แสดงภาพของ Service Interface Layer

จากรูปจะเห็นว่าใน Application Layer จะถูกแยกตามแพลตฟอร์มที่นำมาใช้พัฒนาโปรแกรมประยุกต์ โดยเซอร์วิสต่างๆ จะถูกรวบรวมไว้ภายในระบบขององค์กรในรูปแบบของหน่วยย่อยที่ทำงานได้ด้วยตนเอง (Standalone) และเซอร์วิสต่างๆ ที่ทำงานร่วมกัน อาจความสัมพันธ์กันเป็นลำดับชั้นได้ โดย Parent Service จะรวบรวม Child Service ไว้ภายใน เพื่อนำ Child Service เหล่านั้นมาทำงานร่วมกัน โดยใช้โปรโตคอลสำหรับการติดต่อสื่อสารที่เป็นมาตรฐานเปิด และเซอร์วิสจะห่อหุ้ม Application Layer ไว้ภายในดังรูป 2.3



รูปที่ 2.4 แสดงภาพของเซอร์วิสที่ห่อหุ้ม Application Logic ไว้ภายใน

รูป 2.4 แสดงเซอร์วิสที่อยู่ภายใน Service Interface Layer ซึ่งทำหน้าที่เป็น Service Interface (หรือส่วนติดต่อของ Service) กับ Application Logic ที่อยู่บนแพลตฟอร์มต่างกัน

2.1.6 หลักการที่เป็นพื้นฐานของ SOA

การที่จะนำเอา SOA มาใช้ไม่ว่าจะอยู่ในส่วนใด หรือส่วนของ Enterprise Logic ก็ตามที่มีความจำเป็นต้องรู้ว่าหลักการของ SOA เป็นอย่างไร เพราะหากไม่ทราบถึงหลักการดังกล่าวก็ไม่สามารถสร้างระบบที่ใช้ SOA ให้ทำงานได้แบบ 100% เพราะไม่สามารถทำความเข้าใจและนำไปต่อยอดได้ โดยหลักการของ SOA มีดังนี้

- Service Reusable คือ เซอร์วิสถูกออกแบบมาให้สามารถนำกลับมาใช้ใหม่ได้
- Service Contract คือ เซอร์วิสที่ต้องทำงานร่วมกัน จะต้องกำหนดข้อตกลงของการติดต่อสื่อสารของแต่ละเซอร์วิสไว้ โดยเรียกว่า “Information Exchange”
- Service Loosely Coupled คือ เซอร์วิสต้องถูกออกแบบมาให้มีความเป็นอิสระต่อกัน แต่สามารถทำงานร่วมกันได้โดยไม่ขึ้นกับเซอร์วิสอื่น
- Service Abstraction คือ เซอร์วิสจะต้องซ่อนส่วนการทำงานหรือลอจิกต่างๆ ไว้ โดยสามารถเรียกใช้งานลอจิกผ่านทาง Service Contract
- Service Compostable คือ สามารถนำเซอร์วิสต่างๆ มาประกอบกันเป็นเซอร์วิสใหม่ได้ โดยหลักการนี้ช่วยสนับสนุนหลักการของ Service Reusable อีกด้วย
- Service Autonomous คือ เซอร์วิสจะต้องควบคุมลอจิกต่างๆ ที่อยู่ภายในขอบเขตของตัวเองได้โดยไม่ต้องอาศัยการทำงานจากเซอร์วิสอื่น
- Service Statelessness คือ เซอร์วิสควรเก็บข้อมูลของสถานะหรือกิจกรรมที่ได้ดำเนินการไปแล้วให้น้อยที่สุด
- Service Discoverable คือ เซอร์วิสต้องจัดเตรียมส่วนอธิบายที่เรียกว่า Service Description ไว้เพื่อให้ Service Requester ค้นหา เพื่อเรียกใช้บริการ

สรุปได้คือ การนำหลักการของ SOA มาประยุกต์ใช้กับระบบภายในองค์กรจำเป็นต้องเข้าใจรูปแบบลอจิกต่างๆ ที่ถูกรวบรวมไว้ภายในองค์กร หรือเรียกว่า Enterprise Logic ซึ่งมี 2 ประเภทคือ Business Logic และ Application Logic โดย Business Logic จะแสดงรายละเอียดของกระบวนการทำงานต่างๆ ส่วน Application Logic จะนำ Business Logic มาประยุกต์ใช้งานทำให้เกิดแนวคิดใหม่ที่เรียกว่าเซอร์วิสซึ่งจะถูกวางไว้ระหว่างชั้น ของ Business Logic กับ Application Logic เรียกว่า Service Interface Layer

2.1.7 Service Layer และการวางแผนในการพัฒนา SOA

(Service-Oriented Architecture Concept Technology and Design : Thomas Erl)

ความสำคัญของ Service Layer

Service Interface Layer จะอยู่ระหว่าง Business Layer และ Application Layer เพื่อทำหน้าที่เป็นตัวเชื่อมต่อ Layer ทั้งสอง โดย Service Interface Layer เป็นส่วนที่ใช้กำหนดคุณลักษณะต่างๆ ของ Contemporary SOA ให้กับระบบ แต่การนำเอาคุณลักษณะต่างๆ มาอิมพลีเมนต์เพื่อใช้ประโยชน์จะต้องกระทำผ่าน Service Layer ซึ่งเป็น Layer ย่อยที่อยู่ภายใน Service Interface Layer

Service Layer ทำให้เกิดลักษณะสำคัญต่างๆ ดังนี้

- Service Layer ทำให้เกิดความเป็นอิสระต่อกัน

Service Interface Layer สามารถแบ่งออกเป็น Layer ย่อย ๆ ของเซอร์วิสได้อีก โดยเรียกว่า Service Layer เพื่อให้ระบบมีความเป็นอิสระต่อกันมากขึ้น โดยกลุ่มของเซอร์วิสที่อยู่ในแต่ละ Service layer จะทำหน้าที่รวบรวมลอจิกต่างๆ ไว้ภายใน ทำให้แต่ละส่วนของระบบไม่ขึ้นต่อส่วนอื่นๆ ลอจิกที่อยู่ ใน Business Layer จะทำให้เกิดความเป็นอิสระ และไม่ขึ้นกับเทคโนโลยีใดๆ โดยส่วนที่ระบุถึงเทคโนโลยีจะกำหนดไว้ใน Application Logic แทน ทำให้เกิดความสัมพันธ์ที่เป็นอิสระต่อกันระหว่าง Business Logic และ Application Logic

- Service Layer สามารถใช้เซอร์วิสแทน Application Logic ของระบบเดิมได้

การนำ Application Logic ของระบบเดิมมาใช้งาน จำเป็นต้องนำมาใช้ผ่านเซอร์วิสรวมทั้งการพัฒนาลอจิกใหม่ ๆ ขึ้นมาใช้งาน ก็จำเป็นต้องสนับสนุนการทำงานของเซอร์วิสเช่นกัน โดยสามารถกำหนดข้อบังคับ ข้อจำกัด และความต้องการของระบบเดิมได้ เพื่อนำไปใช้พิจารณา ในขั้นตอนการออกแบบเซอร์วิสต่อไป ในการ Service Layer เพื่อควบคุมแอปพลิเคชั่นต่างๆ ของระบบเดิมนั้นจำเป็นต้องนำหลักการบางอย่างของ SOA มาใช้งาน กลุ่มของเซอร์วิสเหล่านี้จะถูกออกแบบมาเพื่อให้ทำหน้าที่แทน Application Logic และเก็บไว้ใน Service Layer ที่จัดเตรียมไว้เรียกว่า “Application Service Layer”

- Service Layer สามารถใช้เซอร์วิสแทน Business Logic ได้

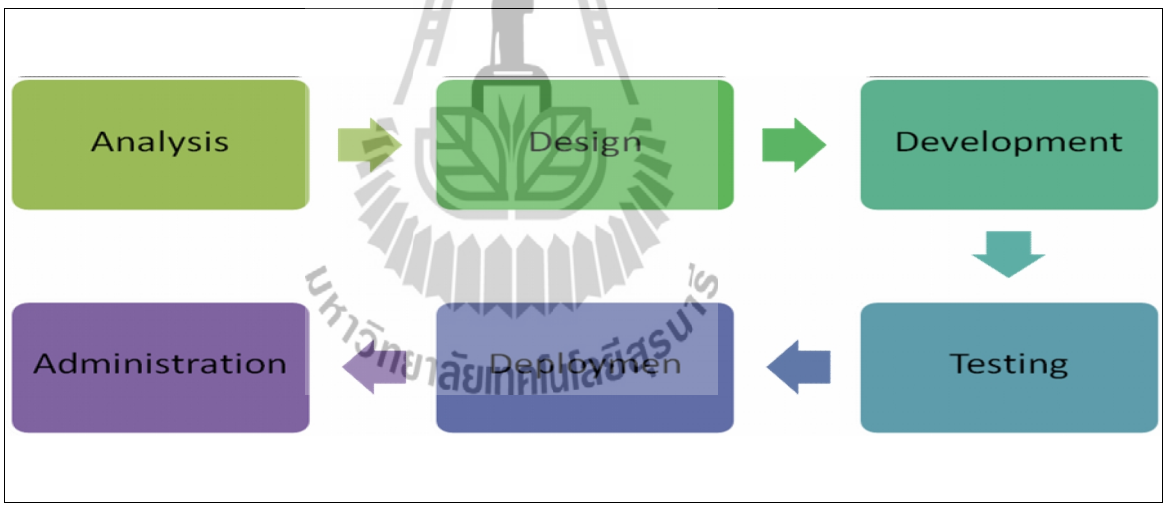
การใช้เซอร์วิสเพื่อบรรจุ Business Logic ของกระบวนการทางธุรกิจ (Business Process) หรือแบบจำลองธุรกิจ (Business Model) ที่ออกแบบไว้ ต้องแน่ใจว่าการออกแบบเซอร์วิสนั้น จะสามารถทำหน้าที่แทนลอจิกเหล่านั้นได้อย่างถูกต้องตาม Business Model ที่กำหนดไว้ โดยเซอร์วิสที่ออกแบบตามลักษณะดังกล่าวจะถูกจัดเก็บไว้ใน Service Layer ที่เรียกว่า “Business Service Layer”

- Service Layer ทำให้การพัฒนาเซอร์วิสมีความคล่องตัว (Agility)

ปัจจัยสำคัญในการสร้างความคล่องตัว (Agility) ให้กับระบบของ SOA คือ ลดการพึ่งพากันระหว่างเซอร์วิสให้เหลือน้อยที่สุด โดยวิธีการเก็บลอจิกที่ใช้ดำเนินงานต่างๆ ไว้ภายในตัวเอง โดยเซอร์วิสจะบรรจุ Business Role ที่ใช้เป็นข้อบังคับสำหรับดำเนินการต่างๆ ในขณะ Runtime ข้อบังคับเหล่านี้ใช้เพื่อจำกัดการใช้งานเซอร์วิสจากภายนอกระบบ โดยเซอร์วิสจะทำหน้าที่เหมือนเป็น Controller Service ซึ่งภายในประกอบด้วยเซอร์วิสย่อย โดย Controller Service สามารถพัฒนาเซอร์วิสที่อยู่ภายในได้อย่างอิสระ แต่ต้องรักษาโครงสร้างของเซอร์วิสที่นำมาประกอบกันไว้ ด้วยลักษณะดังกล่าวจึงทำให้เซอร์วิสมีความคล่องตัวยิ่งขึ้นนั่นเอง

2.1.8 วิธีการวางแผนการพัฒนาระบบด้วย SOA

การพัฒนาระบบด้วย SOA จะมีลักษณะคล้ายกับการพัฒนา Distribute Application กล่าวคือ เซอร์วิสจะถูกออกแบบ (Design), พัฒนา(Develop) และนำไปใช้ (Deploy) ให้เหมาะสมกับ Component มาตรฐานและเทคโนโลยีที่มีอยู่



รูปที่ 2.5 แสดงขั้นตอนการพัฒนาระบบด้วย SOA

ขั้นตอนการวิเคราะห์(Analysis)

ก่อนการพัฒนาระบบด้วย SOA จะต้องมีการวิเคราะห์ระบบก่อน โดยมีวัตถุประสงค์หลัก คือ ศึกษาความเป็นไปได้ และขอบเขตของเซอร์วิสที่ต้องการพัฒนา รวมทั้งศึกษาถึงลอจิกหรือ ดำเนินงานต่างๆ ที่บรรจุไว้ในแต่ละเซอร์วิสโดยใช้กระบวนการในการวิเคราะห์ความต้องการและ ปัญหาที่เกิดขึ้นกับองค์กรอย่างเป็นระบบ หรืออาจเรียกขั้นตอนนี้ว่า “Service-Oriented Analysis”

Service-Oriented Analysis เป็นการวิเคราะห์ความต้องการในการสร้างระบบโดยต้อง วิเคราะห์ว่าระบบต้องการใช้เซอร์วิสอะไร และภายในเซอร์วิสจะบรรจุลอจิกหรือการทำงานอะไร ไว้บ้าง ซึ่งถือว่าเป็นการกำหนดรูปแบบของ Service Layer และกำหนดรูปแบบของโครงสร้าง ระบบทั้งหมดโดยผลจากการวิเคราะห์ทำให้ทราบถึงรายละเอียดต่างๆ ของระบบ เ Service Operation ที่ต้องใช้งานเบื้องต้น รูปแบบของการนำเซอร์วิสมาประกอบกันหรือขอบเขตการทำงาน ของแต่ละเซอร์วิส เป็นต้น

การวิเคราะห์ระบบที่มีอยู่เดิมเพื่อค้นหาวิธีการแก้ปัญหาต่างๆ ที่เกิดขึ้นเป็นเรื่องที่ยุ่งยาก ซับซ้อน และต้องเสียค่าใช้จ่ายเป็นจำนวนมาก เพื่อสร้างกระบวนการสำหรับแก้ปัญหาดังกล่า ดังนั้นการวิเคราะห์ปัญหาที่เกิดขึ้น ควรกระทำอย่างเป็นแบบแผน สิ่งสำคัญที่ควรคำนึงถึงคือ การ อธิบายขั้นตอนต่างๆ ในการนำ SOA มาใช้ในองค์กรร่วมกับระบบที่มีอยู่เดิม โดยสามารถนำ Service-Oriented Analysis ไปใช้เพื่ออธิบายถึง Service Layer ที่ต้องนำมาใช้ในระบบได้

รูปที่ 2.6 แสดงขั้นตอนของ Service-Oriented Analysis

ขั้นตอนที่ 1 : รวบรวมความต้องการ

ขั้นตอนแรกเป็นการรวบรวมความต้องการทางธุรกิจที่เกิดขึ้น เพื่อใช้เป็นเอกสารในการ วิเคราะห์กระบวนการต่างๆ และกำหนดขอบเขต ในการวิเคราะห์ความต้องการ เพื่อใช้ในการสร้าง เซอร์วิสโดยกระบวนการต่างๆ ที่กำหนดขึ้นจะนำไปใช้เป็นข้อมูล ในการกำหนดแบบจำลอง ที่ใช้ อธิบายกระบวนการทำงานของระบบต่อไป

ขั้นตอนที่ 2 : วางแผนการทำงานของระบบเก่าและระบบใหม่

ความต้องการต่างๆ ที่กำหนดไว้ในขั้นตอนที่ 1 จะมีผลต่อการนำ Application logic ต่างๆ ของระบบเดิมมาใช้งาน แต่ Service-Oriented Analysis ไม่ได้กำหนดว่าจะต้องนำเว็บเซอร์วิสมาใช้งานแทนที่ หรือห่อหุ้มแอปพลิเคชั่นเดิมที่มีอยู่ โดยขั้นตอนนี้จะรวบรวมข้อมูลที่เกี่ยวข้องกับความสัมพันธ์ระหว่างเว็บเซอร์วิส กับระบบเดิมจะถูกกำหนดไว้ใน "ขั้นตอนการออกแบบ Design"

ขั้นตอนที่ 3 : กำหนด Service Modeling

แบบจำลองการทำงานต่างๆ ที่เรียกว่า "Service Modeling" เพื่อระบุถึง Operation ของเซอร์วิสและตัวจัดกลุ่มการทำงานให้กับแต่ละเซอร์วิส

ขั้นตอนการออกแบบ (Design)

เมื่อวิเคราะห์ความต้องการต่างๆ และทราบรายละเอียดของเซอร์วิสที่ต้องการสร้างแล้ว ขั้นตอนต่อไปคือ ออกแบบ (Design) โครงสร้างต่างๆ ของระบบ การออกแบบจะต้องเป็นไปตามแบบแผนของมาตรฐานต่างๆ ที่นำมาใช้ และอยู่บนหลักการของ SOA ด้วย โดย Service Layer จะถูกออกแบบในขั้นตอนนี้ เรียกขั้นตอนนี้ว่า "Service-Oriented Design" คือ กระบวนการออกแบบเซอร์วิสที่ได้วิเคราะห์ไว้ โดยวัตถุประสงค์เพื่อออกแบบส่วนประกอบต่างๆ ได้แก่ กำหนดกลุ่มของสถาปัตยกรรม สำคัญที่ต้องใช้ขอบเขตของสถาปัตยกรรมกำหนดมาตรฐานการออกแบบ และออกแบบส่วนติดต่อกับเซอร์วิสกระบวนการออกแบบจะต้องนำข้อมูลต่างๆ ที่ได้จัดเตรียมไว้ในขั้นตอนการวิเคราะห์มาใช้งานโดย Service-Oriented Design ประกอบด้วย 3 ขั้นตอนสำคัญได้แก่

รูปที่ 2.7 แสดงขั้นตอนของ Service-Oriented Design

ขั้นตอนที่ 1 : กำหนดเทคโนโลยีมาตรฐาน

ขั้นตอนแรกเป็นการกำหนดเทคโนโลยี และสถาปัตยกรรมต่างๆ ที่จะนำมาใช้งานร่วมกัน โดนประกอบด้วย 3 ขั้นตอนสำคัญคือ (1) การกำหนด Service Layer (2) เลือกมาตรฐานที่นำมาใช้ และ (3) กำหนดข้อกำหนดเพิ่มเติม

ขั้นตอนที่ 2 :ออกแบบเซอร์วิส

เมื่อกำหนดเทคโนโลยีต่างๆ และเลือกมาตรฐานที่ต้องการใช้งานแล้ว ขั้นตอนต่อไปจะเป็นการออกแบบเซอร์วิสซึ่งประกอบด้วย 3 กระบวนการสำคัญคือ (1) การออกแบบ Entity-Centric Business Service (2) การออกแบบ Application Service และ (3) การออกแบบ Task-Centric Business Service

ขั้นตอนที่ 3 : กำหนดขั้นตอนการทำงานของเซอร์วิส

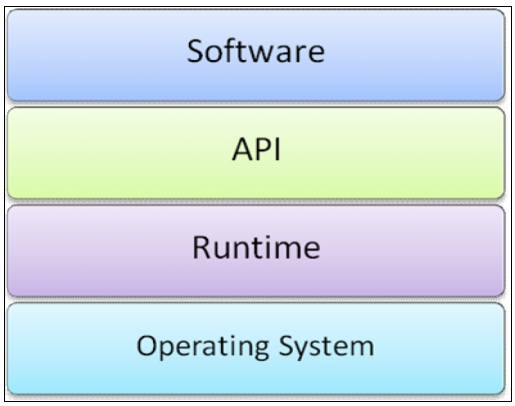
Service ต่างๆ ที่ได้ออกแบบไว้ จะต้องนำมาสร้าง Orchestration Layer โดยการผูกเซอร์วิสกับลอจิกต่างๆ เพื่อกำหนดขั้นตอนการดำเนินงาน(Workflow)

ขั้นตอนการพัฒนา (Development)

เมื่อออกแบบเซอร์วิสแล้ว ขั้นตอนต่อไปเป็นการพัฒนาเซอร์วิสที่ได้ออกแบบไว้ ด้วยการใช้ภาษาในการพัฒนาและแพลตฟอร์มที่ระบุไว้เช่น J2EE หรือ .NET เป็นต้น โดยขั้นตอนนี้จะไม่คำนึงถึงชนิดของเซอร์วิสมากนัก แต่จะเน้นไปที่การกำหนดรูปแบบของเซอร์วิสในเชิงกายภมากกว่า

แนวคิด ข้อกำหนด และเทคโนโลยีเปิดต่างๆ ที่ได้กล่าวถึงจำเป็นต้องนำไปใช้พัฒเป็นซอฟต์แวร์ด้วยแพลตฟอร์มต่างๆ ที่รองรับเทคโนโลยีของเว็บเซอร์วิส โดยแพลตฟอร์มเหล่านี้ได้จัดเตรียมเครื่องมือไว้สำหรับรองรับความต้องการเพื่อพัฒนาซอฟต์แวร์ของเว็บเซอร์วิส ซึ่งตอบสนองความต้องการต่างๆ ดังนี้

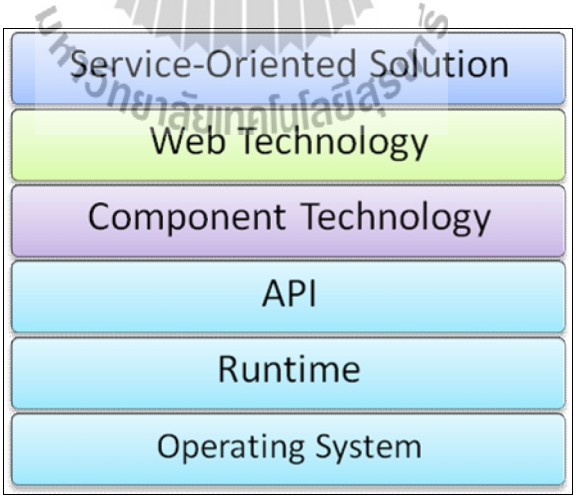
1. จัดเตรียมเครื่องมือสำหรับรองรับการพัฒนาโปรแกรมด้วยภาษาที่ใช้ในการพัฒเต่างๆ
2. จัดเตรียม API ที่มีเครื่องมือและฟังก์ชั่นการทำงานต่างๆ ที่สนับสนุนการทำงานแบบ Runtime ทำให้สามารถสร้างซอฟต์แวร์ได้อย่างมีประสิทธิภาพมากขึ้น
3. ซอฟต์แวร์ API และ Runtime ต้องได้รับการสนับสนุนจาก OS (Operating System)



รูปที่ 2.8 แสดงชั้น (Layer) ของ Software Technology

อย่างไรก็ตามการพัฒนาเว็บเซอร์วิส จำเป็นต้องเพิ่มความต้องการอีก 2 อย่าง

- ความสามารถในการห่อหุ้ม และแสดง Application Logic ด้วยเทคโนโลยีของเว็บเซอร์วิส
- ความสามารถในการแบ่งซอฟต์แวร์ และการแยกแยะคอมโพเนนท์หรือลอจิกต่างๆ ที่มีอยู่ภายในเพื่อความสามารถในการติดต่อสื่อสารบนแพลตฟอร์ม และ Runtime ที่แตกต่างกัน ลักษณะดังกล่าว ทำให้สามารถขยายรายละเอียดในชั้น (Layer) ได้ดังภาพ



รูปที่ 2.9 แสดงชั้น (Layer) ที่ใช้สำหรับพัฒนา SOA

จากรูป เทคโนโลยีต่างๆ ในแต่ละ layer จำเป็นต้องสนับสนุนหลักการของ SOA โดยมีความสัมพันธ์ดังนี้

- API จะต้องจัดเตรียมฟังก์ชั่นการทำงานต่างๆ ที่สนับสนุนการพัฒนากระบวนการต่างๆ ของ Component และเว็บเซอร์วิส
- Runtime จะต้องจัดเตรียมกลุ่มของ API ที่สนับสนุน Component และเว็บเซอร์วิสไว้ นอกจากนี้ยังต้องทำหน้าที่จัดเตรียมพื้นที่ (Hosting) ไว้ให้กับ Component และเว็บเซอร์วิส ด้วย
- Component Technology ต้องสนับสนุนการห่อหุ้มด้วยเว็บเซอร์วิส
- Web Technology ต้องสนับสนุนเทคโนโลยีของเว็บเซอร์วิส ในยุคแรก (First Generation) เพื่อรองรับหลักการต่างๆ ของ Primitive SOA และต้องสนับสนุนข้อกำหนดของ WS-* เพื่อรองรับคุณลักษณะต่างๆ ของ Contemporary SOA นอกจากนี้ยังต้องเตรียมเครื่องมือสำหรับรวบรวม และนำเทคโนโลยีต่างๆ ไปใช้ภายในเว็บเซอร์วิสอีกด้วย

ขั้นตอนการทดสอบ (Testing)

เมื่อพัฒนาเซอร์วิสแล้ว จะต้องนำเซอร์วิสมาทดสอบอย่างเป็นระบบ เพื่อให้ได้เซอร์วิสทำงานได้อย่างสมบูรณ์ ก่อนจะนำไปใช้งาน (Deployment) ในระบบต่อไป ตัวอย่างของการทดสอบเซอร์วิส เช่น ทดสอบการติดต่อกับ Service Description ของเซอร์วิส หรือทดสอบความถูกต้องเกี่ยวกับสิทธิ์ในการเรียกใช้เซอร์วิสของแต่ละ Service Requester เป็นต้น

ขั้นตอนการนำไปใช้ (Deployment)

เมื่อทดสอบเซอร์วิสแล้ว ขั้นตอนต่อไปคือ การติดตั้งองค์ประกอบต่างๆ ลงบนเครื่อง Server เช่น Component และ Service Interface ต่างๆ รวมทั้งซอฟต์แวร์ที่ทำหน้าที่เป็นตัวกลางคอยประสานและให้บริการ (Middleware) ให้เหมาะสมกับแพลตฟอร์มที่ใช้พัฒนาเซอร์วิส

ขั้นตอนการบริหารและจัดการ (Administration)

เมื่อติดตั้งเซอร์วิสแล้ว สิ่งที่ควรให้ความสำคัญอย่างยิ่ง คือ การบริหาร และการจัดการระบบ โดยการตรวจสอบร่องรอยเพื่อจัดการกับ Message ตรวจสอบการใช้เซอร์วิสและ ตรวจสอบประสิทธิภาพในการให้บริการของระบบให้เป็นไปตามมาตรฐานที่กำหนดไว้ โดยใช้วิธีเหมือนกับ การบริหารและจัดการ Distributed Application และ Component-Base Application

2.2 เว็บเซอร์วิส (Web Service)

2.2.1 ความเป็นมาของเว็บเซอร์วิส

W3C ได้นำเสนอข้อกำหนดของ Simple Object Access Protocol (SOAP) ขึ้นในปี 2000 เริ่มแรกข้อกำหนดนี้ถูกออกแบบมาเพื่อการติดต่อสื่อสารแบบ RPC (Remote Procedure Call) ให้เป็นไปในทิศทางเดียวกัน แนวความคิดนี้ใช้เพื่อส่งผ่าน parameter ของข้อมูลระหว่าง Component ที่ถูกจัดเก็บในรูปแบบ XML แล้วส่งออกไปจากนั้นจึงแปลงกลับไปเป็นรูปแบบเดิม

ต่อมาบริษัทและผู้ผลิตซอฟต์แวร์ต่างๆ เริ่มเห็นประโยชน์มากของระบบ ธุรกิจที่สร้างอยู่บนโครงสร้างการติดต่อสื่อสารที่เป็นอิสระผ่านอินเตอร์เน็ต ทำให้เกินเทคโนโลยี Distribute ขึ้นมาจนกลายเป็นโครงสร้างมาตรฐานสำหรับการติดต่อสื่อสารที่ใช้เพื่อแก้ปัญหาที่เกิดจากความแตกต่างของการสื่อสารข้อมูลทั้งภายในและระหว่างองค์กร โดยเรียกแนวคิดนี้ว่าเว็บเซอร์วิส

การประกาศวิธีติดต่อสื่อสารกับเซอร์วิสเป็นกลไกสำคัญของการพั SOA ด้วยเทคโนโลยีเว็บเซอร์วิส โดยจะใช้ Web Service Description Language (WSDL) ซึ่งอยู่ในรูปแบบของ XML เป็นแหล่งเก็บข้อมูลที่แสดงรายละเอียด และการเรียกใช้งานเซอร์วิสไว้

โดย W3C ได้นำเสนอข้อกำหนดของ WSDL เป็นครั้งแรกในปี ค.ศ. 2001 และได้ปรับปรุงข้อกำหนดต่างๆ มาอย่างต่อเนื่องจนถึงปัจจุบัน

การติดต่อสื่อสารของเว็บเซอร์วิสจะอยู่ในรูปแบบของ Messaging โดยใช้ SOAP Protocal สำหรับรับส่ง Message ระหว่างกัน

UDDI คือเทคโนโลยีที่ทำให้แนวคิดของเว็บเซอร์วิสในยุคแรกสามารถทำงานได้อย่สมบูรณ์ โดยได้รับการพัฒนาจากความร่วมมือของ UDDI.org และ OASIS ให้เป็นข้อกำหนดมาตรฐานในการลงทะเบียน Service Description (WSDL) ทำให้ผู้จัดเตรียมบริการ (Service Provider) สามารถลงทะเบียนเซอร์วิสไว้ในสถานที่ ซึ่งผู้ร้องขอบริการ (Service Requester) สามารถค้นหาบริการได้ (สุธี พงศาสกุลชัย, 2550)

2.2.2 ความหมายของเว็บเซอร์วิส

Web Service คือ โปรแกรมประยุกต์ที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากโปรแกรมประยุกต์ อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยมีสื่อกลางในการแลกเปลี่ยนคือภาษา XML ทำให้สามารถเรียกใช้ component ใดในแพลตฟอร์มใด ๆ ก็ได้ บน protocol HTTP ซึ่งเป็น protocol สำหรับ World Wide Web ซึ่งเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่างโปรแกรมประยุกต์ในปัจจุบัน

Web Service ช่วยให้การเข้าถึงข้อมูลสารสนเทศจากโปรแกรมประยุกต์ที่ต่างกันเป็นไปโดยง่าย โดยที่โปรแกรมประยุกต์นั้นๆ สามารถเขียนด้วยภาษาที่แตกต่างกันทำงานภายใระบบปฏิบัติการหรือแพลตฟอร์มที่แตกต่างกันได้ ซึ่งมาตรฐานของเว็บเซอร์วิสจะทำการอธิบายอินเทอร์เฟซของโปรแกรมประยุกต์เหล่านี้ โดย WSDL (Web Service Description language) และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกันโดย XML ผ่าน SOAP(Simple Object Access Protocal) อินเตอร์เฟซ

Web Service สามารถถูกเรียกใช้ภายในองค์กรเองหรือจากภายนอกองค์กร โดยผ่านไฟร์วอล์ได้ ดังนั้นจึงมีองค์กรใหญ่ๆ มากมาย มีความสนใจที่จะพัฒนาระบบที่มีอยู่ของตน ให้เข้ากับมาตรฐานของเว็บเซอร์วิสซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจากเว็บเซอร์วิสสามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้อีกทางหนึ่ง

2.2.3 เทคโนโลยีที่ใช้ในการพัฒนาเว็บเซอร์วิส

XML (The Extensible Markup Language)

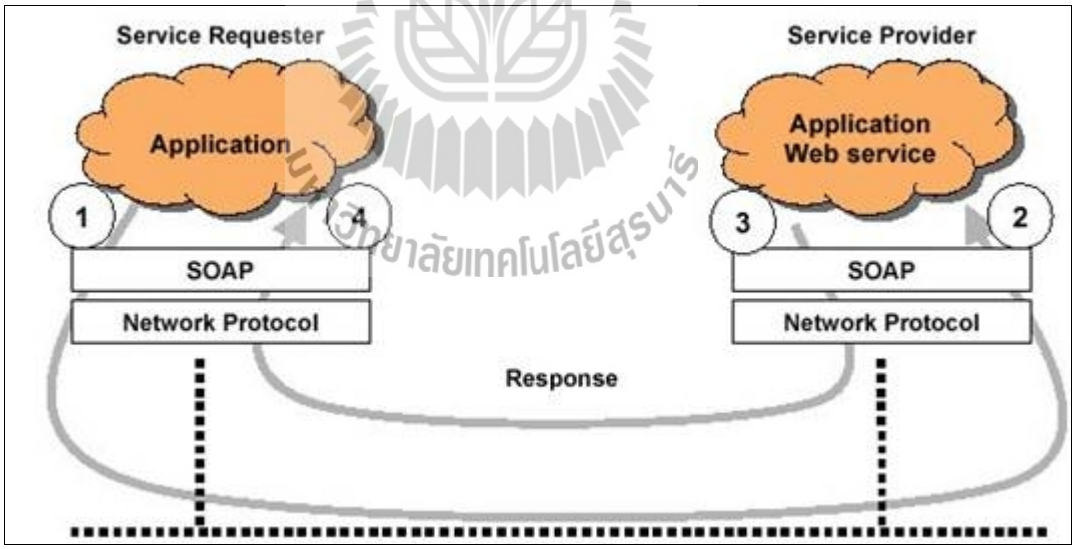
เป็นภาษา Markup ที่เป็น text-based ซึ่งทำให้เป็น มาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตอย่างรวดเร็ว ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ XML คือ World Wide Web Consortium (W3C) ความแตกต่างระหว่าง XML กับ HTML คือ HTML ถูกนำมาใช้ในการสร้าง เว็บเพจ ที่สามารถแสดงผลได้โดยโปรแกรมบราวเซอร์ แต่ XML จะใส่ tags ได้อย่างอิสระแล้วทำการส่ง XML ชุดนี้ไป ประมวลผลยังแอพพลิเคชั่นใด ๆ ที่สามารถใช้ข้อมูลใน XML นี้ XML เป็นภาษาที่มีลักษะเป็น tag คล้าย HTML แต่ไม่ได้มุ่งที่การแสดงผล XML มุ่งที่การสื่อความหมายโดยอนุญาตให้ผู้ใช้สามารถกำหนด tag ขึ้นได้เองเพื่อให้สื่อความหมายทางภาษาของมนุษย์ แต่คอมพิวเตอร์เองก็เข้าใจเช่นกัน ทำให้ข้อมูลระหว่าง tag สามารถนำไปประมวลผลต่อได้ เช่น

```
<ComputerBook>
        <book>
            <name>เว็บเซอร์วิส</name>
            <price>10.00$</price>
        </book>
        <book>
            <name>xml</name>
            <price>10.00$</price>
        </book>
</ComputerBook>
```

จากตัวอย่าง จะเห็นได้ว่ารูปแบบเอกสาร xml ไม่ได้บอกวิธีแสดงผลไว้ แต่เอกสารสามารถสื่อความหมายให้คอมพิวเตอร์เข้าใจได้ และนำค่าไปประมวลผลต่อได้

SOAP(Simple Object Access Protocol)

SOAP (Simple Object Access Protocol) เป็น XML-based โปรโตคอล (lightweight protocol)และใช้ HTTP เป็นโปรโตคอลร่วม สำหรับ การแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (decentralized, distributed environment) SOAP ได้ กำหนด Messaging Protocol ระหว่างผู้ขอบริการ (requestor) กับผู้ให้บริการ (provider) เช่น ผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช่ RMI (Remote Method Invocation) ตามวิธีการของโปรแกรมเชิงวัตถุ บริษัท Microsoft , IBM, Lotus , UserLand และ DeveloperMenter ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น SOAP ได้กำหนดรูปแบบพื้นฐานของการสื่อสารแบบกระจายขึ้นตามรูปแบบการพัฒนา SOA แม้ว่า SOA จะไม่ได้กำหนด Messaging Protocol ไว้ แต่ SOAP ก็ได้ถูกกำหนด ให้เป็น Services-Oriented Architecture Protocol เนื่องจากมันได้ถูกใช้ในการพัฒนา SOA อย่างแพร่หลายนั่นเอง จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง การพัฒนาอนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์มระบบปฏิบัติการ, แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา



รูปที่ 2.10 แผนภาพแสดงการทำงานของ SOAP Protocol

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอพพลิเคชั่น มีการทำงานกับแอพพลิเคชั่นที่ทำงานอยู่ในเครื่องอื่นโดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่าง Object ในระยะไกล (Remote Procedure Calls : RPC) เช่น

DCOM,EJB หรือ CORBAนั้นไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอ HTTP (เครือข่ายที่ปัจจุบันใช้งานแพร่หลาย และสะดวกมากที่สุดได้แก่เครือข่าย Internet ซึ่งให้การสื่อสารผ่านโปรโตคอล HTTP เป็นส่วนใหญ่)

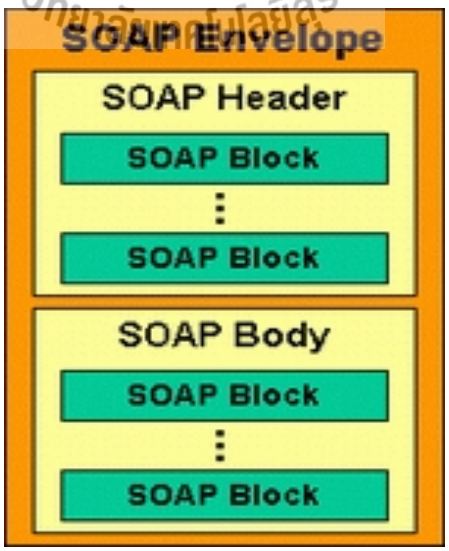
เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย ยกเว้น CORBA ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อ และยังมีปัญหาในส่วนของ Firewalls และ Proxy Server ด้วยเนื่องจากโดยปกติ เซิฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอล HTTP ออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำบริการเว็บเซอร์วิส ก็คือให้ทำงานอยู่บนโปรโตคอล HTTP ไปเลย ซึ่ง SOAP นอกจากจะทำงานบนโปรโตคอล HTTP แล้วยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ, เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วยก็ได้

โครงสร้างของ SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้
3. SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส



รูปที่ 2.11 แสดงโครงสร้างของเอกสาร SOAP

ตัวอย่างเอกสาร SOAP

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
                soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
                        <soap:Header>
                                <!-- ข้อมูลในส่วนของ Header -->
                                <i:local xmlns:i="http://www.i3t.or.th/ws/">
                                <i:currency>Bath</i:currency>
                                </i:local>
                        </soap:Header>
                        <soap:Body>
                                <!-- ข้อมูลในส่วนของ Body -->
                                <GetPrice>
                                        <Item>Rose</Item>
                                        <Quantity>100</Quantity>
                                </GetPrice>
                        </soap:Body>
                        <soap:Fault>
                        <!-- ข้อมูลของ SOAP ในกรณีมีข้อผิดพลาด จาก SOAP Node -->
                        </soap:Fault>
                </soap:Envelope>
        <soap:Fault>
                <faultcode>Client</faultcode>
                <faultstring>Invalid Request</faultstring>
        </soap:Fault>
        <soap:Fault>
                <faultcode>Client</faultcode>
```

ตารางที่ 2.1 อธิบาย Element ของ SOAP Message

Element	**Definition**
<Envelope>	จะเป็น Root ของเอกสาร XML SOAP เสมอ โดยจากตัวอย่างจะใช้ Namespaces จาก "http://schemas.xmlsoap.org/soap/envelope/" (Schema for the SOAP/1.1 envelope) ดังนั้น Element ต่างๆ ในตัวอย่างนี้ที่อยู่ในเอกสาร XML ส่วนที่เป็นมาตรฐาน ของ SOAP เราจะมี Namespaces ที่เราตั้งเป็น soap (<soap:Envelop>, <soap:Header>, <soap:Body>, <soap:Fault> **Attribute ใน Envelop** Attribute encodingStyle ในตัวอย่างมีการอ้างการ encoding จาก "http://schemas.xmlsoap.org/soap/encoding/" (Schema for the SOAP/1.1 encoding) จะใช้ในกรณีที่เรามีการใช้ Parameter หลายชนิดข้อมูลในเอกสาร (Data Type)

	เช่น Boolean, String, Integer เป็นต้น
<Header>	จากตัวอย่างมีการเพิ่มเนื้อหาของเอกสาร SOAP เข้าไปในส่วนของ Header ในที่นี้คือ สกุลเงิน (Currency) ซึ่งส่วนนี้จะใช้ในส่วนของ Application ไม่ได้เป็นส่วนของมาตรฐานของ SOAP แต่ผู้ใช้งานเป็นผู้กำหนดใช้เอง (User-defined **Attribute ใน Header** ใน <Header> Element อาจจะมีการใส่ Attribute mustUnderstand เพื่อให้ฝั่งที่รับเอกสาร SOAP ให้ทำการประมวลผลในส่วน Header ด้วยโดยกำหนดค่าเป็น Boolean เช่น <i:local xmlns:i="http://www.i3t.or.th/ws/"> <i:country mustUnderstand="1">Thailand</i:country> <i:currency mustUnderstand="0">Bath</i:currency> </i:local> ค่าของ mustUnderstand ถ้าเป็น "0" แสดงว่าไม่ต้องประมวลผล ถ้าเป็น "1" : true คือจำเป็นต้องมีการประมวลผล (โดยค่า default = "0" : fault)
<Body>	โดยปกติแล้วเอกสาร SOAP จะต้องมีส่วนของ Body เพราะเป็นส่วนเป็นเนื้อหาสาระจริงๆ ของ SOAP จากตัวอย่างเราต้องการสอบถามราคาของดอกกุหลาบจำนวน 100 ดอก ซึ่ง <GetPrice>, <Item> และ <Quantity>เป็น Element ที่ใช้งานใน Application ไม่ใช่มาตรฐานของ SOAP
<Fault>	เป็นส่วนที่จะถูกใช้เมื่อมีข้อผิดพลาดในการประมวลผลของเอกสาร SOAP ซึ่งโดยปกติจะเห็นเฉพาะที่เป็นเอกสารตอบกลับเท่านั้น (Reply Message หรือ Answer Message) ซึ่งตัวอย่างจะเป็นเอกสารที่เป็นการร้องขอบริการ (Request Message) ซึ่งไม่น่าจะมีส่วนของ <Fault> Element แต่ที่เขียนไว้เพื่อให้เห็นภาพรวมของโครงสร้างเอกสาร SOAP ทั้งหมดก่อน **ตัวอย่างข้อมูลใน <Fault> Element ที่มีการเกิดข้อผิดพลาด** <soap:Fault> <faultcode>Client</faultcode> <faultstring>Invalid Request</faultstring> </soap:Fault>

ตารางที่ 2.2 อธิบาย fault Element

Sub Element	**คำอธิบาย**
<faultcode>	มีข้อผิดพลาดของการอ้าง Namespace ของ Element
<faultstring>	ในการประมวลผลของ Sub Element ใน Header ที่มีการกำหนด attribute

	mustUnderstand เป็น "1" ไม่สามารถประมวลผลได้ (not understood)
<faultactor>	เอกสาร SOAP มีข้อผิดพลาดอันเนื่องมาจากมีโครงสร้างทางผิด หรือมีข้อมูลที่ไม่ถูกต้อง
<detail>	มีข้อผิดพลาดจาก Server ไม่สามารถประมวลผลเอกสารได้

ตารางที่ 2.3 อธิบาย Fault code Sub Element

Error (faultcode)	**คำอธิบาย**
VersionMismatch	มีข้อผิดพลาดของการอ้าง Namespace ของ Element
MustUnderstand	ในการประมวลผลของ Sub Element ใน Header ที่มีการกำหนด attribute mustUnderstand เป็น "1" ไม่สามารถประมวลผลได้ (not understood)
Client	เอกสาร SOAP มีข้อผิดพลาดอันเนื่องมาจากมีโครงสร้างทางผิด หรือมีข้อมูลที่ไม่ถูกต้อง
Server	มีข้อผิดพลาดจาก Server ไม่สามารถประมวลผลเอกสารได้

จะเห็นว่า SOAP Envelope ก็เป็นเอกสารที่อยู่ในรูปแบบ XML ทั่วไป แต่สิ่งที่ทำให้ SOAP มีความสามารถมากขึ้นก็คือ SOAP ได้มีการกำหนดโครงสร้างของเอกสารเป็นส่วนๆ โดยอธิบายว่าส่วนใดมีหน้าที่อะไร (What) มีข้อมูลอะไรอยู่ในส่วนนั้น และใคร (Who) คือผู้ที่จะต้องสนใจในส่วนนั้น ๆ และกล่าวถึงว่าส่วนใดจำเป็นที่ต้องมีในเอกสาร และส่วนใดเป็นส่วนที่เพิ่มเติมอาจจะมีหรือไม่มีส่วนนั้นก็ได้ขึ้นอยู่ความจำเป็นในการใช้งาน ทำให้การพัฒนาโปรแกรมมีทิศทางที่ค้อนข้างชัดเจน และสามารถเพิ่มขยายได้ในอนาคตอีกด้วย

WSDL (Web Services Description Language)

เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของเว็บเซอร์วิสและวิธีการติดต่อกับเว็บเซอร์วิสความต้องการของนิยามนี้เกี่ยวเนื่องกับความต้องการของ distributed system ที่จะกำหนด Interface Definition Language(IDL) โดยใช้ภาษา XML, WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัทไอบีเอ็ม, SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ใน ปัจจุบันคือ WSDL 1.2

WSDL คือ มาตรฐานสำหรับการประกาศ process ที่จำเป็นในการเรียกใช้เซอร์วิส SOAP (Simple Object Access Potocal)

โครงสร้างเอกสาร WSDL

WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) version ที่มีอยู่ในปัจจุบัน คือ WSDL 1.1 ในการใช้งานจริง หากเราสร้างบริการเว็บเซอร์วิสก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับเว็บเซอร์วิส อย่างอัตโนมัติ จุดภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของเว็บเซอร์วิสมีจุดที่ควรทราบ ดังนี้

ตารางที่ 2.4 อธิบาย Element ของ WSDL

Element	**Definition**
<port Type>	เป็นส่วนที่สำคัญที่สุดในWSDL element อธิบาย operations ที่ web service มีให้บริการและ messages ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียนโปรแกรม
<operation>	อธิบาย method ที่ให้บริการ Web Services หนึ่งจะมี method จำนวนกี่ method ก็ได้
<message>	อธิบาย data elements ของ operation แต่ละ message อาจมีมากกว่าหนึ่ง ส่วนเทียบได้กับ parameter ของ function ในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลที่ web service ใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล
<binding>	อธิบาย format ของ message และ protocol details ในแต่ละ port
<service>	สำหรับ web server จะมี Web Services จำนวนกี่บริการก็ได้ และ ชื่อ Web Service ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน

ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการเว็บเซอร์วิสได้มากกว่า 1 บริการ โดยแต่ละเว็บเซอร์วิสจะมี port สื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

ตัวอย่างเอกสาร WSDL

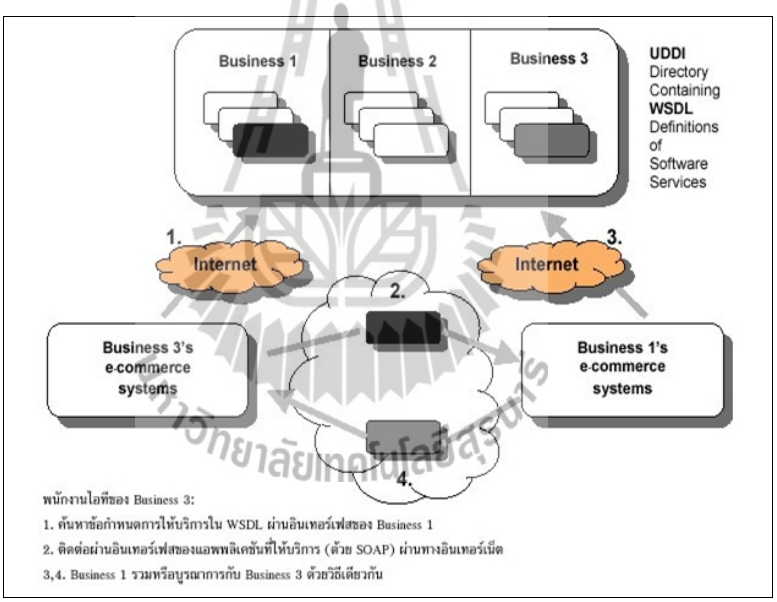
```
<wsdl:definitions name="nmtoken"? targetNamespace="uri">
        <import namespace="uri" location="uri"/> *
        <wsdl:documentation .... /> ?
        <wsdl:types> ?
                <wsdl:documentation .... /> ?
                <xsd:schema .... /> *
        </wsdl:types>
```

```
<wsdl:message name="ncname"> *
    <wsdl:documentation .... /> ?
    <part name="ncname" element="qname"? type="qname"?/>
*
</wsdl:message>
<wsdl:portType name="ncname"> *
    <wsdl:documentation .... /> ?
    <wsdl:operation name="ncname"> *
        <wsdl:documentation .... /> ?
        <wsdl:input message="qname"> ?
            <wsdl:documentation .... /> ?
        </wsdl:input>
        <wsdl:output message="qname"> ?
            <wsdl:documentation .... /> ?
        </wsdl:output>
        <wsdl:fault name="ncname" message="qname"> *
            <wsdl:documentation .... /> ?
        </wsdl:fault>
    </wsdl:operation>
</wsdl:portType>
<wsdl:serviceType name="ncname"> *
    <wsdl:portType name="qname"/> +
</wsdl:serviceType>
<wsdl:binding name="ncname" type="qname"> *
    <wsdl:documentation .... /> ?
    <-- binding details --> *
    <wsdl:operation name="ncname"> *
        <wsdl:documentation .... /> ?
        <-- binding details --> *
        <wsdl:input> ?
            <wsdl:documentation .... /> ?
            <-- binding details -->
        </wsdl:input>
        <wsdl:output> ?
            <wsdl:documentation .... /> ?
            <-- binding details --> *
        </wsdl:output>
        <wsdl:fault name="ncname"> *
            <wsdl:documentation .... /> ?
            <-- binding details --> *
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ncname" serviceType="qname">        *
    <wsdl:documentation .... /> ?
    <wsdl:port name="ncname" binding="qname"> *
        <wsdl:documentation .... /> ?
```

```
                    <-- address details -->
                </wsdl:port>
            </wsdl:service>
        </wsdl:definitions>
```

UDDI (Universal Description, Discovery, and Integration)

เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI ใช้สำหรับค้นหาเซอร์วิสที่ต้องการ และเมื่อได้มาแล้ว UDDI ยังจัดหาข้อตกลงในวิธีการที่จะใช้งานเปรียบได้กับสมุดหน้าเหลือง เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟต์ และบริษัทอารีบา (Ariba) ปัจจุบันมีบริษัทที่ร่วม กันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability (Dan Gisolfi, 2001)



รูปที่ 2.12 Web Service with SOAP, UDDI และ WSDL

2.3 ระบบบริหารจัดการงานขนส่ง

(วิศิษฏ์ วัฒนานุกูล, 2552: 45 – 49)

2.3.1 การขนส่งกับโครงสร้างพื้นฐาน

ปัจจุบันลอจิสติกส์มีความสำคัญกับธุรกิจทุกประเภท และเป็นปัจจัยสำคัญปัจจัยหนึ่งต่อการสร้างความได้เปรียบในการแข่งขันในตลาดโลก ซึ่งนับวันยิ่งทวีความรุนแรงมากขึ้นเรื่อยๆ ธุรกิจในปัจจุบันวัดความได้เปรียบในการแข่งขัน (Competitive Advantage) กันที่ความรวดเร็วและต้นทุนที่ถูกลง ดังจะเห็นได้จากการลดต้นทุนที่ไม่ก่อให้เกิดรายได้ (Non-Value Added Cost) หรือการสร้างมูลค่าเพิ่มให้เกิดขึ้นในระหว่างกระบวนการจัดส่งสินค้าไปสู่ผู้บริโภค ที่เรียกว่า การบริหารจัดการระบบลอจิสติกส์ ซึ่งครอบคลุมกิจกรรมตั้งแต่การบริหารการขนส่งการบริหารวัสดุคงคลัง การบริหารคลังสินค้า การบริหารคำสั่งซื้อ การบริหารข้อมูล การบริหารการเงิน การจัดการวัตถุดิบ การจัดซื้อ การบรรจุ และการบริหารความต้องการ(อุปสงค์) เพื่อให้สามารถนำเสนอสินค้าหรือบริการที่มีคุณภาพไปถึงมือลูกค้าในเวลาที่ต้องการ บนพื้นฐานการบริหารต้นทุนที่มีประสิทธิภาพมากที่สุด เป็นเรื่องที่ประเทศต่างๆ ทั่วโลกให้ความสำคัญทั้งในภาคธุรกิจอุตสาหกรรม และภาครัฐ รวมทั้งประเทศไทย ซึ่งรัฐบาลโดยคณะกรรมการพัฒนาระบบการบริหารจัดการขนส่งสินค้าและบริการแห่งประเทศไทย (กบส.) ได้เล็งเห็นความสำคัญและทุ่มงบประมาณมหาศาลเพื่อเพิ่มประสิทธิภาพระบบขนส่งและโลจีสติกส์

นอกเหนือจากความจำเป็นที่ต้องปรับตัว เพื่อแข่งขันได้ในสถานการณ์โลกที่ทวีความรุนแรงมากขึ้นทุกขณะ ประเทศไทยยังต้องเผชิญกับสถานการณ์ราคาน้ำมันที่มีความผันผวนตลอดเวลา ซึ่งย่อมส่งผลกระทบเป็นวงกว้างต่ออุตสาหกรรมทั้งขนาดใหญ่ และอุตสาหกรรมเล็กถึงกลางหรือ SMEs ในด้านการเพิ่มขึ้นของต้นทุนการผลิตและด้านขนส่ง จากข้อเท็จจริงพบว่า “ในภาวะราคาน้ำมันเพิ่มสูงขึ้นนั้น ไม่ได้มีผลกระทบโดยตรงกับต้นทุนทางตรงในการผลิต การค้าและบริการในระดับที่สูงมากนัก หากแต่มีผลกระทบทางอ้อมในระดับที่สูงกว่าในเรื่องต้นทุนการขนส่ง”

ความสำคัญในการพัฒนาเรื่องลอจิสติกส์นั้น ที่ผ่านมาประเทศไทยถือว่าพัฒนาไปอย่างค่อนข้างล่าช้า มีการลงทุนแต่ละปีด้วยงบประมาณนับแสนล้านบาทเพื่อตัดถนนไปทั่วประเทศ แต่ไม่เคยมีการนำระบบบริหารจัดการการขนส่งที่ดีมาใช้ เป็นผลให้ต้นทุนด้านการขนส่งของประเทศไทยในบางปีสูงถึง 25 – 30 % ของผลผลิตมวลรวมประชาชาติ (GDP) ในขณะที่ประเทศที่พัฒนาแล้วมีต้นทุนอยู่ที่ 10% ต้นๆ เท่านั้น ทำให้เราเสียเปรียบด้านการแข่งขัน แต่ละปีสูญเปล่าพลังงานน้ำมันไม่ต่ำกว่า 20,000 ล้านบาท (วิศิษฏ์ วัฒนานุกูล, 2552)

ประเด็นสำคัญคือ เมื่อมีการเปิดเสรีด้านการขนส่งขึ้นตามกรอบที่ไทยเป็นสมาชิกขององค์การค้าโลก (WTO) มีบริษัทลอจิสติกส์ข้ามชาติมากมายเข้ามาประกอบการในประเทศ ซึ่งบริษัทเหล่านี้มีรถขนสินค้าที่มีประสิทธิภาพสูงและมีสภาพใหม่ มีระบบจัดการที่ดีด้วยเทคโนโลยีสมัยใหม่ มีบุคลากรที่มีคุณภาพ สภาวะการเช่นนี้ทำให้ประเทศเกิดการเสียดุล เพราะเท่ากับว่าเรามีหน้าที่เพียงจัดสรรงบประมาณในการซ่อมแซมถนนให้รถบรรทุกเหล่านี้วิ่งโดยเราไม่ได้อะไร หัวใจสำคัญที่สุดคือ เส้นเลือดการขนส่งของประเทศเหมือนหนึ่งตกอยู่ในมือต่างชาติ หมดโอกาสพึ่งพาตนเองด้านระบบขนส่งสินค้าในที่สุด

จึงจำเป็นอย่างยิ่งที่ภาครัฐต้องวางกลยุธศาสตร์ด้านการขนส่งและลอจิสติกส์ให้ดี เร่งพัฒนาโครงสร้างพื้นฐาน ที่สำคัญที่สุดคือ การพัฒนาคน การเปลี่ยนแปลงรูปแบบการขนส่ง และเริ่มนำเทคโนโลยีใหม่ๆ มาใช้ อาศัยความร่วมมือของภาครัฐและเอกชนอย่างจริงจัง มิฉะนั้นธุรกิจบริการด้านการขนส่งและลอจิสติกส์จะตกเป็นของต่างชาติ

2.3.2 เทคโนโลยีที่เกี่ยวกับการขนส่ง

เทคโนโลยีที่เกี่ยวกับการขนส่งที่สำคัญที่นำมาใช้ในการบริหารจัดการลอจิสติกส์ซัพพลายเชนหรือเรียกอีกอย่างหนึ่งว่า ระบบบริหารจัดการงานขนส่ง ซึ่งจะกล่าวถึงที่นี้ ได้แก่

- ระบบบริหารจัดการกองรถขนส่ง (Fleet Management System)
- ระบบการวางแผนและจัดเส้นทางเดินรถ (Vehicle Routing and Planning)
- ระบบตรวจหาตำแหน่งละควบคุมการเดินรถ (Vehicle Based System)

1. ระบบบริหารจัดการกองรถขนส่ง (Fleet Management System)

ระบบนี้เป็นระบบเดียวที่มักจะดำเนินการโดยหน่วยงานภายในองค์กรเอง วัตถุประสงค์หลักของการติดตั้งระบบนี้ ก็เพื่อใช้ในการบริหารจัดการรถบรรทุกที่มีจำนวนมากเกินกว่าที่จะบริหารจัดการด้วยคน โดยไม่มีเทคโนโลยีเข้ามาช่วย โดยเฉพาะเรื่องต่างๆ ดังต่อไปนี้

1.1 การจัดทำตารางบำรุงรักษา (Maintenance Scheduling)

1.2 การควบคุมปริมาณอะไหล่รถบรรทุก(Vehicle parts Control, Stock Re-ordering and Inventory Control)

1.3 ธุรการงานบรรทุก (Fleet Administration)

1.4 การควบคุมดูแลระหว่างการใช้งาน (Operational Monitoring)

2. ระบบการวางแผนและเส้นทางการเดินรถ (Vehicle Routing and Planning)

เป็นอีกระบบหนึ่งของการบริหารจัดการการขนส่งที่มีความสำคัญไม่น้อย ซึ่งมีผลต่อเนื่องกับการลดค่าใช้จ่ายของค่าขนส่ง กล่าวคือ ช่วยให้การควบคุมการวิ่งของรถและจำนวนเที่ยวรถเป็นไปอย่างมีระบบ ลดจำนวนเที่ยวรถที่ไม่จำเป็น และการจัดเส้นทางเดินรถ มค่าที่สุด ซอฟต์แวร์ประเภทนี้ช่วยงานใน 2 ด้านหลักคือ

2.1 การวางแผนระดับกลยุทธ์ ซอฟต์แวร์ประเภทนี้ถูกนำมาใช้เพื่อการวางแผนและออกแบบโครงการ การคำนวณจำนวนรถบรรทุกที่จำเป็นสำหรับโครงการนั้นๆ การวางเส้นทางและกำหนดการเดินรถ และการวิเคราะห์ต้นทุนค่าขนส่ง ตัวอย่างซอฟต์แวร์ประเภทนี้ได้แก่ Paragon, CAST และ DPM เป็นต้น ซึ่งมักจะถูกนำมาใช้งานในช่วงต้นของการเริ่มโครงการ หรือเมื่อมีการเปลี่ยนแปลงธุรกิจเกิดขึ้น โดยเฉพาะผู้ให้บริการประเภท 3PL (Third-parties Logistic Provider) มักจะนำมาใช้เป็นเครื่องมือสำหรับฝ่ายพัฒนาธุรกิจในการช่วยวางแผนก่อนการนำเสนอโครงการแก่ลูกค้า เพื่อช่วยให้เห็นโซลูชั่นที่นำเสนอเป็นรูปธรรมมากขึ้น

2.2 การวางแผนและอำนวยความสะดวกด้านการปฏิบัติการ ซอฟต์แวร์ประเภทนี้ถูกนำมาประยุกต์ใช้เพื่อช่วยอำนวยความสะดวกในระดับปฏิบัติการ เช่น การวางแผนเส้นทางการเดินรถและกำหนดการเดินรถ การกำหนดจุดแวะรับสินค้า หรือการส่งสิ้นค้าตามรายทาง เช่นระบบ “Milk Runs” ซึ่งมีการหยุดแวะรับสินค้า หรือการลงสินค้าตามรายทางก่อนนำสินค้าป้อนเข้าโรงงาน นอกจากนี้ระบบนี้ยังเอื้อต่อการทำงานร่วมกับระบบตรวจสอบหาตำแหน่งและควบคุมการเดินรถ ซึ่งจะได้กล่าวต่อไป ซอฟต์แวร์ประเภทนี้ได้แก่ Universe Land Transport Management System (uLTMS) และ Fleet Management System (FMS) เป็นต้น

3. ระบบตรวจหาตำแหน่งและควบคุมการเดินรถ (Vehicle Based System)

ระบบนี้ครั้งหนึ่งเคยอาศัยสัญญาณดาวเทียวมรการจับทิศทางของตำแหน่งรถในท้องถนน แต่ปัจจุบันเทคโนโลยี GPRS (General Packet Radio Service) เข้ามามีบทบาทและเครือข่ายเกือบจะครอบคลุม อีกทั้งง่ายต่อการนำมาประยุกต์ใช้อย่างกว้างขวาง ซึ่งนอกจากจะควบคุมคนขับรถและตัวรถที่วิ่งอยู่บนท้องถนนแล้ว ยังสามารถนำมาใช้ประโยชน์อีกหลายๆ ด้านเช่น

- ควบคุมดูแลการใช้น้ำมัน
- การล๊อคตู้คอนเทนเนอร์ด้วยระบบอิเล็กทรอนิกส์ เพื่อป้องกันการถูกเปิดระหว่างทาง
- ตรวจสอบความผิดปรกติของตัวรถและประสิทธิภาพของรถไปในตัว
- ควบคุมพฤติกรรมคนขับรถ
- ดูแลรถให้อยู่บนเส้นทางที่ควรจะเป็นและสภาพของท้องถนน
- สามารถรู้ความคืบหน้าของเส้นทางการเดินรถและตรวจสอบได้

- หากมีการติดกล่องวัดอุณหภูมิไว้ในห้องบรรทุก สามารถควบคุมอุณหภูมิ
- การดักฟังการสนทนาหากมีสิ่งบอกเหตุผิดปรกติ

2.9.3 ข้อเท็จจริงว่าด้วยระบบจัดการงานขนส่ง

ระบบจัดการงานขนส่งในท้องตลาดยังเรียกแตกต่างกันออกไป องค์กรที่ให้บริการด้าน การตรวจสอบสถานะสินค้าหรือรถขนส่งด้วยระบบ GPS (Global Positioning System) เรียก บริการของตนเองว่า Fleet Management System ในขณะที่องค์กรที่สนใจเรื่องการจับคู่ความ ต้องการขนส่งสินค้ากับรถวิ่งเที่ยวเปล่าในเส้นทางที่สอดค องกันเพื่อหาโอกาสในการใช้รถวิ่ง เที่ยวเปล่า ซึ่งคิดค่าขนส่งที่ต่ำกว่ามาก (เช่น Thaitruckcenter.com) เรียกบริการของตนซึ่งเป็น บริการเว็บเซอร์วิสนี้ว่า Transport Management System หรือ Fleet Management System ส่วน ซอฟต์แวร์พื้นฐานที่ผู้ประกอบการขนส่งหรือผู้ให้บริการขนส่งจะใช้เป็นเครื่องมือในการบริหาร จัดการขนส่งโดยเฉพาะ มีหน้าที่หลักสำคัญคือ การจัดสรรรถบรรทุก หรือ Fleet Optimization และการจัดเส้นทางเดินรถหรือ Route Optimization มีชื่อเรียกโดยรวมว่า TMS (Transport Management System)

TMS คือซอฟต์แวร์สำหรับแก้ไขปัญหาบริการขนส่งที่มีความซับซ้อนสูง เป็นเครื่องมือที่ ผู้ประกอบการขนส่งมืออาชีพเริ่มให้ความสนใจศึกษาพัฒนามาใช้ ในการบริหารจัดการ ฐานข้อมูล สำคัญใน TMS ประกอบด้วย

1. เส้นทางการเดินรถ
2. กองรถ
3. พนักงานขับรถ
4. ข้อจำกัดด้านกฎหมาย
5. จุดหลักหรือสถานที่ที่รถต้องแวะรับและส่งสินค้า
6. ระบบรับคำสั่งจากลูกค้า (ประเภทและจำนวนสินค้า ต้นทางปลายทางการ ขนส่ง เวลานัดหมาย และบริการเพิ่มเติมอื่นๆ)

โปรแกรม TMS ดังกล่าวอาจจะประกอบด้วยโปรแกรมย่อยหลายตัว เช่น โปรแกรมการ จัดสรรรถให้ได้ประโยชน์สูงสุด (Fleet Optimization) ที่เรียกโดยรวมว่าโปรแกรมการจัดสรร เส้นทางที่สั้นที่สุด (Shortest Path) หรือที่เร็วที่สุด (Shortest Time) หรือการจัดเส้นทางให้ได้ ประโยชน์สูงสุด (Route Optimization) โปรแกรมเหล่านี้หาซื้อลิขสิทธิ์ได้ง่ายจากต่างประเทศ แต่ ราคาค่อนข้างสูง โอกาสที่ผู้ประกอบการจะลงทุน ทุกรายก็คงเป็นไปไม่ได้ การพัฒนา โปรแกรมเหล่านี้เป็นโอเพ่นซอส(Open-Source)จึงเป็นโอกาสที่ดีกว่าในระยะยาว ประเทศไทยมี ผลงานที่เกี่ยวข้องออกมาบ้างแล้ว หากจะมีการพัฒนาให้เป็นเครื่องมือที่สามารถใช้กันได้อย่าง

แพร่หลาย ก็จะเป็นประโยชน์อย่างยิ่งกับผู้ประกอบการข าดเล็กถึงกลางในยุคปัจจุบันที่มีการแข่งขันสูง ซึ่งหากไม่บริหารประสิทธิภาพให้ดีคงยากที่จะอยู่รอดได้

2.4 GPS Tracking System

(วิศิษฏ์ วัฒนานุกูล, 2552)

GPS หรือ Global Positioning System คือระบบบอกพิกัดผ่านดาวเทียม ซึ่งโคจรสูงจากพื้นโลกประมาณ 20,000 กิโลเมตร โดยทำหน้าที่ส่งสัญญาณให้กับอุปกรณ์ลูกข่ายเพื่อคำนวณ ตรวจสอบ และถอดรหัสสัญญาณที่ได้จากดาวเทียม เพื่อให้ได้พิกัดตำแหน่งและข้อมูลการเคลื่อนที่ของยานพาหนะที่ถูกต้องตลอด 24 ชั่วโมง ดังนั้นการนำระบบ GPS Tracking System มาใช้เพื่อช่วยสนับสนุนการบริหารการใช้ยานพาหนะได้เต็มประสิทธิภาพสามารถนำข้อมูลที่ได้รับมาใช้ในการวิเคราะห์ต้นทุนการใช้ยานพาหนะตามจริงได้ จึงช่วยลดค่าใช้จ่ายจากการใช้ยานพาหนะที่ไม่จำเป็นหรือผิดวัตถุประสงค์ รวมทั้งลดการสูญเสียและติดตามแก้ไขปัญหาอุบัติเหตุหรือปัญหาอื่นๆ ที่อาจจะเกิดขึ้นระหว่างการขนส่ง หรือใช้ยานพาหนะได้อย่างทันท่วงที ฉะนั้นการทำงานของ GPS จึงสามารถตอบสนองความต้องการของลูกค้าได้อย่างมากมาย เช่น

- ตรวจสอบตำแหน่งปัจจุบันของรถ และแสดงพฤติกรรมของการใช้งานรถ
- ตรวจสอบสถานะปัจจุบันของรถ เช่น จอดรถ ติดเครื่อง-ดับเครื่อง หรือขับรถเร็วเกินกำหนด
- แสดงเส้นทางการเดินรถย้อนหลังในแต่ละวัน และแสดงเวลาเมื่อรถผ่านสถานที่ต่างๆ
- แสดงเวลาและสถานที่ที่มีการเริ่มต้นใช้งานรถ ขับรถเร็ว หรือจอดรถดับเครื่องจอดรถติดเครื่องทิ้งไว้ เข้าสถานีการเข้าไปในพื้นที่หวงห้าม
- สามารถกำหนดตำแหน่งสถานที่สำคัญในแผนที่ และบันทึกเวลาการถึงที่หมายในแต่ละวันได้
- สามารถบริหารเวลาการทำงานของรถ ทำให้ใช้งานรถได้เต็มประสิทธิภาพ

2.4.1 ประโยชน์ของ GPS Tracking System

- ลดต้นทุนค่าใช้จ่ายจากการเดินรถ โดยการประหยัดค่าน้ำมัน และลดค่าใช้จ่ายจากการซ่อมบำรุง อันเนื่องมาจากการออกนอกเส้นทาง การติดเครื่องยนต์ทิ้งไว้ การขับรถเร็วซึ่งอาจทำให้เกิดอุบัติเหตุ รวมถึงสามารถตรวจสอบในเรื่องการลักลอบดูดน้ำมันไปขายของพนักงานขับรถ

- ป้องกันการนำรถไปใช้ผิดวัตถุประสงค์ ลดพฤติกรรมการใช้งานที่ไม่เหมาะสม เช่น การหยุดพักที่นานเกินควร หรือจอดรถโดยติดเครื่องเป็นระยะเวลานาน
- เพิ่มความปลอดภัยในทรัพย์สิน และช่วยลดการเกิดอุบัติเหตุจากการใช้ความเร็วที่ไม่เหมาะสม
- บริหารเวลาการทำงานของรถได้ดียิ่งขึ้น ทำให้ใช้งานรถได้เต็มประสิทธิภาพ และเพิ่มประสิทธิภาพในการเดินรถให้เกิดประโยชน์สูงสุด
- สามารถตรวจสอบพฤติกรรมของคนขับรถแต่ละคน จึงสะดวกต่อการควบคุมดูแลและขอความร่วมมือจากพนักงานขับรถ
- เพิ่มคุณภาพในการบริการลูกค้า และการแข่งขันทางธุรกิจ

2.5 ห่วงโซ่อุปทาน (Supply Chain)

(INTERTRANSPORT LOGISTICS ปีที่ 3 ฉบับที่ 63, 2546 : 4)

ความหมายของซัพพลายเชน การจัดการซัพพลายเชน หรือการจัดการห่วงโซ่อุปทาน เป็นการจัดลำดับของกระบวนการทั้งหมดที่มีต่อการสร้างความพอใจให้กับลูกค้า โดยเริ่มต้นตั้งแต่กระบวนการจัดซื้อ (Procurement) การผลิต (Manufacturing) การจัดเก็บ (Storage) เทคโนโลยีสารสนเทศ (Information Technology) การจัดจำหน่าย (Distribution) และการขนส่ง (Transportation) ซึ่งกระบวนการทั้งหมดนี้จะจัดระบบให้ประสานกันอย่างคล่องตัว

นอกจากนี้ การจัดการซัพพลายเชนไม่ได้ครอบคลุมเฉพาะหน่วยงานต่าง ๆ ภายในองค์กรเท่านั้น แต่ที่สำคัญจะสร้างความสัมพันธ์เชื่อมต่อกับองค์กรอื่น ๆ อย่างมีประสิทธิภาพ เช่น ผู้จัดหาวัตถุดิบ/สินค้า (Suppliers) บริษัทผู้ผลิต (Manufactures) บริษัทผู้จำหน่าย (Distribution) รวมถึงลูกค้าของบริษัท จึงเป็นการเชื่อมโยงกระบวนการดำเนินธุรกิจทุกขั้นตอนที่เกี่ยวข้องด้วยกันเป็นห่วงโซ่หรือเครือข่ายให้เกิดการประสานงานกันอย่างต่อเนื่อง ทั้งนี้เพื่อเพิ่มประสิทธิภาพในการผลิตสินค้า/บริการ สร้างความพึงพอใจแก่ลูกค้า แต่ละหน่วยงานจึงมีความเกี่ยวเนื่องกันเหมือนห่วงโซ่

ในห่วงโซ่อุปทานนั้นข้อมูลต่างๆ จะมีการแชร์หรือแจ้งและแบ่งสรรให้ทุกแผนก/ทุกหน่วยงานในระบบรับทราบและใช้งาน ทำให้หน่วยงานแต่ละหน่วยงานสามารถทำงานได้อย่างมีประสิทธิภาพ ยกตัวอย่างเช่น ในการประกอบรถยนต์หนึ่งคัน แผนกจัดซื้อจะจัดซื้อวัตถุดิบหลายอย่างไม่ว่าจะเป็นเครื่องยนต์ น้ำมันเครื่อง แบตเตอรี่ ยางรถยนต์ เป็นต้น
เมื่อสั่งซื้อเสร็จอุปกรณ์ดังกล่าวจะเก็บให้ในคลังสินค้า เพื่อรอฝ่ายการผลิตรถยนต์นำไปผลิตรถยนต์ตามที่ต้องการ และถ้าองค์กรนี้มีระบบการจัดการซัพพลายเชนที่ดี แผนกต่าง ๆ มีการแชร์หรือแลกเปลี่ยนข้อมูลกันจะทำให้การสั่งซื้อวัตถุดิบเป็นไปด้วยความถูกต้องและเป็นระบบ

2.6 ระบบขนส่งอัจฉริยะ (Intelligent Transport System หรือ ITS)

(สรวิศ นฤปิติ, ไม่ปรากฏปีที่พิมพ์)

ระบบขนส่งอัจฉริยะ เป็นการพัฒนาระบบขนส่งและการจราจร (ทางบก) ด้วยการเลือกสรร พัฒนา และ ประยุกต์ใช้ระบบเทคโนโลยีต่างๆ โดยเฉพาะเทคโนโลยีที่ก้าวหน้าของคอมพิวเตอร์ สารสนเทศ และ อิเลกทรอนิคส์ ซึ่งการประยุกต์ใช้เทคโนโลยีต่าง ๆ ไม่ใช่จะทำให้ระบบขนส่ง และการใช้งานทันสมัยมากยิ่งขึ้นเท่านั้น แต่จะเป็นการนำเทคโนโลยีที่เหมาะสมมาเพิ่มสมรรถนะของระบบการจราจรและขนส่งลดปัญหาการจราจรที่เป็นอยู่เพิ่มความปลอดภัยของการสัญจร และ ลดมลภาวะอันเกิดจากการขนส่ง นอกจากนี้ระบบขนส่งอัจฉริยะจะก่อให้เกิผลประโยชน์ทางเศรษฐกิจในการขนส่ง เพิ่มคุณภาพชีวิตในการใช้รถใช้ถนน เสริมสร้างความเป็นอยู่ที่ดี ได้รับความสะดวกสบายในการเดินทางและขนส่งมากยิ่งขึ้น

2.6.1 GIS and ITS

GIS ถูกนำมาใช้ในงานระบบขนส่งอัจฉริยะอย่างมากเพื่อเพิ่มประสิทธิภาพในการพัฒนาเทคโนโลยีเพื่องานเฉพาะหน้าที่ อย่างไรก็ดี GIS อาจถูกมองว่าเป็นส่วนหนึ่งของระบบITSได้เพราะGISเองจะเพิ่มความ“อัจฉริยะ”ในกิจกรรมที่เกี่ยวข้องกับการเดินทาง/ขนส่ง ทำให้กิจกรรมขนส่งเกิดความสะดวกสบาย คล่องตัว และ ปลอดภัย
มากยิ่งขึ้น GIS ถูกนำมาใช้ประกอบกับงาน ITS ดังต่อไปนี้

- งานสารสนเทศการเดินทางแบบก้าวหน้า
- งานวางแผนการเดินทางแบบก้าวหน้า
- งานจัดการจราจรแบบก้าวหน้า
- งานจัดการที่จอดรถแบบก้าวหน้า
- งานการติดตามยานพาหนะ
- งานระบบขนส่งสาธารณะแบบก้าวหน้า
- งานระบบการจัดการเดินรถขนส่งเชิงพาณิชย์

2.6.2 GIS ในงานการติดตามยานพาหนะ

GIS เป็นองค์ประกอบที่สำคัญในการติดตามยานพาหนะหรือ บุคคล (Vehicle or Personal tracking) ปัจจุบันเทคโนโลยีการติดตามตำแหน่งของยานพาหนะ ถูกพัฒนาไปอย่างรวดเร็วทั้งระบบ Wireless และ GPS (Global Positioning System) โดยเฉพาะระบบ GPS มีใช้อย่างแพร่หลายในอุปกรณ์ส่วนบุคคล เช่น โทรศัพท์พกพา เป็นต้น

GIS และ GPS กำลังเป็นที่นิยมในการประยุกต์ใช้สำหรับงานระบบขนส่งอัจฉริยะ โดยใช้ในงานหลากหลายตั้งแต่ การรวบรวมข้อมูลการจราจรด้วย Probe Car การตรวจสอบตำแหน่ง

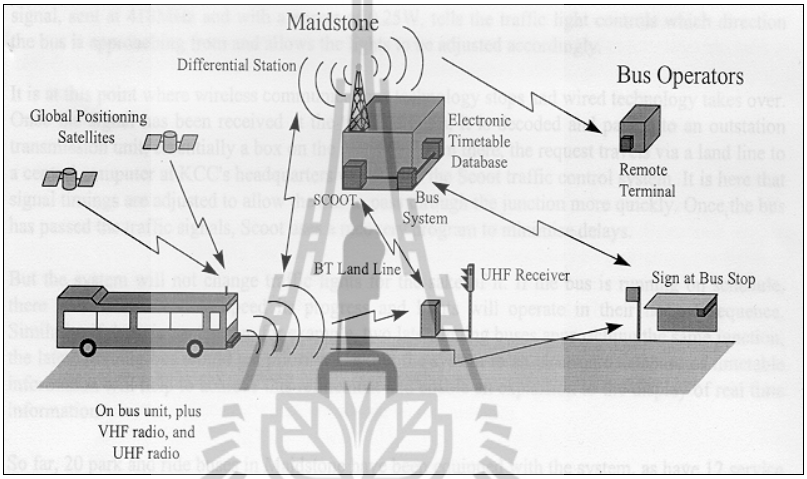
ยานพาหนะ เพื่อการนำทาง (Navigation)และ แนะนำเส้นทาง (Route Guidance) การติดตามตำแหน่งรถขนส่งสาธารณะ เพื่อการจัดการเดินรถ และสารสนเทศการเดินทาง จนถึง การติดตามตำแหน่งรถขนส่งเชิงพาณิชย์ เพื่อการจัดการเดินรถขนส่งที่มีประสิทธิภาพ

การรวบรวมข้อมูลการจราจร ด้วย Probe Car เป็นการรวบรวมข้อมูลการจราจร จากยานพาหนะแต่ละคันที่ใช้ระบบถนน ลักษณะการเคลื่อนที่ของยานพาหนะแต่ละคัน รวมถึงข้อมูลประจำรถ (เช่น การเปิดไฟหน้า การใช้ใบปัดน้ำฝน อุณหภูมิรอบรถ เป็นต้น) เมื่อนำมารวบรวมและวิเคราะห์ จะได้ข้อมูลการจราจรที่สำคัญที่ประสบจริงโดยผู้ขับขี่ที่ใช้ระบบถนน ณ ตำแหน่งต่างๆ การรวบรวมข้อมูลจราจรดังกล่าว กระทำได้โดยง่ายผ่าน GIS และ นำมาจัดเป็นสถิติการจราจร ณ ตำแหน่งต่างๆ บนโครงข่ายถนนเพื่อใช้ในการจัดทำสารสนเทศเพื่อการเดินทาง และ การจัดการจราจรที่ เหมาะสม

รูปที่ 2.13 แผนที่แสดงเส้นทางภายในรถช่วยแนะนำการเดินทางแก่ผู้ขับขี่

การตรวจสอบตำแหน่งยานพาหนะ เพื่อการนำทาง(Navigation) และ แนะนำเส้นทาง (Route Guidance) เป็นการตรวจสอบตำแหน่งของยานพาหนะคันใด ๆ เพื่อระบุตำแหน่งบนแผนที่ระบบ GIS ที่จะใช้เป็นเครื่องมือในการแสดงผลเพื่อให้ข้อมูลแก่ผู้ขับขี่ในการขับขี่ยานพาหนะไปยังจุดหมายที่ต้องการ GIS ยังทำหน้าที่เป็นเครื่องมือในการวิเคราะห์หารายละเอียดที่ผู้ขับขี่ต้องการ เช่น ระยะห่างจากทางแยกที่ต้องการเลี้ยวข้อมูลสถานที่ระหว่างเส้นทาง การคำนวณระยะทางและความเร็ว นอกจากนี้ ยังอาจมีการติดต่อแลกเปลี่ยนข้อมูลอื่นๆ จากยานพาหนะอื่นๆ (หรือ ข้อมูลการจราจรโดยรวมทันกาล) เพื่อใช้ในการวิเคราะห์หาเส้นทางที่เหมาะสม ข้อมูลเพิ่มเติมที่จะเป็นในการวิเคราะห์อาทิเช่น สภาพการจราจร (ปริมาณการจราจร เวลาในการเดินทาง)บนช่วงถนนต่างๆ อุบัติการณ์ (Incident)หรือ กิจกรรมพิเศษบนโครงข่ายถนน ณ เวลาต่างๆ โดยGIS จะช่วยให้การรวบรวม และ วิเคราะห์ข้อมูลเป็นไปโดยดี

การติดตามตำแหน่งรถขนส่งสาธารณะ เพื่อการจัดการเดินรถ และ สารสนเทศการเดินทาง GIS ใช้ในการกำหนดตำแหน่งรถประจำทางรถรางไฟฟ้า และ รถขนส่งสาธารณะอื่น ที่ติดตั้งอุปกรณ์ติดตามยานพาหนะ (Automatic Vehicle Locator) โดยเป็นเครื่องมือในการจัดเก็บข้อมูล และ วิเคราะห์ตำแหน่ง เพื่อนำเสนอ ตำแหน่ง เวลาที่รถจะถึงป้าย หรือที่หมายต่างๆ นอกจากนี้ตำแหน่งของรถแต่ละคันยังใช้ในการจัดการเดินรถอย่างมีประสิทธิภาพ เช่น การให้สิทธิ์แก่รถประจำทาง ณ ทางแยก การจัดรถเสริม หรือ การเปลี่ยนแปลงเส้นทางเดินรถแบบทันกาลเพื่อตอบสนองต่อสภาพการจราจรปัจจุบัน



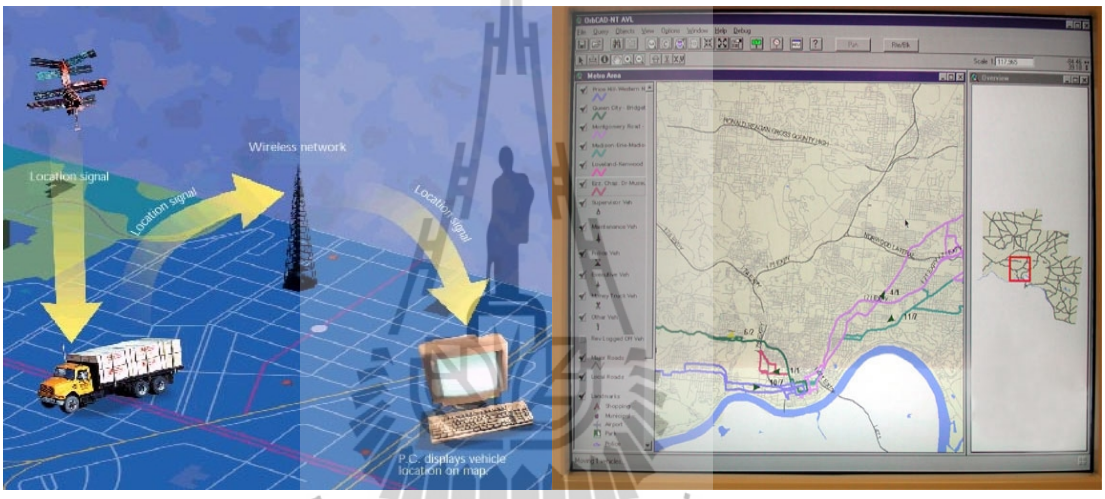
รูปที่ 2.14 GIS ใช้ร่วมกับ GPS ในการติดตามรถประจำทาง และ รถฉุกเฉิน

การติดตามตำแหน่งรถขนส่งเชิงพาณิชย์ GIS ถูกนำมาใช้ในรถขนส่งเชิงพาณิชย์ในลักษณะคล้ายกันกับรถขนส่งสาธารณะ เพื่อช่วยในการตรวจสอบการเดินทางและเส้นทางที่รถขนส่ง (เช่น รถบรรทุก)เดินทาง ช่วยในการจัดการสินค้าได้อย่างถูกต้อง รวดเร็วแม่นยำ และตอบสนองต่อการจัดการเพื่อสามารถบริการจัดส่งได้ตรงตามต้อ นอกจากนี้ยังมีส่วนสนับสนุนในความปลอดภัยของสินค้าสูญหาย และเฝ้าติดตามความปลอดภัยสาธารณะ ในกรณีของการขนส่งสินค้า/วัตถุอันตราย

การพัฒนาระบบขนส่งสาธารณะในกรอบของระบบขนส่งอัจฉริยะนั้นครอบคลุมการเพิ่มประสิทธิภาพของการให้บริการระบบขนส่งสาธารณะโดยรวม โดยการเพิ่มประสิทธิภาพนั้นเกิดขึ้นทั้งในการจัดการระบบขนส่งและการเดินรถ และการใช้บริการกับผู้ใช้ที่มีคุณภาพดีขึ้น ระบบ GIS ถูกนำมาใช้ในงานระบบขนส่งสาธารณะ ทั้งในการวางแผนเส้นทาง และการจัดการออกแบบการเดินรถ นอกจากนี้ด้วยข้อมูลที่ทันการณ์ ผู้จัดการเดินรถสามารถใช้ GIS ในการ

ตรวจสอบวิเคราะห์ การเดินรถ ณ เวลาใดๆ ด้วยการติดตามตำแหน่งรถสาธารณะดังที่นำเสนอใน

หัวข้อก่อนหน้านี้ ช่วยให้การตัดสินใจปรับเปลี่ยนการเดินรถได้อย่างมีระบบ ระบบ GISถูกมาใช้ไม่เพียงแต่ระบบขนส่งสาธารณะ ประเภทสายทางแน่นอนเท่านั้น แต่ยังนำมาใช้อย่างมากกับระบบขนส่งกึ่งสาธารณะ (Paratransit) เช่น Demand-Responsive Transport System และระบบรถรับจ้างสาธารณะ (i.e.รถแท็กซี่) GIS ใช้ในการรวบรวมข้อมูลความต้องการเดินทางของผู้เดินทาง ทำให้สามารถกำหนดตำแหน่งของผู้เดินทางและ สามารถจัดรถรับส่งได้สอดคล้องกับความต้องการได้โดยปัจจุบันมีการพัฒนาโปรแกรมเพื่อวิเคราะห์การเดินรถที่มีประสิทธิภาพเพื่อตอบสนองต่อความต้องการเดินทางของผู้ใช้ที่แสดงความต้องการแบบทันการณ์



รูปที่ 2.15 การติดตามตำแหน่งรถบรรทุกและแสดงผลบน GIS

2.7 การพัฒนาเฟรมเวิร์ค (Framework Development)

โดยทั่วๆ ไปแล้ว เราสามารถให้ความหมายของเฟรมเวิร์ค ว่าเป็นโครงสร้างที่รวมความเข้าใจในกรอบงานนั้นโดยทั่วไป ซึ่งสามารถนำมาเพิ่มรายละเอียดเพิ่มเติมเพื่อให้กลายเป็นโปรแกรมประยุกต์ที่สมบูรณ์ได้(Anderson J, 1983)

Framework เป็นกรอบการทำงาน โดยมีการระบุโครงสร้างการทำงานพร้อมกับรายละเอียดส่วนอื่นๆ ประกอบกันทำให้เกิดผล ซึ่งเป็นตัวประสานการทำงานระหว่างตัวโปรแกรมกับระบบ โดย

Framework เป็นที่นิยมใช้มากในวิศวกรรมซอฟต์แวร์ โดยเฉพาะอย่างยิ่งที่เกี่ยวข้องกับการวิเคราะห์และออกแบบเชิงวัตถุ ซึ่ง Object Oriented Framework ตัวแรกที่ได้ถูกพัฒนาขึ้นคือ

MVC(Model View Controller) สำหรับ Small talk และ MacApp สำหรับ Apple Application เป็นต้น

สำหรับนิยมที่ได้รับการยอมรับของ Object Oriented Framework คือ “a framework is a set of classes that embodies an abstract design for solutions to a family of problems” (Johnson R. E. and Foote J., 1988 : 22-35)

เฟรมเวิร์ค คือกลุ่มของ Class ที่ประกอบด้วย การออกแบบที่เป็นเชิงนามธรรมเพื่อแนวทางแก้ปัญหาของกลุ่มปัญหาหนึ่งๆ ซึ่งสรุปได้ว่า Framework คือ เป็นการออกแบบระบบหรือส่วนประกอบของระบบเพื่อให้สามารถนำกลับมาใช้ซ้ำซึ่งจะรวบรวม abstract Class และแนวทางในการสร้างอินสแตนท์ของ class ซึ่งโปรแกรมเมอร์สามารถใช้ สืบทอด หรือปรับเปลี่ยนเพื่อการใช้งานที่เฉพาะเจาะจงตามลักษณะปัญหา ซึ่งเฟรมเวิร์ค ก็อาจจะเปรียบเสมือนโปรแกรมประยุกต์เชิงนามธรรม (Abstract Application) ขนาดใหญ่ในขอบเขต (Domain) ที่เฉพาะเจาะจงซึ่งสามารถปรับปรุงต่อยอดเป็นโปรแกรมประยุกต์เฉพาะทางได้ เฟรมเวิร์คเป็นสถาปัตยกรรมซอฟต์แวร์ที่สามารถถูกใช้ซ้ำได้ซึ่งประกอบด้วยทั้งการออกแบบแบบและโค๊ดโปรแกรม(Beck K. and Johnson R., 1994)

2.7.1 การจำแนกประเภทของเฟรมเวิร์ค

Application Framework จะเน้นไปในด้านการพัฒนาซอฟต์แวร์เฉพาะด้านที่เฉพาะเจาะจงหรือส่วนประกอบของโปรแกรมประยุกต์

Domain framework หรือ Support Framework จะเน้นไปในทาง System level Service เช่น file access หรือ Distribute Computing (Taligent, 1994)

มหาวิทยาลัยเทคโนโลยีสุรนารี

Framework สามารถแบ่งออกได้เป็น 3 ระดับดังนี้

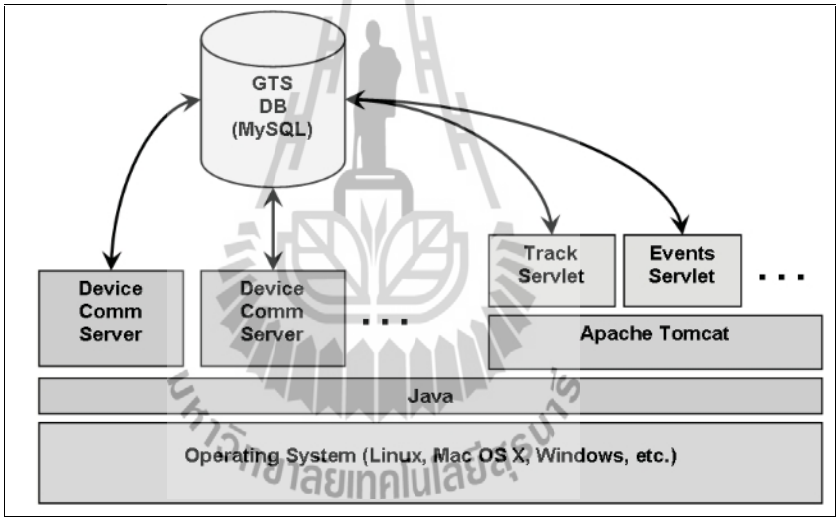
1. System Infrastructure framework เป็นเฟรมเวิร์ค ที่สนับสนุนการพัฒนาโครงสร้างพื้นฐานของระบบ เช่น การติดต่อสื่อสาร ส่วนประสานกับผู้ใช้ (User Interface) และคอมไพเลอร์(Compiler)
2. Middleware Integration Framework เป็นเฟรมเวิร์ค ที่ประกอบด้วยกลุ่มของ object และ Class เพื่อสนับสนุนการทำงานของ component ในการติดต่อสื่อสารและสนับสนุนการแลกเปลี่ยนข้อมูล เช่น Web Service , .NET และ Enterprise Java Bean
3. Enterprise Application Framework เป็นเฟรมเวิร์ค ที่ใช้เป็นกรอบการพัฒนาระบบสำหรับองค์กรขนาดใหญ่ ที่การทำงานมีความซับซ้อน และต้องรวบรวมระบบย่อย

หลายระบบเข้าด้วยกัน เช่น งานด้านโทรคมนาคม ระบบเกี่ยวกับการเงินและการคลังเป็นต้น

การพัฒนาระบบด้วยเทคนิคเฟรมเวิร์ค จะช่วยสนับสนุนการนำกลับมาใช้ใหม่ทำได้ง่ายขึ้น เนื่องจากสามารถปรับปรุงและสร้างเฟรมเวิร์ค ใหม่ได้อย่างสะดวกไม่ส่งผลกระทบต่อระบบโดยรวม การขยายขอบเขตการทำงานของเฟรมเวิร์คทำได้ไม่ยาก โดยการเพิ่มส่วนประกอบหรือ Class ให้กับเฟรมเวิร์ค และยังสามารถเพิ่มประสิทธิภาพด้วยการนำแบบแผน (pattern) เข้ามารวมไว้ในเฟรมเวิร์คได้อีกด้วย (Timothy C. Lethbridge and Robert Laganiere, 2005)

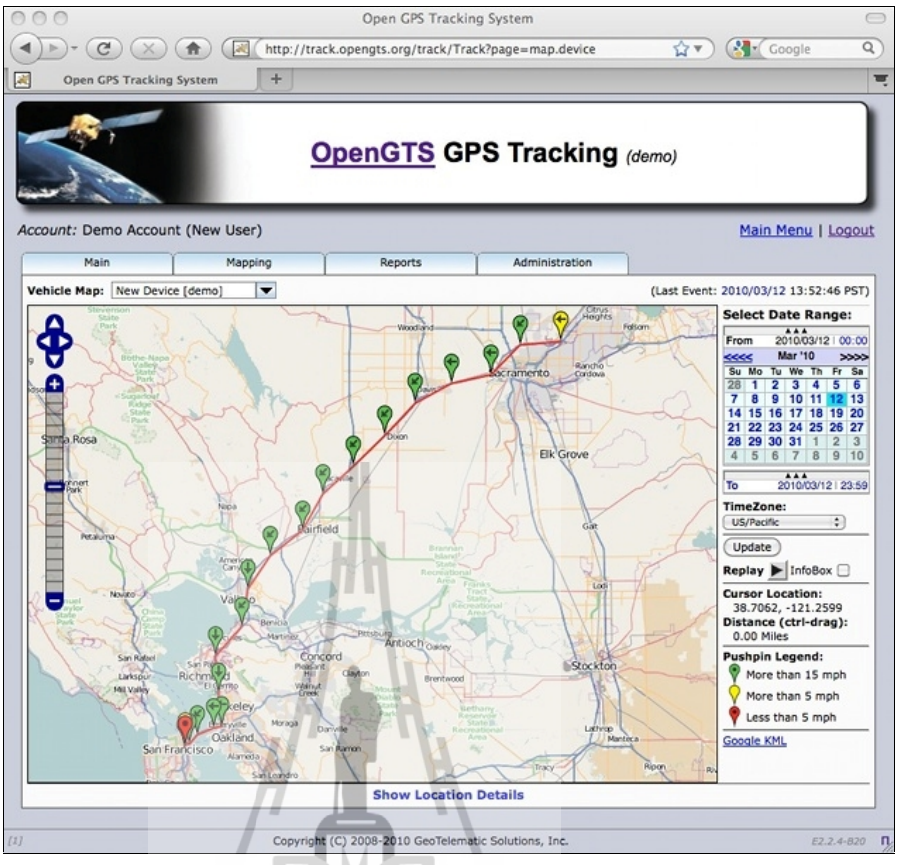
2.8 OpenGTS (Open GPS Tracking System)

(K. Hausknecht, 2011:606 – 611)



รูปที่ 2.16 OpenGTS Architecture

OpenGTS (Open GPS Tracking System) เป็นโครงการโอเพ่นซอสโครงการแรกๆ ที่ออกแบบมาเฉพาะเพื่อให้บริการติดตามด้วย GPS ในรูปแบบของบริการผ่านเว็บไซต์เพื่อที่จะติดตามยานพาหนะหรือทรัพย์สิน ซึ่งสามารถติดตามยานพาหนะและสินทรัพย์หลายประเภ รวมทั้งแทกซี่, รถตู้ขนส่ง, รถบรรทุกสินค้า เป็นต้น



รูปที่ 2.17 OpenGTS Web UI

ซึ่งระบบ OpenGTS ได้ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อรองรับความต้องการเบื้องต้นเกี่ยวกับ Web based ซึ่งจะสามารถตอบสนองความต้องการพื้นฐานของธุรกิจการขนส่ง แต่ OpenGTS สามารถปรับแต่งได้ง่ายและมีความสามารถในการต่อขยาย (scalable) เพื่อไปสู่ธุรกิจระดับใหญ่ขึ้นได้เช่นกัน

OpenGTS ไม่เพียงแต่รองรับการรับและเก็บรักษาข้อมูลการติดตามด้วยดาวเทียมและข้อมูลด้านการสื่อสารจากอุปกรณ์ระยะไกลเท่านั้น แต่ยังประกอบด้วยความสามารถที่หลากหลายอีกมากมาย อาทิเช่น

- ความสามารถในการตรวจสอบการใช้งาน(Web base authentication) ซึ่งแต่ละบัญชีรองรับได้หลายผู้ใช้งาน และผู้ใช้งานแต่ละคนสามารถกำหนดชื่อผู้ใช้งานและรหัสผ่านเองได้และสามารถควบคุมการใช้งานได้ผ่านบัญชี
- สามารถเลือกและปรับแต่งรูปแบบส่วนประกอบของการแสดงผลหน้าเว็บเองได้

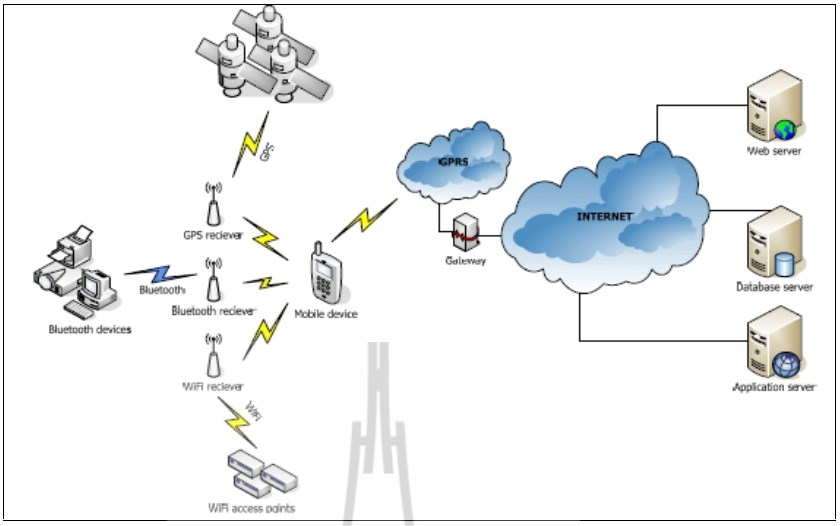
- สามารถที่จะเลือกบริการแผนที่เองได้ ซึ่ง opengts มาพร้อมกับความสามารถในการรองรับระบบแผนที่ที่หลากหลาย อาทิเช่น Openlayers/OpenstreetMap , Google Maps, Microsoft virtual earth และด้วยความสามารถของ OpenGTS Framework ก็สามารถประยุกต์ให้สามารถรองรับระบบแผนที่อื่นๆ ได้ไม่ยากนัก
- สามารถแสดงผลรายงานในรูปแบบของ XML
- สามารถกำหนดพื้นที่ทางด้านภูมิศาสตร์เองได้ (GeoZone) ซึ่งสามารถตั้งค่าให้ทำการแจ้งเตือนอัตโนมัติเมื่อมาที่บริเวณที่กำหนดได้ ซึ่งอาจจำเป็นที่จะต้องทำการเขียนโค๊ดเพิ่มเติม
- OpenGTS ไม่ขึ้นกับอุปกรณ์ติดตามด้วยดาวเทียมใดๆ โดยเฉพาะ ระบบ OpenGTS มาพร้อมกับความสามารถในการรองรับอุปกรณ์ได้หลากหลายยี่ห้อ Sanav, ICare, SIPGEAR และอุปกรณ์ที่รองรับมาตรฐาน OpenDMTP แต่ก็สะดวกต่อการทำงานร่วมกับอุปกรณ์ประเภทต่างๆ ซึ่งมีความใกล้เคียงกัน ซึ่งเป็นผลให้สามารถใช้ Web Interface ชุดเดียวรับข้อมูลและติดตามอุปกรณ์ที่มีความแตกต่างกันได้
- สามารถรองรับการทำงานได้หลากหลายระบบปฏิบัติการ ด้วยความที่ตัวระพัฒนาขึ้นโดยภาษา JAVA และใช้เทคโนโลยีอาทิเช่น Apache Tomcat สำหรับการติดตั้งระบบเว็บ และ MySQL เป็นแหล่งข้อมูล ส่งผลให้ OpenGTS สามารถทำงานได้ในระบบปฏิบัติการทุกระบบที่รองรับเทคโนโลยีเหล่านี้
- i18n Compliant OpenGTS รองรับมาตรฐาน i18n เป็นผลให้สามารถรองรับได้หลากหลายภาษานอกเหนือจากภาษาอังกฤษ

2.9 A Framework for Developing Distributed Location Based Applications

(Andrej Krevl and Mojca Ciglari, 2006)

LBS (Location Based Service) เป็นที่พูดถึงมากในปัจจุบัน ซึ่งมีโปรแกรมประยุกต์ประเภทนี้อยู่หลากหลาย แต่สามารถเข้าถึงได้ยากเนื่องจากใช้ต้นทุนที่สูงในการเข้าถึงระบุตำแหน่ง งานวิจัยนี้ได้กล่าวถึงทางเลือกใหม่สำหรับการนำเอาตำแหน่งที่ตั้งของอุปกรณ์พกพาตัวอย่างเช่น โทรศัพท์มือถือหรืออุปกรณ์คอมพิวเตอร์พกพา ซึ่งสามารถอธิบายตำแหน่งที่ตั้งผ่านทางเครือข่ายไร้สาย, เครือข่าย GSM รวมทั้ง GPS นอกจากนี้ ยังนำเสนอเฟรมเวิร์คสำหรับพัฒนาระบบเชิงตำแหน่งที่ตั้งแบบกระจาย ซึ่งเอกสารฉบับนี้ระบุถึงส่วนประกอบของเฟรมเวิร์ค โครงสร้างของข้อมูลซึ่งจะใช้เป็นตัวกลางในการแลกเปลี่ยน และเว็บเซอร์วิสที่ใช้ในการติดต่อสื่อสารกันระหว่างส่วนต่างๆ พร้อมกันนั้นได้อธิบายถึงต้นแบบโปรแกรมประยุกต์ซึ่งพัฒนาขึ้นต่อยอดจากเฟรมเวิร์ค เป็นการแสดงถึงความเป็นไปได้และกล่าวถึงประโยชน์และข้อเสียของวิธีการนี้

2.9.1 สถาปัตยกรรมของเฟรมเวิร์ค



รูปที่ 2.18 แสดงสถาปัตยกรรมของเฟรมเวิร์ค

เฟรมเวิร์คนี้ประกอบด้วย 3 ชั้นหลักๆ

1. Mobile Device (location aware client)
2. Application Server (Broker)
3. Database Server

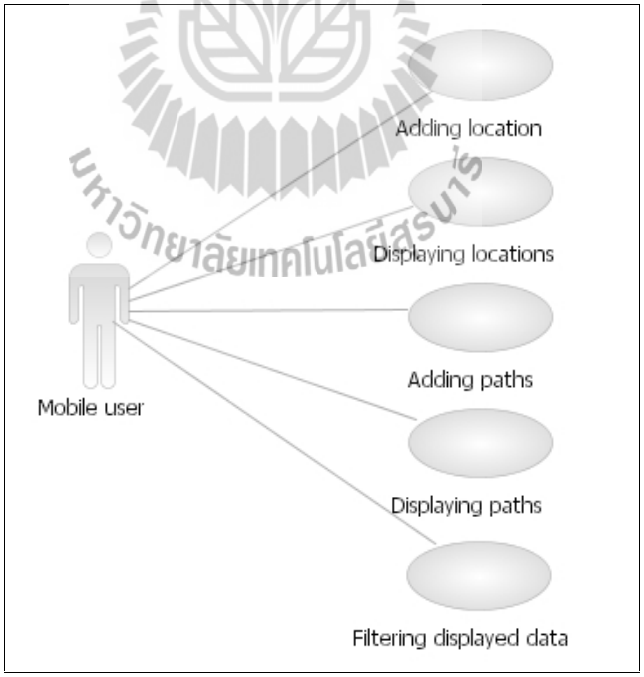
ซึ่งวิธีการนี้จะช่วยสดปัญหาการทำงานร่วมกันระหว่างส่วนประกอบที่แตกต่างกันคือ หมายถึง สามารถปรับเปลี่ยนคอมโพเน้นใหม่ได้ ตราบใดที่ยังคงความเจ้ากันได้ของ API เอาไว้ ในขณะเดียวกันการออกแบบให้มี 3 layer จะทำให้สามารถบรรจุกระบวนการทำงานและเก็บข้อมูลเชิงพื้นที่ไว้ที่ฝั่ง Server เพื่อลดภาระของอุปกรณ์พกพา

ข้อมูลเชิงพื้นที่ถูกเก็บไว้ในฐานข้อมูลเชิงสัมพันธ์บน Database server ซึ่งอาจจะทำให้ประสิทธิภาพลดลง แต่การทำงานร่วมกันและความแตกต่างด้านระบบการบริหารจัดการฐานข้อมูลนั้นมีความสำคัญมากกว่าในการพัฒนาระบบฐานข้อมูลเชิงภูมิศ ฐานข้อมูลเชิงพื้นที่ประกอบด้วยพิกัดตำแหน่งของเสาร์สัญญาณและพิกัดตำแหน่งของจุดที่น่าสนใจและเส้นทางน่าสนใจ เนื่องจากการออกแบบที่ไม่ซับซ้อน ทำให้สามารถปรับแต่งให้สามารถเก็บพิกัดตำแหน่งประเภทต่างๆ หรือ meta-data ได้ง่าย

Application Server ให้บริการเว็บเซอร์วิสสำหรับให้บริการแก่ mobile client และติดต่อสื่อสารกับ Database Server การติดต่อสื่อสารกับเครื่องลูกข่ายจะอยู่บนพื้นฐานของ SOAP

Message การสื่อสารภายในจะใช้เมื่อต้องการเรียกข้อมูลจาก Database Server ซึ่งก็เนื่องมาจากวัตถุประสงค์ของผู้วิจัยที่ต้องการใช้มาตรฐานกลาง ซึ่งก็ไม่ใช่ปัญหาใหญ่ใดๆ ที่จะตัดสินใจใช้เว็บเซอร์วิส การ native communication สามารถทำการแทนที่ได้ด้วย ODBC หรือ JDBC ซึ่งก็ขึ้นอยู่กับแพลตฟอร์มของเครื่องแม่ข่าย ซึ่งก็แนะนำให้ใช้การใช้ native database communication เพื่อประสิทธิ์ภาพสูงสุด

อุปกรณ์พกพาจะรับตำแหน่งภูมิศาสตร์ หรือรายการของเสาสัญญาณผ่านทางช่องทางไร้สาย ซื้อข้อมูลเหล่านี้จะถูกห่อหุ่มด้วย SOAP Envelop และส่งโดยทางเว็บเซอร์วิสไปยังเซอร์เวอร์ ซึ่งเว็บเซอร์วิสจะประมวลผลข้อมูลและทำการ Query ฐานข้อมูลและส่งข้อมูลพื้นที่กลับไปยังอุปกรณ์พกพา ซึ่งอุปกรณ์พกพาจะสามารถแสดงผลข้อมูลได้ผ่านแผนที่ถ้าพบข้อมูลจากเว็บเซอร์วิส ซึ่งข้อมูลพื้นที่จะประกอบด้วยสถานที่น่าสนใจและเส้นทางในบริเวณรอบๆ ตำแหน่งปัจจุบัน เนื่องจากอุปกรณ์พกพายุ่งยากในเรื่องการป้อนข้อความซึ่งเว็บเซอร์วิสบนเครื่องแม่ข่ายจะต้องใช้เป็น meta-data แต่ข้อมูลใหม่ๆ ก็จำเป็นที่จะต้องถูกบันทึกโดยอุปกรณ์พกพาเช่นเดียวกัน การทำงานของอุปกรณ์พกพาและโครงสร้างสถาปัตยกรรมของระบบภายในอุปกรณ์พกพาสามารถแสดงได้ดังรูปที่ 2.19



รูปที่ 2.19 แสดง Use case Diagram ของ mobile device

เฟรมเวิร์คนี้เพียงพอที่จะสามารถสามารถใช้ในการพัฒนาโปรแกรมประยุกต์สำหรับก
รับรู้ตำแหน่งที่ตั้งได้อย่างง่าย และด้วยการใช้เว็บเซอร์วิสเราสามารถทำงานร่วมกันกับเครื่องลูก
ข่ายในทุกแพลทฟอร์มเช่น J2ME หรือ .NET Compact Framework กับฐานข้อมูลเชิงพื้นที่ใดๆ ก็
ได้ ซึ่งก็อาจจะเป็น Microsoft SQL Server, Oracle 10g หรืออื่นๆ และเว็บเซอร์วิสก็สามารถทำการ
ติดตั้งบน Application Server ใดๆ ที่มีอยู่ในทุกวันนี้ได้

ผู้เขียนงานวิจัยนี้ระบุว่ายังพบปัญหาเล็กน้อยในเรื่องของประสิทธิภาพในการส่งผ่านกลุ่ม
ข้อมูลขนาดใหญ่ (raster maps) ผ่านทางเครือข่ายความเร็วต่ำของ GPRS ซึ่งก็สามารถแก้ไขโดยการ
ใช้เครือข่ายที่มีความเร็วสูงกว่ามากอาทิเช่น UMTS enabled mobile networks หรือโดยการแทนที่
raster map ด้วย vectorized counterparts ซึ่งกำลังอยู่ระหว่างการทดสอบความเป็นไปได้ของการ
สร้าง vectorized maps ขึ้นจากเส้นทางที่ถูกเก็บไว้ในฐานข้อมูลเชิงพื้นที่

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 3
วิธีดำเนินงานวิจัย

งานวิจัยนี้มุ่งเน้นการพัฒนากรอบงานเชิงบริการของระบบติดตามการขนส วัตถุประสงค์เพื่อประยุกต์ใช้ในการพัฒนาระบบติดตามห่วงโซ่อุปทานสำหรับการขนส่งซึ่งเป็ ส่วนหนึ่งของระบบโลจีสติกส์โดยจะกรอบงานจะถูกพัฒนาขึ้นโดยภาษาจาวาโดยทำการออกแบบ ให้สามารถรองรับการพัฒนาระบบภายใต้สถาปัตยกรรมเชิงบริการ(Service oriented architecture) หรือSOA ซึ่งมีข้อดีในหลายๆด้านและเป็นรูปแบบสถาปัตยกรรมที่กำลังได้รับความสนใจอย่างมาก ในปัจจุบัน

3.1 ระเบียบวิธีวิจัย

3.1.1 ศึกษางานวิจัยที่เกี่ยวข้อง

3.1.2 ศึกษาระบบ GPS Tracking เพื่อวิเคราะห์หาแนวทางในการพัฒนาต่อยอด โดยการทดสอบการทำงานเบื้องต้นของระบบ โดยใช้กรณีศึกษาคือระบบOpenGTS(Open GPS Tracking System) ซึ่งเป็นระบบโอเพ่นซอสและอนุญาตให้นำไปพัฒนาต่อได้

3.1.3 ศึกษาการใช้งานซอสโค๊ดและเฟรมเวิร์คของระบบ OpenGTS

3.1.4 ศึกษาแนวคิดของสถาปัตยกรรมเชิงบริการ(SOA)

- ศึกษาการพัฒนาเว็บเซอร์วิสภายใต้เทคโนโลยี JAVA

3.1.5 กำหนดปัญหา (Problem Definition) นำข้อมูลจากการศึกษาถึงกระบวนการขนส่งสินค้าจากงานวิจัยที่เกี่ยวข้องมาทำการวิเคราะห์เพื่อแยกแยะปัญหาที่เกิดขึ้น

3.1.6 ศึกษาความเป็นไปได้ (Feasibility Study) จากปัญหาที่เกิดขึ้นจากกระบวนการขนส่งสินค้าและการติดตามการขนส่งด้วยเทคโนโลยี GPS มาแยกแยะเพื่อมองหาแนวคิดที่จะพัฒนาต่อยอดระบบจากแนวคิดของระบบเดิมที่มีอยู่แล้วของระบ1 GPS Tracking (OpenGTS) เพื่อให้สามารถรองรับการพัฒนาตามรูปแบบของสถาปัตกรรมเชิงบริการ

3.1.7 การวิเคราะห์ระบบ (System Analysis) ข้อมูลที่ได้จากการศึกษานำมาแยกแยะกระบวนการทำงานทางด้านการขนส่งของเป็นส่วนย่อยๆ เพื่อนำข้อมูลการวิเคราะห์ไปทำออกแบบโครงสร้างของกรอบงานตามหลักการของ Software Engineering ต่อไป

3.1.8 ออกแบบกรอบงาน (Framework Design) นำผลจากการศึกษาและวิเคราะห์มาทำการออกแบบเฟรมเวิร์คด้วยวิธีการต่างๆ ดังนี้

1. ออกแบบสถาปัตยกรรมของกรอบงาน (Framework Architecture)
2. กำหนดส่วนของบริการ (Service Definition)
3. กำหนด hook, Slot และ API (Application Programming Interface)
4. ออกแบบ Class diagramเพื่ออธิบายโครงสร้างของกรอบงาน

3.1.9 การทดสอบการประยุกต์ใช้กรอบงาน (Framework Implementation) โดยทำการแบ่งการทดสอบเป็น2 ระดับดังนี้

1. ทดสอบ Framework Implementation เพื่อสร้างเป็นเว็บเซอร์วิสบนเครื่องมือที่ใช้พัฒนา โดยการทดสอบการติดต่อกับService Description
2. ทดสอบ Web Service Invoke โดยการสร้างระบบต้นแบบขึ้นมาเพื่อทดสอบความถูกต้องในการทำงานของเซอร์วิส

3.1.10 สรุปผลการทำงาน

ปัจจุบันการขนส่งสินค้าในประเทศไทยซึ่งเป็นส่วนหนึ่งของกระบวนการทางด้านโลจีสติกส์ประสบปัญหาในหลายๆด้านเช่นการติดตามและควบคุมการขนส่งเป็นไปได้ด้วยความยากลำบากกระบวนการขนส่งขาดประสิทธิภาพทำให้เกิดปัญหาความล่าช้าความผิดพลาดในการขนถ่ายสินค้า

จากการศึกษาค้นคว้าและการวิเคราะห์หลักการทำงานของกระบวนการติดตามวัตถุหรือสินทรัพย์ด้วยเทคโนโลยี GPS รวมทั้งการศึกษาความสามารถของซอฟต์แวร์ GPS Tracking (OpenGTS) พบว่าระบบการติดตามสินทรัพย์ด้วยเทคโนโลยี GPS เป็นระบบที่ค่อนข้างซับซ้อน ทั้งในส่วนของการออกแบบโปรแกรมทางฝั่งผู้ให้บริการ(Server) ที่จำเป็นต้องออกแบบให้มีช่องทางที่จะใช้ติดต่อสื่อสารเพื่อรับข้อมูลกับอุปกรณ์ GPS (GPS Tracking Device) ซึ่งในปัจจุบันก็มีหลากหลายยี่ห้อหลากหลายผู้ผลิตซึ่งก็ส่งผลให้ข้อกำหนดในเรื่องของการติดต่อสื่อสารกับ GPSTracking Server ก็มีหลากหลายมาตรฐานเช่นกัน อุปกรณ์แต่ละชนิดรวมถึงความต้องการของผู้ใช้งานแต่ละกลุ่มก็มีความต้องการที่จะนำข้อมูลจากระบบไปใช้ประโยชน์แตกต่างกัน เป็นเหตุให้เพียงแค่การติดตามตำแหน่งได้เพียงอย่างเดียวอาจไม่เพียงพอที่จะรองรับความต้องการได้ทั้งหมด ดังนั้นหากมีการออกแบบระบบที่สามารถรองรับการแลกเปลี่ยนข้อมูลกับระบบอื่นที่เกี่ยวข้องได้ย่อมสามารถทำให้ขีดความสามารถและประโยชน์ที่ได้รับจากระบบ GPS Tracking มีเพิ่มมากขึ้น มูลค่าของข้อมูลที่ได้จากการติดตามตำแหน่งด้วย GPS Tracking ก็จะมีมากขึ้นอีกด้วย

จากปัญหาและข้อจำกัดดังกล่าว งานวิจัยชิ้นนี้จึงได้พัฒนากรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง (Service Based Framework for Supply Chain Tracking in Transportation) ขึ้น ซึ่งเป็นกรอบงาน (Framework) ที่ได้รับการปรับแต่งให้เหมาะสมสำหรับการสร้าง Service Layer เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลด้วยเว็บเซอร์วิสตามหลักการของสถาปัตยกรรมเชิงบริการ Service Oriented Architecture

3.2 การกำหนดปัญหา (Problem Definition)

จากการวิเคราะห์ปัญหาของระบบการขนส่งและลอจิสติกส์พบว่า หากทีมพัฒนา (Development Team) ต้องมีการพัฒนาระบบประเภทนี้ขึ้นมาขึ้นมาใช้ในองค์กร จะต้องประสบปัญหาหลายด้าน ทั้งในเรื่องของต้นทุน (Cost) ในการพัฒนาที่ค่อนข้างสูงความซับซ้อนของระบบซึ่งจำเป็นต้องอาศัยความรู้ความเข้าใจ (Knowledge) จึงเป็นการยากที่จะพัฒนาระบบขึ้นมาใช้เองทั้งหมด และในระหว่างการพัฒนาอาจพบกับปัญหาข้อผิดพลาดในการพัฒนาระบบ ทำให้เวลาที่ใช้ในการพัฒนาเพิ่มมากขึ้นตามไปด้วย รวมทั้งหากต้องการพัฒนาระบบที่มีขีดความสามารถในการแลกเปลี่ยนข้อมูลกับระบบอื่นๆ ในองค์กรย่อมเพิ่มความยุ่งยากและซับซ้อนให้กับการพัฒนาระบบ GPS Tracking ขึ้นอีกด้วย

จากปัญหาดังกล่าว ผู้วิจัยได้ทำการศึกษากระบวนการขนส่งสินค้าและโครงการ OpenGTS (Open GPS Tracking System) ซึ่งเป็นโครงการโอเพ่นซอสรวมทั้งศึกษาแนวคิดการพัฒนาระบบบนพื้นฐานของสถาปัตยกรรมเชิงบริการ (Service Oriented Architecture : SOA) ซึ่งเป็นแนวคิดในการพัฒนาระบบที่กำลังได้รับความนิยมเป็นอย่างสูงในปัจจุบัน และนำแนวคิดทั้งหมดนั้นมาปรับใช้และพัฒนาต่อยอดขึ้นเป็นกรอบงาน เพื่อรองรับการพัฒนาระบบ GPS Tracking ที่มีความสามารถในการทำงานด้านตรวจหาตำแหน่งละควบคุมการขนส่งสินค้ารวมทั้งสามารถเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลด้านการตรวจหาตำแหน่งและควบคุมการขนส่งสินค้าไปใช้ร่วมกับระบบอื่นๆ โดยการสร้าง Service Layer ตามแนวคิดของ SOA ซึ่งจะมีส่วนช่วยในการลดต้นทุนพัฒนาและเพิ่มมูลค่า Logistic Solution สำหรับองค์กรต่างๆ ได้

3.3 การศึกษาความเป็นไปได้ (Feasibility Study)

ผู้วิจัยได้ศึกษาความเป็นไปได้ในการพัฒนากรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง ซึ่งประกอบด้วยปัจจัยต่างๆ ให้พิจารณาดังต่อไปนี้

1. ความเป็นไปได้ในการพัฒนา ส่วนประกอบ (Component) ของระบบขึ้นมาเองทั้งหมด

ผู้วิจัยได้ตัดสินใจเลือกที่จะพัฒนาต่อยอดจากระบบ GPS Tracking เดิมคือ โครงการ OpenGTS (Open GPS Tracking System) เนื่องจาก จากการศึกษาเบื้องต้นพบว่าการพัฒนาระบบ GPS Tracking ขึ้นมานั้น จำเป็นที่จะต้องศึกษาทฤษฎีต่างๆ ที่เกี่ยวข้องหลายด้าน อาทิเช่น ทฤษฎีที่เกี่ยวเนื่องกับเทคโนโลยี GPS (Global Positioning System) , มาตรฐานการเชื่อมต่อของ GPS Device แต่ละรุ่นแต่ละยี่ห้อ, การใช้งาน Map Service ของแต่ละผู้ให้บริการ (Google Map, Microsoft Visual Earth, OpenLayer/OpenStreetMap) เป็นต้น รวมทั้งการทดสอบ Component ในระบบ GPSTracking ซึ่งทำงานแบบ Real-Time ต้องแน่ใจได้ว่ามีประสิทธิภาพที่สูงจนเป็นที่ยอมรับในการใช้งาน

ดังนั้น การเลือกใช้ระบบ OpenGTS ซึ่งเป็นระบบโอเพ่นซอสที่ได้รับการยอมรับและใช้งานในมากกว่า 90 ประเทศทั่วโลก และสามารถรองรับมาตรการเชื่อมต่อของอุปกรณ์ได้หลากหลาย มาเป็นส่วนประกอบเพื่อทำการพัฒนาต่อยอดขึ้นเป็นกรอบงานจึงเป็นตัวเลือกที่ดีกว่าในทุกๆ ด้าน

2. ความเป็นไปได้ในทางเทคนิค (Technical Feasibility)

ผู้วิจัยได้ตัดสินใจเลือกใช้ภาษา JAVA ในการพัฒนากรอบงาน เนื่องจากเป็นภาษาโปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งมีเครื่องมือในการพัฒนาที่มีประสิทธิภาพสูง มีส่วนประกอบต่างๆ ที่รองรับการพัฒนาระบบบนพื้นฐานสถาปัตยกรรมเชิงบริการได้ อีกทั้งยังเป็นโอเพ่นซอสอีกด้วยและนอกจากนี้โครงการ OpenGTS (Open GPS Tracking System) ถูกพัฒนาขึ้นมาด้วยภาษา JAVA ซึ่งการพัฒนาต่อยอดจำเป็นต้องใช้ภาษา JAVA เช่นกัน

3. ความเป็นไปได้ในการลงทุน (Economic Feasibility)

เนื่องจากเครื่องมือในการพัฒนาระบบทั้งหมดเป็นโอเพ่นซอสจึงสามารถนำมาพัฒนาต่อยอดได้โดยไม่มีค่าใช้จ่ายแต่อย่างใด

ในส่วนของอุปกรณ์ GPS(GPS Device) ซึ่งจำเป็นในการนำมาทดสอบระบบ ผู้วิจัยได้เลือกใช้อุปกรณ์โทรศัพท์พกพาที่มีการติดตั้งอุปกรณ์รับสัญญาณ (Smart Phone) ซึ่งในปัจจุบันมีราคาไม่สูงนักและง่ายต่อการปรับแต่งและเขียนโปรแกรมควบคุม

3.4 การวิเคราะห์ระบบ (System Analysis)

จากการศึกษากระบวนการเบื้องต้นในการขนส่งสินค้าสามารถสรุปเป็นตารางได้ดังนี้

ตารางที่ 3.1 กระบวนการทั่วไปในการขนส่งสินค้า

Business Step	Action
Preparation	จัดหีบห่อและทำเครื่องหมายตัวสินค้า
Loaded	ทำการบรรทุกสินค้าขึ้นบนยานพาหนะขนส่ง
Shipping out	ทำการขนส่งถึงที่หมาย
Unload	เมื่อถึงที่หมายจึงมีการนำออกจากยานพาหนะขนส่ง

กระบวนการจะเริ่มขึ้นจากการจัดการเรื่องหีบห่อ การทำเครื่องหมายและบันทึกข้อมูลของสินค้าที่จะทำการขนส่งเพื่อจัดเตรียมการขนส่ง จากนั้นก็จะมีการนำสินค้าที่ได้ทำการจัดเตรียมไว้มาทำการบรรทุกขึ้นบนยานพาหนะขนส่งที่ได้ระบุไว้และมีการขนส่งไปยังปลายทาง เมื่อถึงปลายทางซึ่งอาจจะเป็นผู้รับ สถานีย่อย หรือศูนย์กระจายสินค้า ก็จะมีการนำลงจากยานพาหนะเพื่อคัดแยก จัดเก็บ นำไปผลิตหรือส่งต่อ ต่อไป

ซึ่งในระหว่างที่มีการขนส่งไปยังปลายทาง ก็จะมีการตรวจสอบและรายงานตำแหน่งของยานพาหนะในแบบ Real-time โดยผ่านทาง GPS Tracking Device ซึ่งติดตั้งอยู่กับยานพาหนะมายังเครื่องแม่ข่ายขององค์กรซึ่งสามารถทำการตรวจสอบและติดตามได้ตลอดเวลา

จากการวิเคราะห์ระบบเชิงบริการ (Service-Oriented Analysis) สามารถกำหนดเซอร์วิสและ Component ของระบบได้ดังนี้

1. BusinessDataImportServiceมีหน้าที่จัดการในส่วนการนำเข้าข้อมูลการจัดส่งสินค้าหรือสิ่งของ ทั้งในส่วนการนำเข้าข้อมูลโดยตรงหรือนำเข้าผ่านการเชื่อมต่อกับระบบอื่น

2. PackageManagementService มีหน้าที่จัดการในส่วนการจัดระเบียบของหีบห่อในการจัดส่งทั้งในส่วนของการจัดกลุ่มและการระบุตัวสิ่งของเช่นการกำหนดรหัสการจัดส่งด้วยบาร์โค๊ดหรือ RFID

3. GPS-vehicleAssociationService มีหน้าที่จัดการในส่วนของการติดตั้งระบบระบุตำแหน่งของพาหนะขนส่งด้วยอุปกรณ์ตรวจหาพิกัดโดยการใช้ดาวเทียม

4. ShipmentManagementService มีหน้าที่จัดการในส่วนของการนำสินค้าหรือสิ่งของขึ้นสู่พาหนะจัดส่ง

5. GeoZoneInfoService มีหน้าที่ให้บริการในการตำแหน่งพิกัดและข้อ
เบื้องต้นของสถานีขนถ่ายสินค้า

6. GPS PositionCaptureService มีหน้าที่รับตำแหน่งพิกัดดาวเทียมที่ถูกส่งมาจาก
อุปกรณ์GPS ที่ถูกติดตั้งในพาหนะขนส่ง

7. TransportMonitoringModule มีหน้าที่แสดงตำแหน่งและข้อมูลสถานะข
การขนส่งปัจจุบันของสินค้า

8. InformationQueryModuleมีหน้าที่แสดงข้อมูลประวัติการขนส่งหรือข้อมูล
อื่นๆ เพื่อการติดตามและตรวจสอบย้อนหลังในการจัดส่งสินค้าหรือสิ่งของ

จากการแยกแยะเซอร์วิสและ Component ของระบบเมื่อนำมาทำการวิเคราะห์
ร่วมกับ Component และความสามารถของระบบ OpenGTS ดั้งเดิมซึ่งสามารถนำเอา Component
กลับมาใช้ประโยชน์ใหม่แทนการสร้างขึ้นใหม่ทั้งหมด (Reuse) ทำให้สามารถลดเซอร์วิสที่ผู้พัฒนา
จะต้องพัฒนาขึ้นได้ดังนี้

1. BusinessDataImportServiceมีหน้าที่จัดการในส่วนการนำเข้าข้อมูลการจัดส่ง
สินค้าหรือสิ่งของ ทั้งในส่วนการนำเข้าข้อมูลโดยตรงหรือนำเข้าผ่านการเชื่อมต่อกับระบบ
อื่น

2. PackageManagementService มีหน้าที่จัดการในส่วนการจัดระเบียบของหีบห่อ
ในการจัดส่งทั้งในส่วนของการจัดกลุ่มและการระบุตัวสิ่งของเช่นการกำหนดรหัสกา
จัดส่งด้วยบาร์โค๊ดหรือ RFID

3. ShipmentManagementService มีหน้าที่จัดการในส่วนของการนำสินค้าหรือ
สิ่งของขึ้นสู่พาหนะจัดส่ง

นอกจากนี้ยังมีเซอร์วิสเพิ่มเติมที่จำเป็นต้องมีการพัฒนาโดยการห่อหุ้มลอจิกการ
ทำงานของระบบ OpenGTS เพื่อให้สามารถนำข้อมูลจากการตรวจสอบและติดตามด้วย GPS ไปใช้
ในระบบอื่นๆ ได้ดังนี้

1. GeoZoneInfoService มีหน้าที่ให้บริการข้อมูลเกี่ยวกับพิกัดตำแหน่ง ซึ่งอาจจะ
เป็นตำแหน่งบริเวณสำคัญ, จุดตรวจสอบ(Check Point), สถานีขนถ่ายสินค้าหรือปลายทาง

2. GPSEventInfoService มีหน้าที่ให้บริการข้อมูลเกี่ยวกับตำแหน่งทางดาเทียมที่
ได้จากการติดตามด้วย GPS ของระบบ OpenGTS

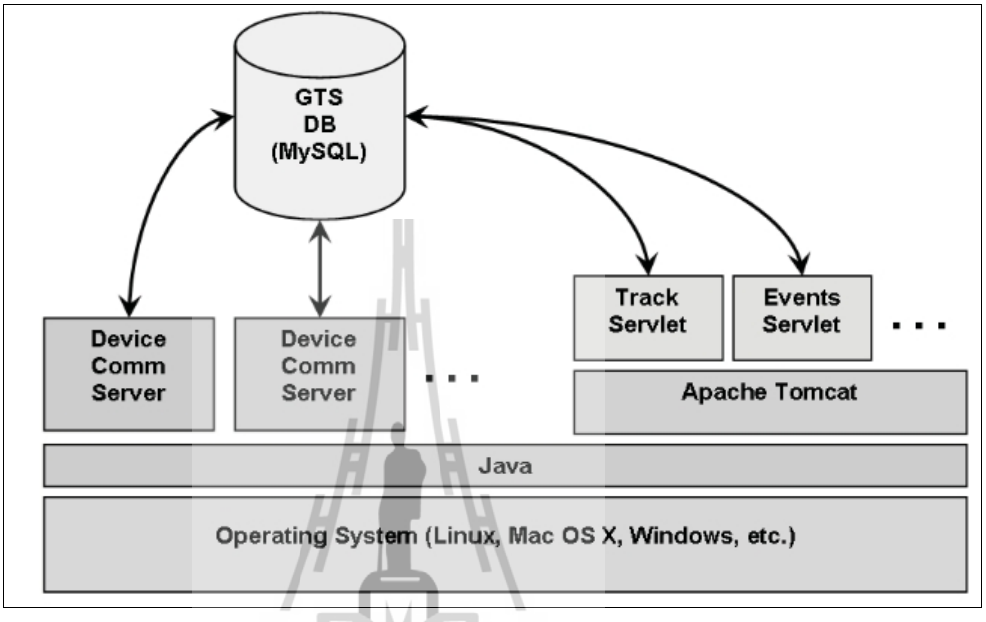
3. UtilityService มีหน้าที่ให้บริการข้อมูลอื่นๆ ที่จำเป็นจากระบบ OpenGTS เช่น
ข้อมูลที่เกี่ยวข้องกับ GPSDevice เป็นต้น

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.5 การออกแบบระบบ (System Design)

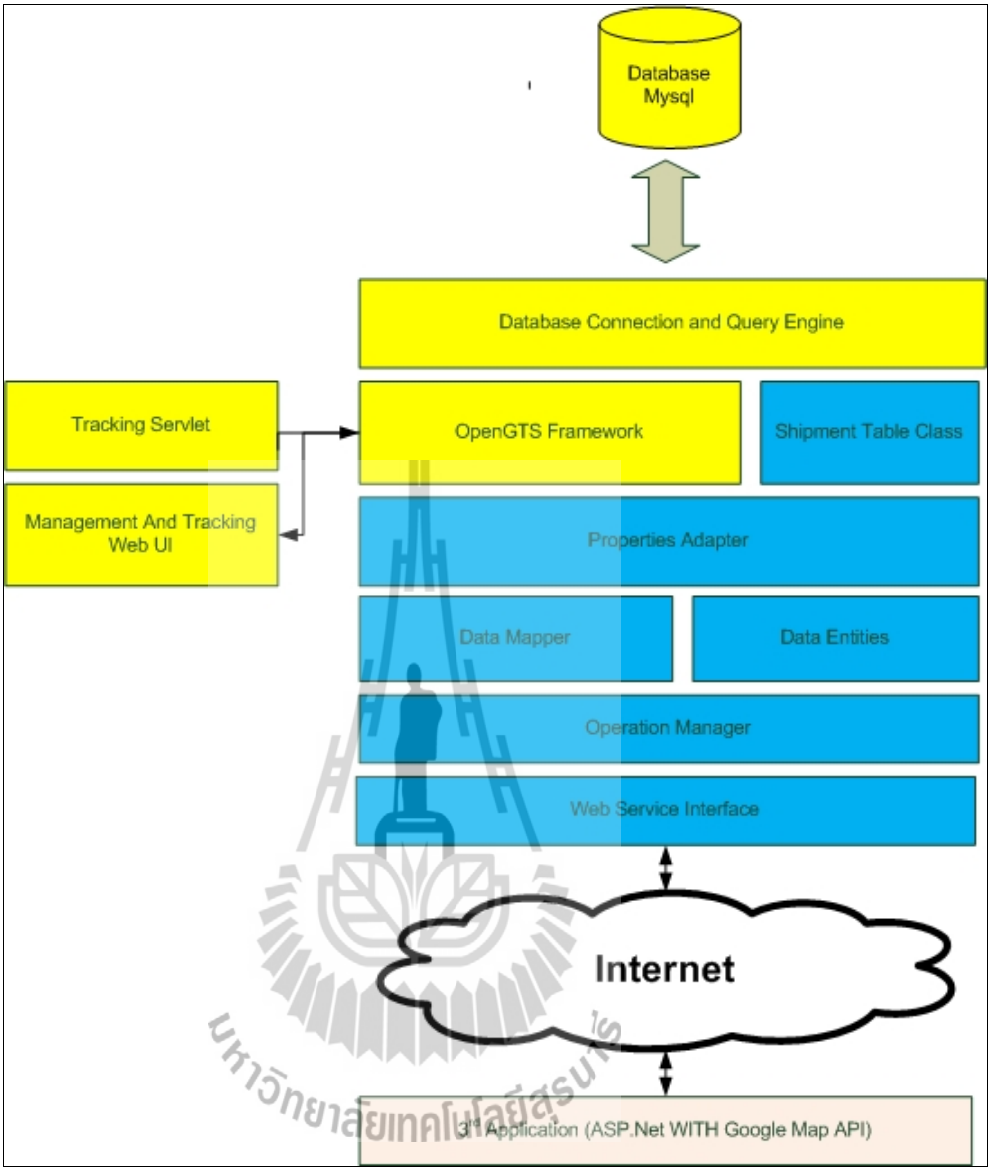
3.5.1 ออกแบบสถาปัตยกรรมของเฟรมเวิร์ค (Framework Architecture Design)

ผู้วิจัยได้ทำการออกแบบเฟรมเวิร์คโดยการพัฒนาต่อยอดจาก ระบบ OpenGTS ซึ่งเป็นระบบโอเพ่นซอสที่ถูกพัฒนาขึ้นจากภาษา Java และประกอบด้วย module ต่างๆ ดังรูป



รูปที่ 3.1 OpenGTS Architecture Overview

ซึ่งเมื่อมีการนำมาพัฒนาต่อยอดได้มีการสร้าง module เพิ่มเติมเพื่อรองรับงานทางด้านการขนส่งสินค้าและปรับแต่งเพื่อให้รองรับการนำไปอิมพลีเมนต์เป็นเว็บเซอร์วิสเพื่อแลกเปลี่ยนข้อมูลที่ได้จากการติดตามยานพาหะกับระบบอื่นๆ ต่อไป



รูปที่ 3.2 แสดงสถาปัตยกรรมของกรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง

จากรูปที่ 3.2จะเห็นได้ว่าเฟรมเวิร์คที่สร้างขึ้นนั้นจะประกอบด้วย 2 ส่วนใหญ่ๆ คือ

1.ส่วนของ component ของระบบ OpenGTS เดิมซึ่งผู้วิจัยเห็นว่าสามารถนำมาใช้แทนการพัฒนาขึ้นใหม่ได้ซึ่งจะช่วยลดขั้นตอนในการพัฒนาระบบและข้อผิดพลาดต่างๆ ที่อาจจะเกิดขึ้นระหว่างการพัฒนาระบบได้เป็นอย่างมาก รวมทั้งยังสามารถใช้ประโยชน์จากmodule ต่างๆ ที่มีอยู่ใน OpenGTS ได้ ซึ่งจะช่วยให้เฟรมเวิร์คที่พัฒนาขึ้นสามารถรองรับมาตรฐานของ GPS Tracking รวมทั้งสามารถทำงานร่วมกับ GPS Device ที่มาจากหลายผู้ผลิตได้อีกด้วย

2.ส่วนที่ผู้วิจัยทำการพัฒนาขึ้นเพิ่มเติมเพื่อให้เฟรมเวิร์คสามารถรองรับงานด้านการขน - ส่งสินค้า รวมทั้งให้บริการข้อมูลแก่โปรแกรมประยุกต์อื่นโดยผ่านช่องทางของเว็บเซอร์วิสได้ซึ่งจะประกอบด้วยส่วนย่อยๆ ดังนี้

- Shipment Table Class ในส่วนนี้จะเป็น Class ที่เป็นโครงสร้างของตารางข้อมูลที่จะทำการสร้างตารางข้อมูลโดยอาศัยช่องทางของ OpenGTS Frameworkซึ่งกลุ่มคลาสเหล่านี้จะมีการอิมพลีเมนต์ตามรูปแบบและข้อกำหนดของ OpenGTS Framework เพื่อให้ทำงานเข้ากับระบบเดิมได้อย่างสมบูรณ์

- Properties Adapter ในส่วนนี้ ได้ออกแบบไว้เพื่อรองรับการสร้างคุณสมบัติเพิ่มเติมของข้อมูลที่ต้องการจะเก็บลงฐานข้อมูล เนื่องจากแต่ละองค์กรแต่ละธุรกิจย่อมมีความต้องการใช้หรือเก็บข้อมูลที่แตกต่างกันไป ตามรูปแบบการบริหารจัดการ ประเภทสินค้า หรือเทคนิคการจัดส่ การมีส่วนนี้จะช่วยให้การอิมพลีเมนต์เฟรมเวิร์คให้รองรับระบบของแต่ละองค์กรเป็นไปได้ยิ่งขึ้น

- DataEntities ในส่วนนี้ จะเป็นกลุ่มEntities Class ซึ่งมีหน้าที่เป็นโครงสร้างของข้อมูลที่จะทำการแลกเปลี่ยนกับระบบอื่นๆ ภายนอก โดยอาศัยเว็บเซอร์วิสได้อย่างมีประสิทธิภาพ เนื่องจาก OpenGTS ไม่ได้ออกแบบมาเพื่อรองรับหลักการของ SOA โดยตรง Class ต่างๆ ที่สร้างขึ้นมาจึงไม่สามารถรองรับการทำการแลกเปลี่ยนข้อมูลโดยอาศัยเทคโนโลยีเว็บเซอร์วิสได้ จึงจำเป็นต้องมีการพัฒนาส่วนนี้ขึ้นมา

- DataMapper ส่วนนี้เป็นInterface ซึ่งประกอบด้วย method ต่างๆ ที่จำเป็นในการจับคู่ข้อมูลจากส่วนของ DataEntities และ Properties Adapter เพื่อให้การห่อหุ้มข้อมูลของเว็บเซอร์วิสเป็นไปได้อย่างสมบูรณ์ ซึ่งในส่วนนี้นักพัฒนาจำเป็นต้องทำการอิมพลีเมนต์เพื่อให้ระบบที่สร้างขึ้นจากเฟรมเวิร์คนี้ทำงานได้จริง

- OperationManager เป็นกลุ่มคลาสที่รวบรวมmethod หรือ Operation ต่างๆ ที่จำเป็นในการทำงานกับข้อมูลทั้งในส่วนของการ Create, Upadate, Query,Delete ข้อมูลเพื่อลดความซับซ้อนของการเข้าถึง method ต่างๆ ที่เกี่ยวกับ Data manipulation ของ OpenGTS Framework พร้อมทั้งยังเป็นตัวกลางที่จะเรียกใช้คลาสที่ทำหน้าที่เป็น DataMapper เพื่อทำการจับคู่ข้อมูลจากภายนอกและภายในระบบอีกด้วย

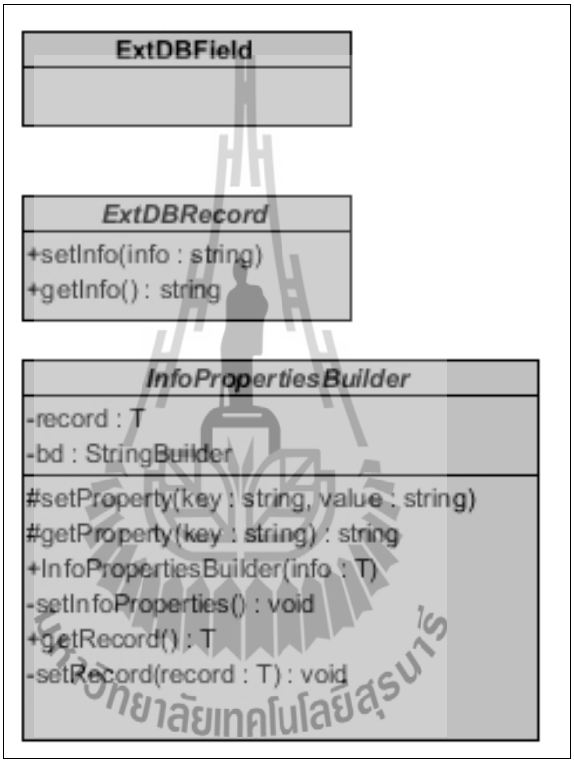
- Web Service Interface เป็นกลุ่มคลาสที่พร้อมที่จะอิมพลีเมนต์เพื่อนำไปสร้างเป็นเว็บเซอร์วิสด้วยWeb Service framework ต่างๆ ในงานวิจัยชิ้นนี้ผู้วิจัยได้ใช้ Apache CFX 2.5.2 เป็นตัวอย่างในการอิมพลีเมนต์ซึ่งคลาสกลุ่มนี้ยังเป็น Abstact Class ซึ่งจะยังไม่สามารถทำงานได้ ต้องทำการอิมพลีเมนต์ Slot ที่กำหนดไว้ก่อนจึงจะทำงานได้

3.5.2 Package Design

ทำการพัฒนาส่วนประกอบของระบบเพิ่มเติมซึ่งได้มีการแบ่งแยกเป็น 5 Package ดังนี้

1. ชื่อแพคเกจ org.opengts.custom.dbtools

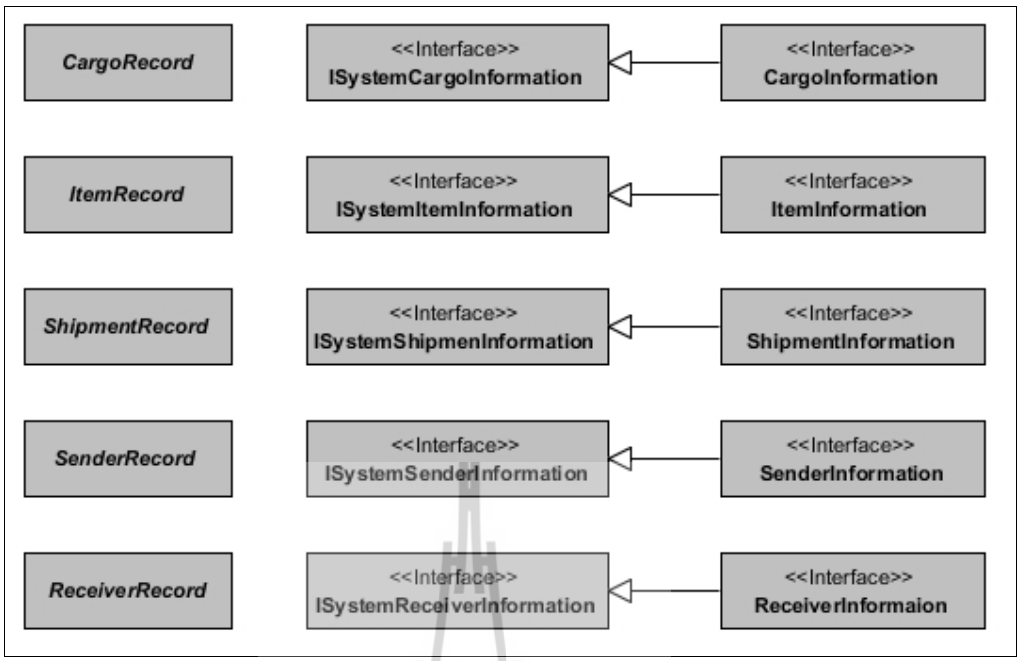
แพคแกจนี้ประกอบด้วยกลุ่มคลาสที่เป็นพื้นฐานในการสร้างคลาสเพื่อกำหนดรูปแบบของตารางข้อมูลในฐานข้อมูล รวมทั้งควาสแม่ที่จำเป็นของเฟรมเวิร์คซึ่งคลาสส่วนใหญ่จะสืบทอดคุณสมบัติมาจากคลาสที่มีอยู่แล้วใน OpenGTS Framework



รูปที่ 3.3 แสดงคลาสที่มีอยู่ใน package org.opengts.custom.dbtools

2. ชื่อแพคเกจ org.opengts.custom.db

แพคเกจนี้จะรวบรวม Base Class และ Interface ที่จำเป็นในการสร้าง Table Class เพื่อรองรับระบบการขนส่งสินค้า ซึ่ง class และ Interface ที่อยู่ใน package นี้จะถูกอิมพลีเมนต์ในแพคเกจ org.opengts.custom.db.tables ต่อไป

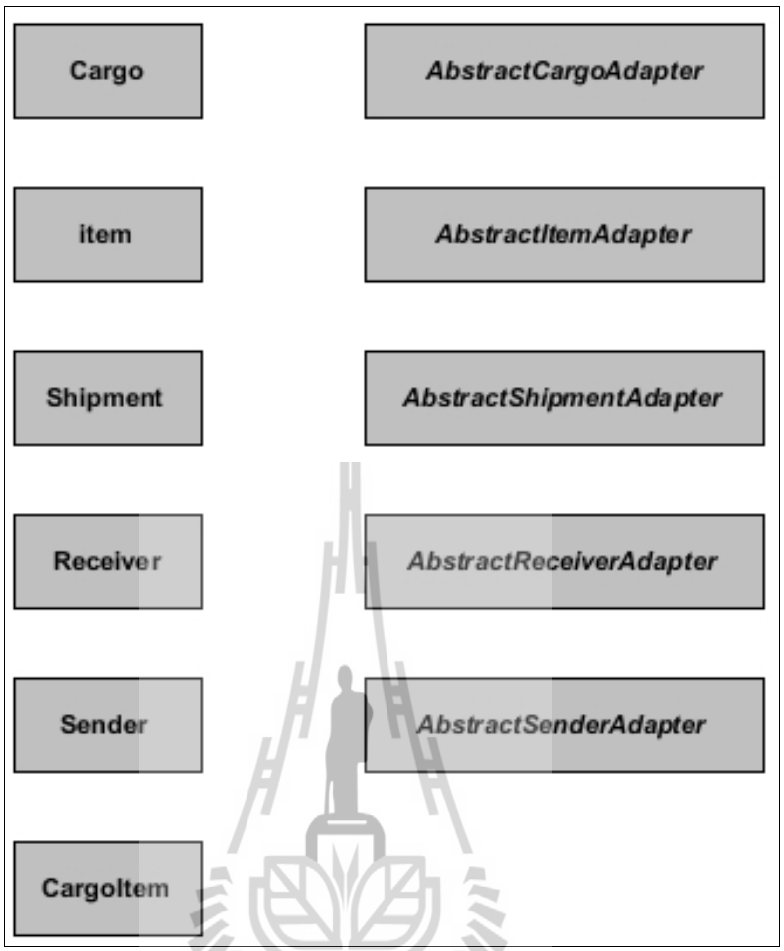


รูปที่ 3.4 แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db

3. ชื่อแพคเกจ org.opengts.custom.db.table

แพคเกจนี้จะเป็นส่วนของ Table class และ Adapter class ซึ่งยังคงเป็น abstract ไว้เพื่อให้นักพัฒนาที่ต้องการนำเฟรมเวิร์คไปทำการพัฒนาระบบสามารถเพิ่มเติม properties ที่ต้องการได้ง่าย Table class จะเป็นส่วนที่กำหนดโครงสร้างของตารางข้อมูลที่จะถูกระบบสร้างขึ้น เช่น Cargo, Item, Shipment, Receiver , Sender, CargoItem

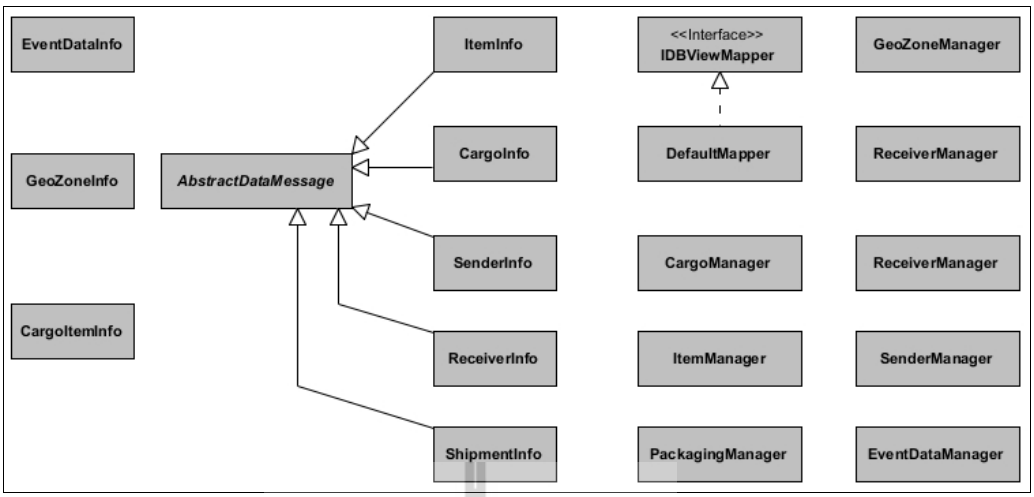
Adapter class จะถูกต้องถูกอิมพลีเมนต์เพื่อนำไปใช้ร่วมกับคลาสอื่นในการทำ DataMapper เพื่อให้ข้อมูลภายนอกและภายในระบบเชื่อมโยงกันได้



รูปที่ 3.5 แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db.table

4. ชื่อแพคเกจ org.opengts.custom.db.views

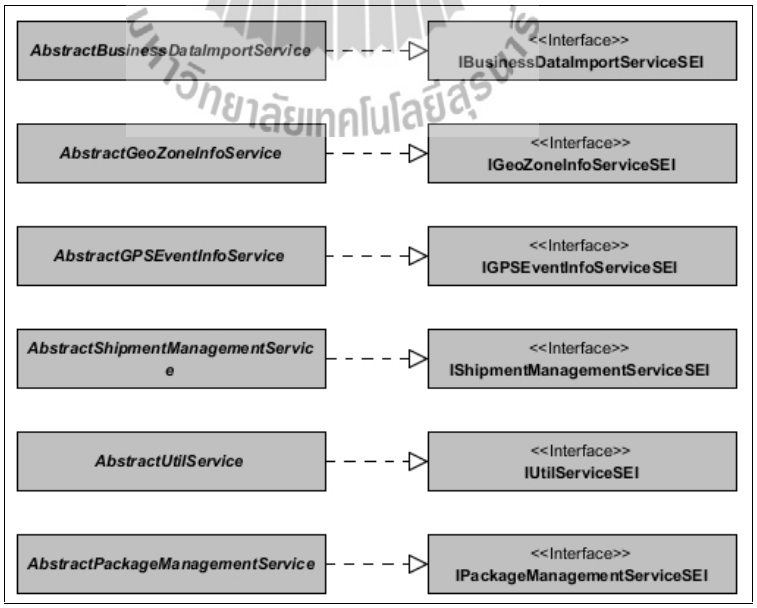
แพคเกจนี้จะเป็นส่วนของ External view ซึ่งจะประกอบด้วยกลุ่มคลาสที่ทำหน้าที่เป็นตัวแทน Entities ที่จะกลายเป็นข้อมูลที่เป็น Complex Type ของเว็บเซอร์วิสเช่น CargoInfo, ItemInfo, SenderInfo, ReceiverInfo, ShipmentInfo, EventDataInfo, GeoZoneInfo เป็นต้น และนอกจากนี้ยังประกอบด้วย Mapper Interface ซึ่งประกาศ method ที่จำเป็นในการจับคู่ข้อมูลที่เป็น External View กับ Data table ของระบบ OpenGTS รวมทั้งมีคลาสกลุ่ม DataManager ที่รวบรวม Method จำเป็นที่ Service class จำเป็นต้องเรียกใช้ไว้อีกด้วย



รูปที่ 3.6 แสดงคลาสที่มีอยู่ใน package org.opengts.custom.db.views

5. ชื่อแพคเกจ org.opengts.custom.services

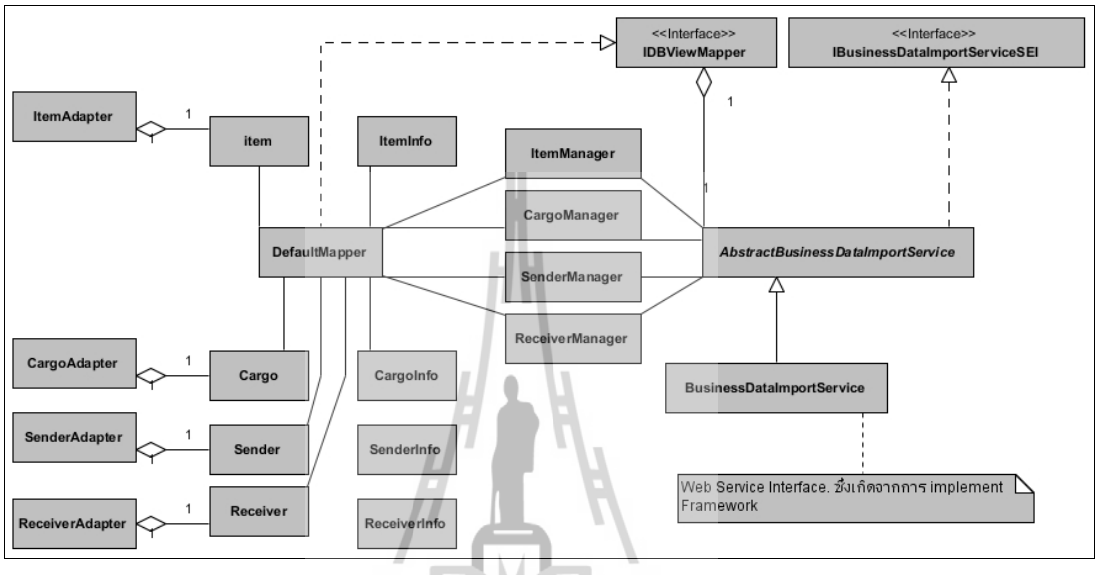
แพคเกจนี้เป็น Abstract class ซึ่งเป็น service interface ของเฟรมเวิร์คซึ่งจะประกอบด้วย method ซึ่งจะกลายเป็น Service Operation ของเว็บเซอร์วิสเมื่อการอิมพลีเมนต์เฟรมเวิร์คเสร็จสิ้นแล้ว ซึ่งเซอร์วิสแต่ละตัวจะมีลักษณะการทำงานที่แยกอิสระจากกัน นักพัฒนาสามารถเลือพัฒนาขึ้นทีละเซอร์วิสได้โดยไม่กระทบเซอร์วิสอื่นๆ



รูปที่ 3.7 แสดงคลาสที่มีอยู่ใน package org.opengts.custom.services

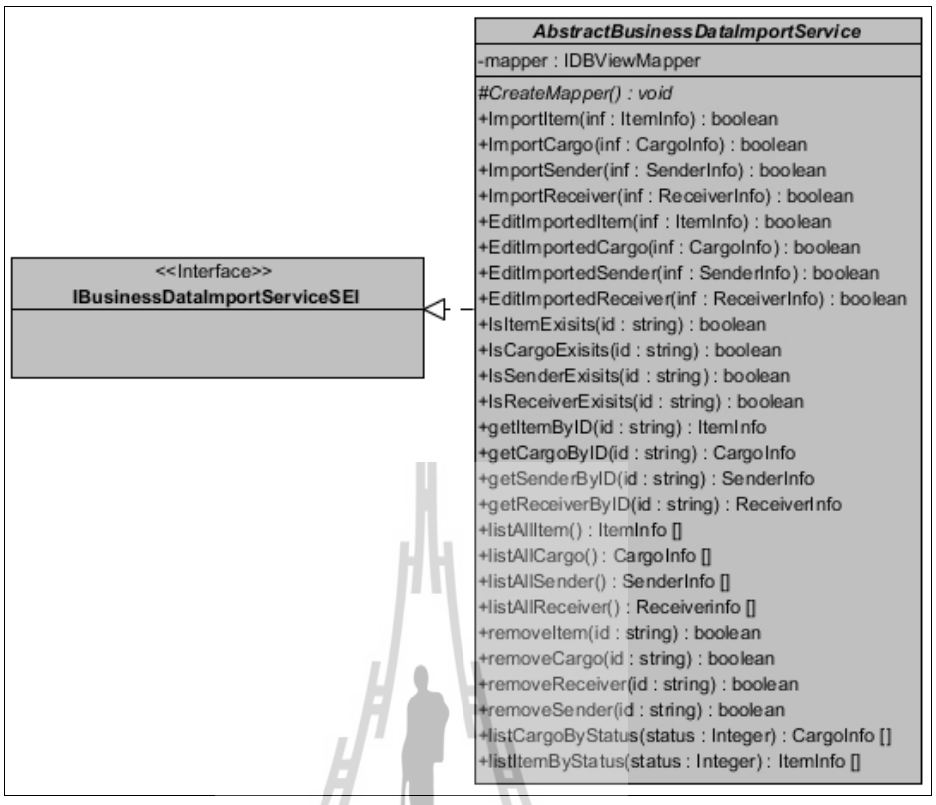
จากเซอร์วิสที่ได้ทำการวิเคราะห์แยกแยะออกมาเป็นที่เรียบร้อยแล้วเราสามารถนำแต่ละเซอร์วิสมาวิเคราะห์และแสดงโครงสร้างและความสำคัญในของแต่ละคลาสที่ประกอบกันขึ้นเป็นแต่ละเซอร์วิสในFramework ด้วย Class Diagram ได้ดังนี้

1. BusinessDataImportService



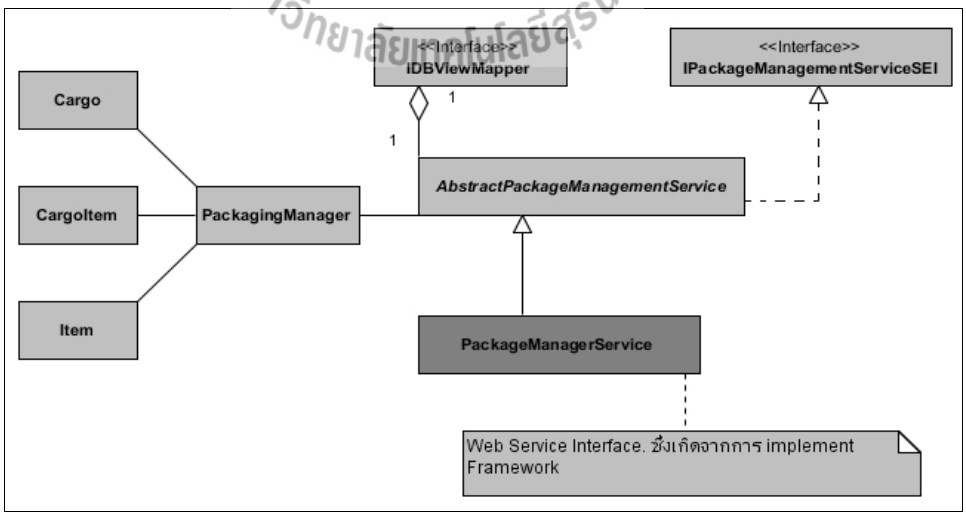
รูปที่ 3.8 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น BusinessDataImportService

มหาวิทยาลัยเทคโนโลยีสุรนารี

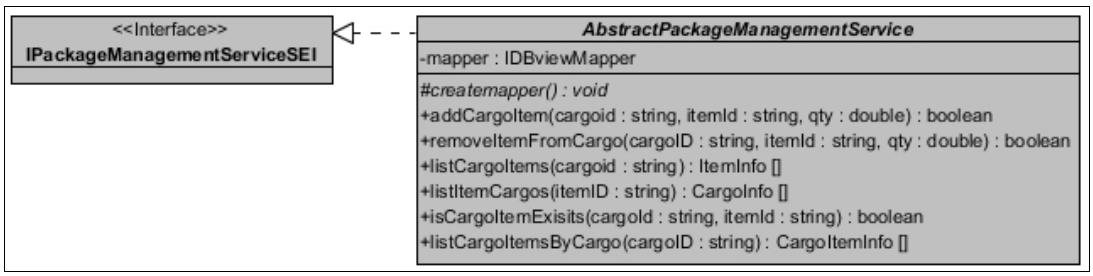


รูปที่ 3.9 แสดง Operation ของ BusinessDataImportService

2. PackageManagementService

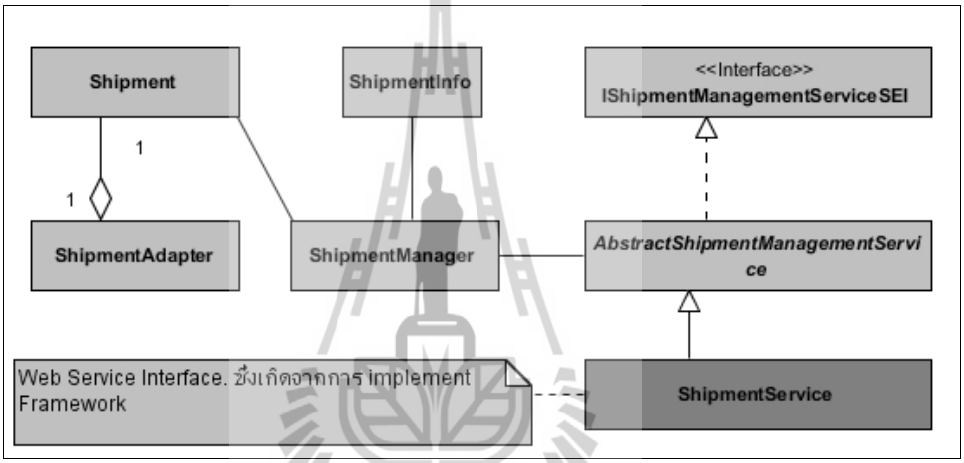


รูปที่ 3.10 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น PackageManagementService

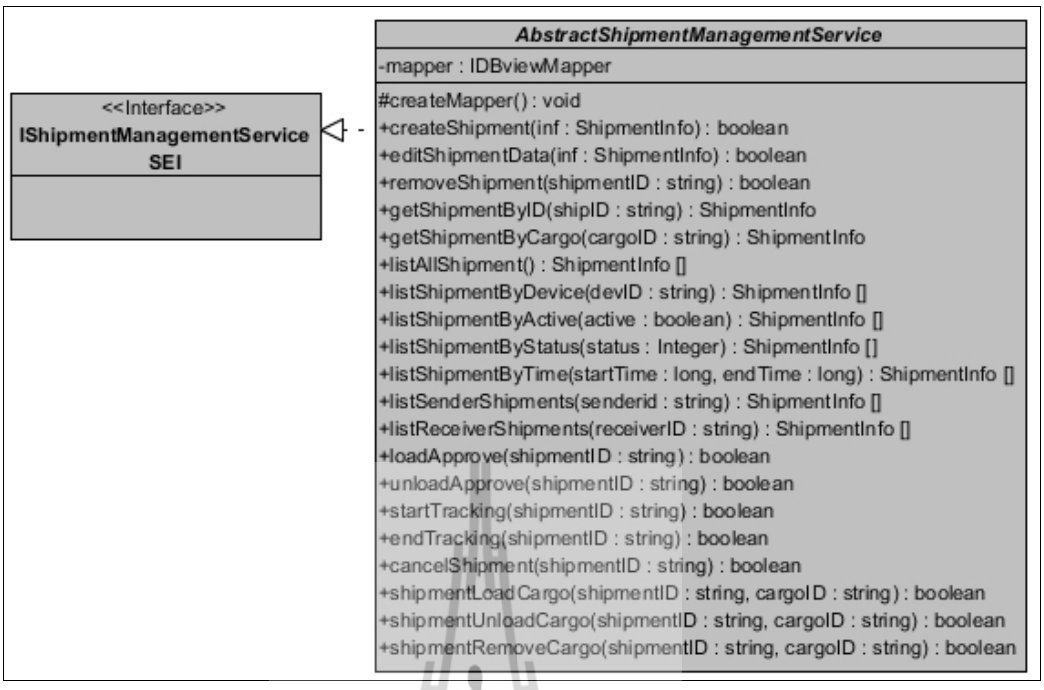


รูปที่ 3.11 แสดง Operation ของ PackageManagementService

3. ShipmentManagementService

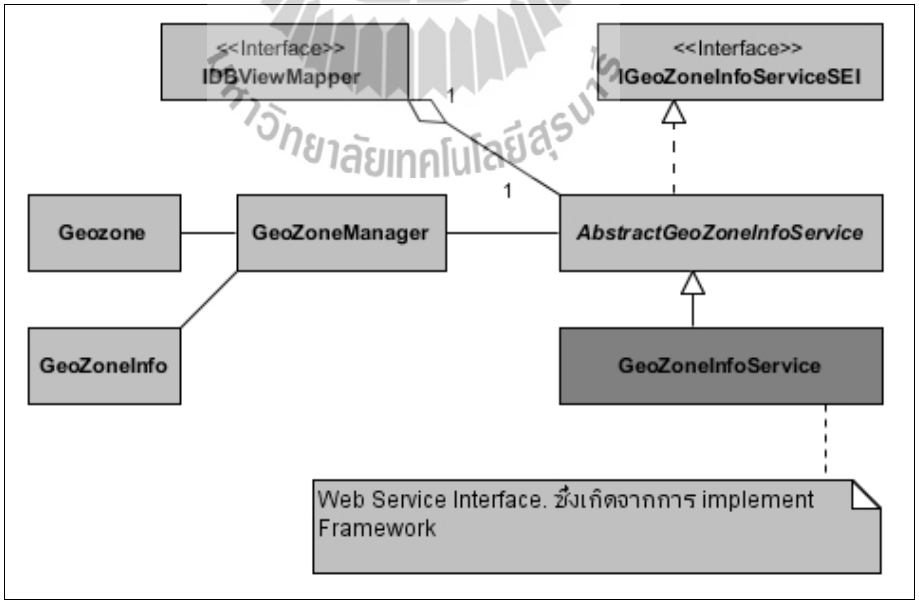


รูปที่ 3.12 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น ShipmentManagementService

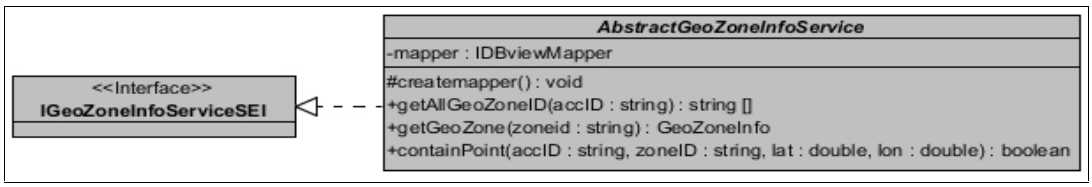


รูปที่ 3.13 แสดง Operation ของ ShipmentManagementService

4. GeoZoneInfoService

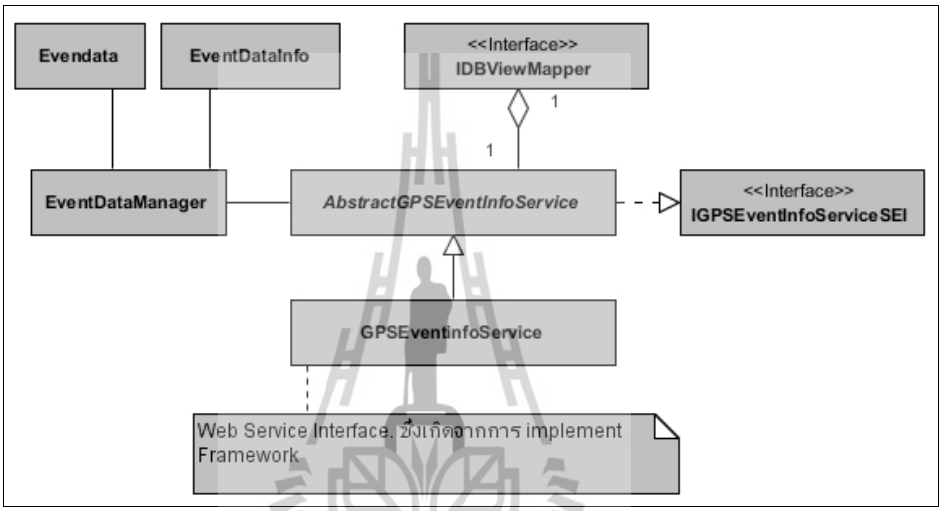


รูปที่ 3.14 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น GeoZoneInfoService

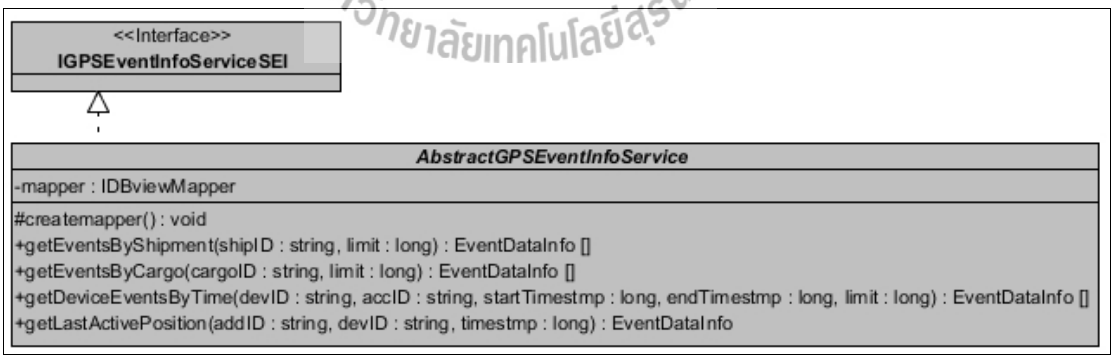


รูปที่ 3.15 แสดง Operation ของ GeoZoneInfoService

5. GPSEventInfoService

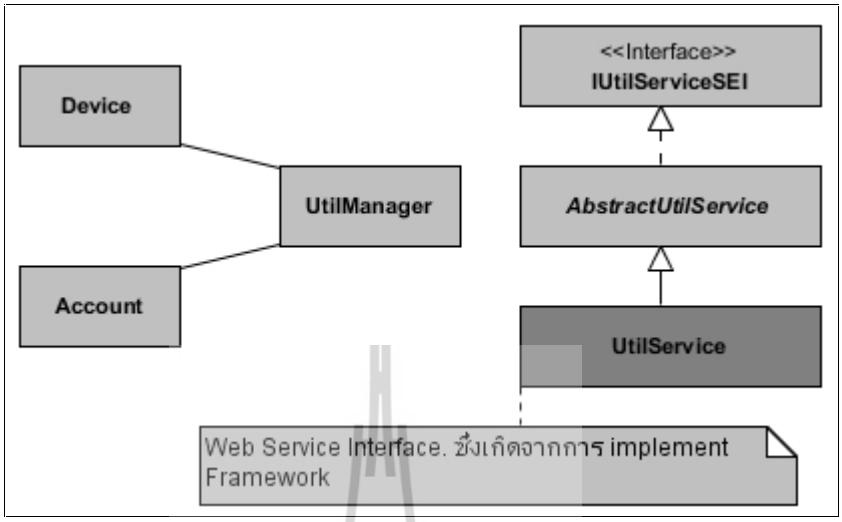


รูปที่ 3.16 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น GPSEventInfoService

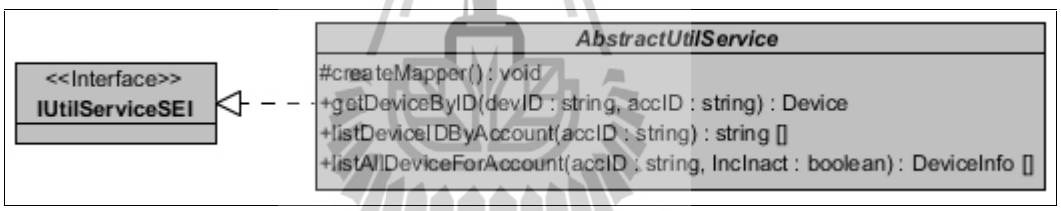


รูปที่ 3.17 แสดง Operation ของ GPSEventInfoService

6. UtilService



รูปที่ 3.18 แสดงโครงสร้างและความสัมพันธ์ของคลาสที่ประกอบขึ้นเป็น UtilService



รูปที่ 3.19 แสดง Operation ของ UtilService

3.6 การพัฒนาระบบ (Implementation)

ในขั้นตอนการพัฒนา framework นั้นได้นำเอาข้อมูลต่างๆ ที่ได้จากการวิเคราะห์และออกแบบระบบและต้นแบบ (Model) ที่ได้ออกแบบไว้ ซึ่งสามารถแบ่งขั้นตอนการพัฒนาเป็น 2 ขั้น ดังนี้

3.6.1 จัดเตรียมสิ่งแวดล้อมสำหรับการพัฒนาระบบและตั้งค่าเริ่มต้น

เนื่องจากการพัฒนากรอบงานเชิงบริการสำหรับติดตามห่วงโซ่อุปทานในการขนส่งนี้ เป็นการพัฒนาต่อยอดจากระบบ OpenGTS(Open GPS Tracking System) ซึ่งก่อนที่จะเริ่มซอสโค๊ดจำเป็นที่จะต้องจัดการในส่วนของสิ่งแวดล้อมในการพัฒนาเพื่อให้ระบบ OpenGTS สามารถทำงานได้ ก่อนที่จะเพิ่มเติม module ที่ต้องพัฒนาเพิ่มเติมเข้าไป ซึ่งมีรายละเอียดดังนี้

Hardware and Software Environment Pre-configuration

การพัฒนากรอบงานเชิงบริการสำหรับติดตามห่วงโซอุปทานในการขนส่ง ผู้วิจัยได้จัดเตรียม hardware และซอฟต์แวร์ดังนี้

Hardware Component :

1. เครื่องคอมพิวเตอร์ Notebook Lenovo Y560 CPU i7 740 1.73 Ghz 4 Gb Ram HDD 500 Gb

Windows 7 Operating System

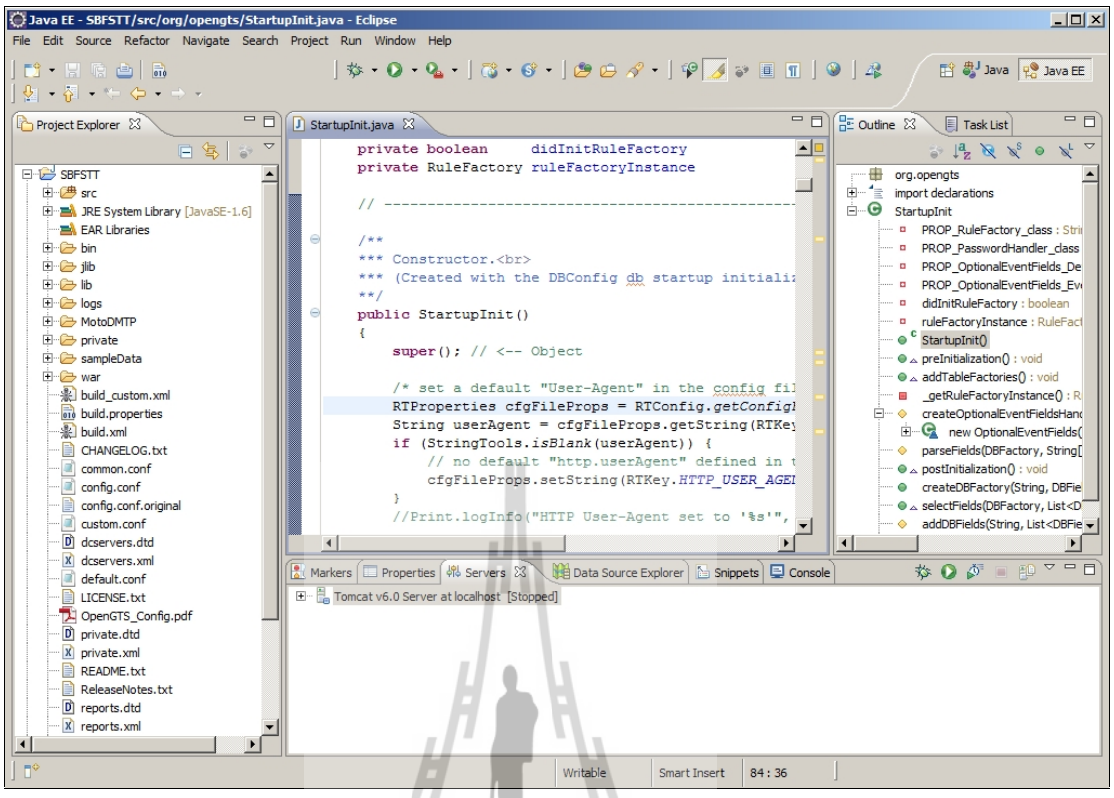
2. Smart Phone Garmin-Asus nuvipone M10 windows mobile 6.5.3 with GPS Receiver

Software Component :

1. Java Platform JDK 6
2. Mysql 5.0.51b Database Server
3. Apche tomcat 6.0 Web Server
4. Apache CXF 2.5.2
5. Mysql Connector 5.1.15
6. OpenGTS 2.3.5
7. Eclipse Java EE IDE for Web Developers. Helios Service Release 2

ในการพัฒนา ผู้วิจัยเลือกใช้เครื่องมือในการพัฒนาคือ Eclipse Java EE IDE for web Developer เป็นเครื่องมือหลักในการพัฒนาระบบ เนื่องจากซอสโค๊ดของ OpenGTS สามารถตั้งค่าให้พัฒนาระบบบน Eclipse ได้ (Pierluigi Bucolo : OpenGTS How to Use Eclipse with OpenGTS projectV 0.0.3)

มหาวิทยาลัยเทคโนโลยีสุรนารี



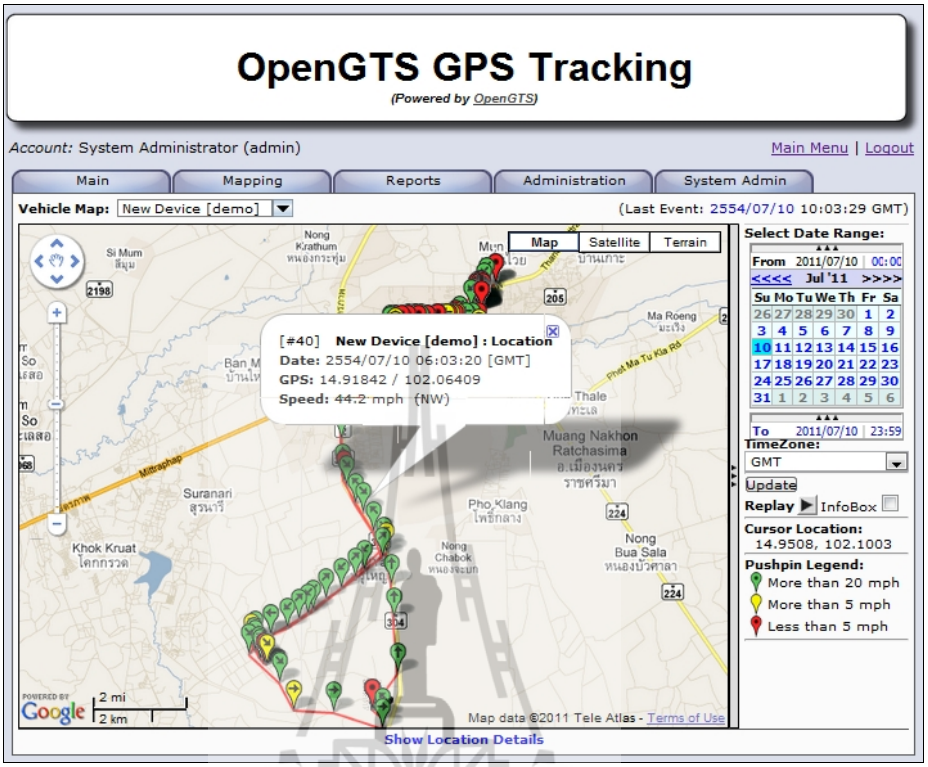
รูปที่ 3.20 แสดง OpenGTS Project หลังจากตั้งค่าให้ทำงานบน Eclipse IDE

เมื่อทำการติดตั้ง OpenGTS ซอสโค๊ดเรียบร้อยแล้วทำการ Build และ ติดตั้งระบบ OpenGTS (OpenGTS Configuration and Installation Manual) ซึ่งงานวิจัยนี้ ได้ทำการตั้งค่าของระบบ OpenGTS ให้ใช้ Google Map เป็น Map Service เพื่อแสดงผลการติดตามด้วย GPS และเลือกใช้ HTTP Based Device Communication Server ซึ่งเป็น Java Servlet ในการรับข้อมูลจากอุปกรณ์ GPS (GPS Device) ซึ่งมีตัวอย่างการส่งข้อมูลไปยัง Servlet ดังนี้

ตัวอย่างการส่งข้อมูลจาก GPS Device ไปยัง HTTP Base Device Communication Server

```
http://archxeron.no-ip.org:9090/gprmc/Data
?id=0837357404&code=0xF020&lat=15.09195056&lon=104.33408614&time=2
1/1/2555%201:50:55&speed=0&head=0
```

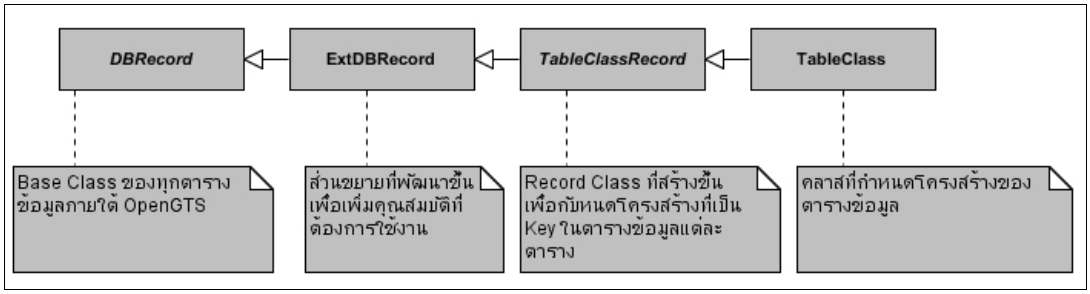
ผลลัพธ์จากการทดสอบการตั้งค่าเบื้องต้นแสดงผลผ่าน Web UI ได้ดังนี้



รูปที่ 3.21 แสดงหน้าจอและผลการทำงานของ OpenGTS

3.6.2 เขียนโปรแกรม (Coding)

เราสามารถใช้ base class ที่ OpenGTS Framework จัดเตรียมไว้เพื่อทำการสร้างคลาสที่เป็นโครงสร้างตารางข้อมูลเพิ่มเติมเพื่อรองรับการทำงานด้านการขนส่งสินค้าของเฟรมเวิร์คที่จะทำการสร้างขึ้น โดยในงานวิจัยชิ้นนี้ ต้องทำการสร้างตารางขึ้นมาใหม่จำนวน 6 ตารางด้วยกันคือตาราง Item , Cargo, Sender, Receiver, Shipment, CargoItem โดยการสร้างตารางต้องทำการ coding เพื่อกำหนดโครงสร้างขึ้นแล้วทำการ run ระบบเพื่อทำการติดตั้งตารางข้อมูลลงฐานข้อมูล รูปแบบการอิมพลีเมนต์สามารถแสดงตัวอย่างเป็น class diagram ได้ดังนี้



รูปที่ 3.22 แสดงรูปแบบการอิมพลีเมนต์ Table Class

หลังจากอิมพลีเมนต์ table class แล้วจึงทำการ Coding ในส่วนของ Module อื่นๆ ต่อไป เพื่อสร้างเซอร์วิสตามที่ได้ออกแบบไว้ในส่วน SystemDesign ซึ่งในกรอบงานเชิงบริการสำหรับการติดตามห่วงโซอุปทานในการขนส่ง จะมีลักษณะที่เป็น OO Framework ซึ่งประกอบด้วยส่วนต่างๆ ตามที่ออกแบบไว้ใส่วนของการ Design ซึ่งมี Hook และ Slot ที่นักพัฒนาระบบที่ต้องการนำไปอิมพลีเมนต์เป็นระบบจะต้องทำความเข้าใจหลักเกณฑ์และวิธีการดังนี้

นักพัฒนาระบบต้องทำการอิมพลีเมนต์คลาสและอินเตอร์เฟสต่อไปนี้ เพื่อให้เฟรมเวิร์คสามารถทำงานได้

1. AbastractItemAdapter, AbstractCargoAdapter, AbstractShipmentAdapter, AbstractSenderAdapter, AbstractReceiverAdapter

ซึ่งเป็น Abstract Class ที่ประกาศ method และ properties เพื่อทำหน้าที่เป็น adapter ที่เชื่อมต่อข้อมูลระหว่างภายนอกกับตารางที่สร้างขึ้นภายในระบบ

2. IDBViewMapper

ซึ่งเป็น Interface หลักของเฟรมเวิร์คนี้ที่ประกาศ method ที่จำเป็นในการทำการจับคู่ข้อมูลระหว่างภายนอกและภายในระบบให้สามารถสื่อสารกันได้ ดังตัวอย่างการอิมพลีเมนต์ได้ดังนี้

```
package org.opengts.custom.db.views;
import org.opengts.custom.db.tables.*;
import org.opengts.db.tables.*;
publicclassDefaultMapperimplements IDBViewMapper {
        @Override
        public Item mapInputData(ItemInfo inf, Item dbinf) {
                ItemAdapter adpt = newItemAdapter(dbinf);
                adpt.setName(inf.getName());
                adpt.setDesc(inf.getDesc());
                adpt.setBaseUnit(inf.getBaseUnit());
                adpt.setWeight(inf.getWeight());
```

```
                return adpt.getRecord();
        }
@Override
        public Cargo mapInputData(CargoInfo inf, Cargo dbinf) {
                // TODO Auto-generated method stub
                CargoAdapter adpt = newCargoAdapter(dbinf);
                adpt.setName(inf.getName());
                adpt.setBaseUnit(inf.getBaseUnit());
                adpt.setDesc(inf.getDesc());
                adpt.setLoadTimestamp(inf.getLoadTimestamp());
                adpt.setUnloadTimestamp(inf.getUnloadTimestamp());
                adpt.setWeight(inf.getWeight());
                return adpt.getRecord();
        }
…
}
```

3. AbsractBusinessDataImportService, AbtractPackageManagementService, AbstractShipmentManagementService, AbstractGeoZoneInfoSerice, AbstractGPSEventInfoService, AbstractUtilityService

ซึ่งเป็น Abstract Class ซึ่งบรรจุ Service Operation ไว้พร้อมสำหรับการทำงานซึ่งการที่จะนำไปใช้ได้นั้นต้องทำการอิมพลีเมนต์ slot เพื่อกำหนดค่าเริ่มต้นของ DBViewMapper ที่จะใช้ในการจับคู่ข้อมูลก่อน ซึ่ง DBViewMapper ที่จะนำมากำหนดค่าให้ได้นั้นต้องสืบทอดคุณสมบัติ จาก IDBViewMapper Interface เท่านั้น ดังตัวอย่างต่อไปนี้

```
package org.opengts.custom.services;
import org.opengts.custom.db.views.DefaultMapper;
publicclass BusinessDataImportService extends
AbstractBusinessDataImportService{
        @Override
        protectedvoid createMapper() {
                super.mapper = new DefaultMapper();
        }

}
```

เมื่อทำการอิมพลีเมนต์ครบทุกส่วนที่กำหนดไว้เรียบร้อย สามารถสร้างเว็บเซอร์วิสด้วย Apache CXF 2.5.2 ที่จะเชื่อมโยงกับระบบอื่นๆ ในกระบวนการโลจิสติกส์และการบริหารงานห่วงโซ่อุปทานเพื่อให้บริการเว็บเซอร์วิสต่อไป

บทที่ 4
การทดสอบและสรุปผล

ในขั้นตอนการทดสอบ เป็นการทดสอบโดยอาศัยข้อมูลจากการเก็บข้อมูลของอุปกรณ์ ซึ่งเป็นโทรศัพท์มือถือ (Smart phone) ซึ่งมีการติดตั้งอุปกรณ์รับสัญญาณ GPS

4.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

ผู้วิจัยได้ใช้เครื่องคอมพิวเตอร์โน้ตบุ๊กของตนเองในการทดสอบครั้งนี้ โดยมีเครื่องมือและองค์ประกอบในการพัฒนาดังนี้

4.1.1 ฮาร์ดแวร์ (Hardware)

- คอมพิวเตอร์โน้ตบุค CPU Core i7 740M 1.73 GHz
- หน่วยความจำ 4 Gigabyte
- ฮาร์ดดิส 500 Gigabyte

4.1.2 ซอฟต์แวร์ (Software)

- ระบบปฏิบัติการ Windows 7 64bit
- platform JAVA JDK 6 , ASP.NET 3.5
- Web Server Apache tomcat 6, IIS 7
- Eclipse Helios for java EE IDE, Visual Studio 2008 Professional
- Web service Framework Apache CXF 2.5.2
- Web Browser FireFox, Google Chrome, Microsoft Internet Explorer
- Mysql 5.0.51b

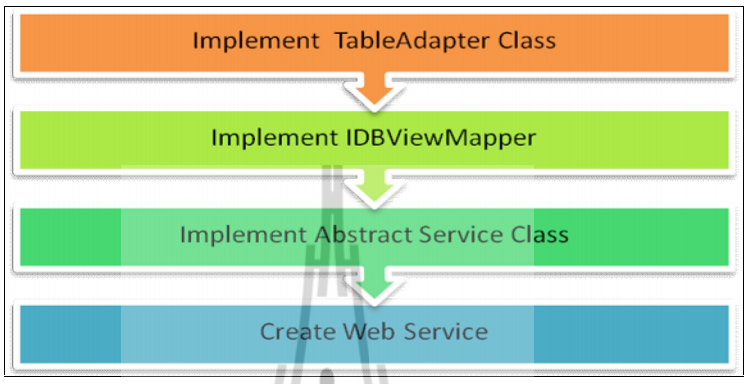
4.1.3 ส่วนประกอบอื่นๆ (Component)

- Google map Control for ASP.NET v 4.1

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.2 การเตรียมการทดสอบ

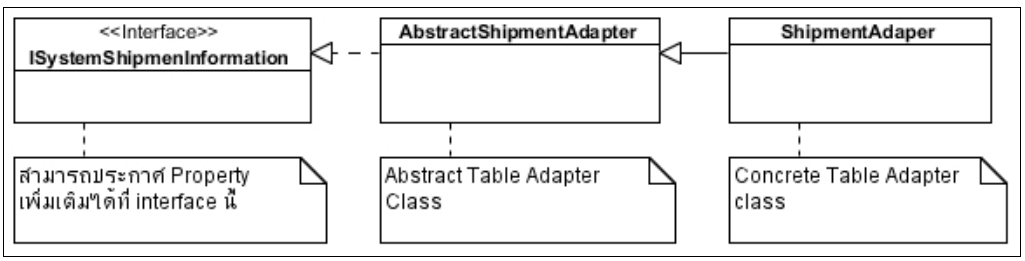
ในขั้นตอนนี้ ผู้วิจัยได้ทดลองอิมพลีเมนต์เว็บเซอร์วิสจากเฟรมเวิร์คโดยใช้คุณสมบัติมาตรฐานที่เฟรมเวิร์คจัดเตรียมไว้ และ ทำการสร้างเว็บเซอร์วิสโดยใช้ Apache CXF 2.5.2 ซึ่งมีขั้นตอนการอิมพลีเมนต์สามารถแสดงเป็นแผนภาพดังนี้



รูปที่ 4.1 แสดงขั้นตอนการอิมพลีเมนต์ SBFSTT Framework

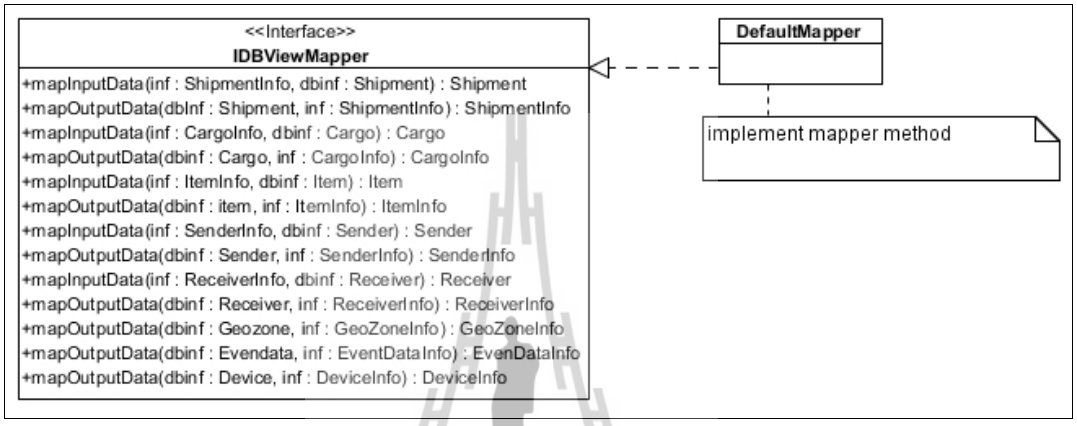
จากรูป 4.1 สามารถอธิบายแยกเป็นขั้นตอนดังนี้

4.2.1 อิมพลีเมนต์ TableAdapter Class ขั้นตอนนี้เป็นขั้นตอนแรกของการนำ SBFSTT Framework ไปใช้งาน ซึ่งนักพัฒนาซอฟต์แวร์จำเป็นที่จะต้องอิมพลีเมนต์ AbstractTableAdapter class ซึ่งเดิมจะอยู่ในรูปแบบของ abstract class ให้เป็น concrete class ก่อนที่จะดำเนินการในส่วนอื่น โดยในส่วนนี้ นักพัฒนาซอฟต์แวร์สามารถสร้างคุณสมบัติ (property) ของข้อมูล Shipment, Cargo, Item , Sender , Receiver เพิ่มเติมได้ โดยจะต้องไปทำการเพิ่มในส่วนของ IInformation interface ของแต่ละclass ซึ่งอธิบายเป็น class diagram ได้ดังนี้



รูปที่ 4.2 แสดงการอิมพลีเมนต์ TableAdapter class

4.2.2 อิมพลีเมนต์ IDBViewMapper Interface ขั้นตอนนี้เป็นการอิมพลีเมนต์ Mapper class ซึ่ง Mapper class ถือเป็นส่วนประกอบหลักของ SBFSTT Framework ซึ่งจะทำหน้าที่เป็นตัวกลางที่จะเชื่อมต่อข้อมูลเพื่อให้สามารถทำการนำเข้าและส่งออกข้อมูลจากระบบได้ โดย SBFSTT Framework ได้จัดเตรียม Interface ไว้เพื่อให้นักพัฒนาซอฟต์แวร์ สามารถนำไปอิมพลีเมนต์เป็น mapper class ได้ ซึ่งอธิบายเป็น class diagram ได้ดังนี้



รูปที่ 4.3 แสดงการอิมพลีเมนต์ table adapter class

ตัวอย่างการอิมพลีเมนต์ IDBViewMapper Interface

```
package org.opengts.custom.db.views;
import org.opengts.custom.db.tables.*;
import org.opengts.db.tables.*;

public class DefaultMapper implements IDBViewMapper {
        @Override
        public Item mapInputData(ItemInfo inf, Item dbinf) {
                ItemAdapter adpt = new ItemAdapter(dbinf);
                adpt.setName(inf.getName());
                adpt.setDesc(inf.getDesc());
                adpt.setBaseUnit(inf.getBaseUnit());
                adpt.setWeight(inf.getWeight());
                return adpt.getRecord();
        }
@Override
        public Cargo mapInputData(CargoInfo inf, Cargo dbinf) {
                CargoAdapter adpt = new CargoAdapter(dbinf);
                adpt.setName(inf.getName());
                adpt.setBaseUnit(inf.getBaseUnit());
                adpt.setDesc(inf.getDesc());
```

```
                adpt.setLoadTimestamp(inf.getLoadTimestamp());
                adpt.setUnloadTimestamp(inf.getUnloadTimestamp());
                adpt.setWeight(inf.getWeight());
                return adpt.getRecord();
        }
}
```

4.2.3 อิมพลีเมนต์ Abstract Service Class

ในขั้นตอนนี้ นักพัฒนาซอฟต์แวร์ จะต้องทำการอิมพลีเมนต์ Abstract Service Class ซึ่งมีทั้งหมด 6 Class ด้วยกันคือ

1. AbsractBusinessDataImportService,
2. AbtractPackageManagementService,
3. AbstractShipmentManagementService,
4. AbstractGeoZoneInfoSerice,
5. AbstractGPSEventInfoService,
6. AbstractUtilityService

ซึ่งการอิมพลีเมนต์นี้ จะมี slot ที่สำคัญที่จำเป็นต้องอิมพลีเมนต์เพื่อให้เซอร์วิสทำงานได้ถูกต้อง คือ protected abstract void createMapper(); ซึ่งมีตัวอย่างการอิมพลีเมนต์ ดังนี้

ตัวอย่างการอิมพลีเมนต์ createMapper slot

```
package org.opengts.custom.services;

import org.opengts.custom.db.views.DefaultMapper;

public class BusinessDataImportService extends
AbstractBusinessDataImportService{
        @Override
        protected void createMapper() {
                super.mapper = new DefaultMapper();
        }

}
```

4.2.4 Create Web Service ในขั้นตอนนี้ ผู้พัฒนาได้ทดสอบโดยใช้ Apache CXF 2.5.2 เป็น Service Framework โดยทำการ สร้างเว็บเซอร์วิสจาก Concrete Service Class ซึ่งได้จากการอิมพลีเมนต์ SBFSTT Framework ซึ่งได้เซอร์วิสทั้งหมด 6 เซอร์วิส ดังแสดงเป็นตารางได้ดังนี้

ตารางที่ 4.1 แสดงเซอร์วิส และรายละเอียดของเซอร์วิสของ SBFSTT Framework

IBusinessDataImportServiceSEI listAllItem removeItem listCargoByStatus editImportedReceiver getReceiverByID isItemExisits importItem isSenderExisits isCargoExisits getCargoByID editImportedCargo importSender importReceiver editImportedItem getSenderByID listAllReceiver isReceiverExisits listAllSender removeReceiver getItemByID importCargo removeSender listAllCargo editImportedSender removeCargo	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/BusinessDataImportServicePort WSDL : {http://services.custom.opengts.org/}BusinessDataImportServiceService Target namespace: http://services.custom.opengts.org/
IGPSEventInfoServiceSEI getLastActivePosition getEventsByCargo getDeviceEventsByTime getEventsByShipment	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/GPSEventInfoServicePort WSDL : {http://services.custom.opengts.org/}GPSEventInfoServiceService Target namespace: http://services.custom.opengts.org/
IGeoZoneInfoServiceSEI getAllGeoZoneIDs listAllGeoZone getGeoZone containPoint	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/GeoZoneInfoServicePort WSDL : {http://services.custom.opengts.org/}GeoZoneInfoServiceService Target namespace: http://services.custom.opengts.org/
IPackageManagementServiceSEI removeItemFromCargo	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/PackageManagementServicePort

isCargoItemExisits listCargoItemByCargo listCargoItems addCargoItem listItemCargos	WSDL : {http://services.custom.opengts.org/}PackageManagementServiceService Target namespace: http://services.custom.opengts.org/
IShipmentManagementServiceSEI getShipmentByID createNewShipment updateShipment listAllShipment listShipmentByStatus listShipmentCargos listSenderShipments removeShipment listReceiverShipments listDeviceShipments editShipment getShipmentByCargo	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/ShipmentManagementServicePort WSDL : {http://services.custom.opengts.org/}ShipmentManagementServiceService Target namespace: http://services.custom.opengts.org/
UtilityServiceSEI getDeviceIDForAccount listAllDeviceForAccount getDeviceByID	Endpoint address: http://localhost:9090/SBFSTTCXFService/services/SystemUtilServicePort WSDL : {http://services.custom.opengts.org/}SystemUtilServiceService Target namespace: http://services.custom.opengts.org/

เมื่อได้ทำการอิมพลีเมนต์ SBFSTT Framework แล้ว ผู้วิจัยได้นำเซอร์วิสทั้งหมดมาทำการทดสอบโดยเทคนิคการทำ Service Reference และ Unit Test ในแพลตฟอร์ม ASP.NET โดยใช้ Visual Studio 2008 professional Unit test project เป็นเครื่องมือในการทดสอบ ซึ่งจะอธิบายในส่วนของขั้นตอนการทดสอบต่อไป

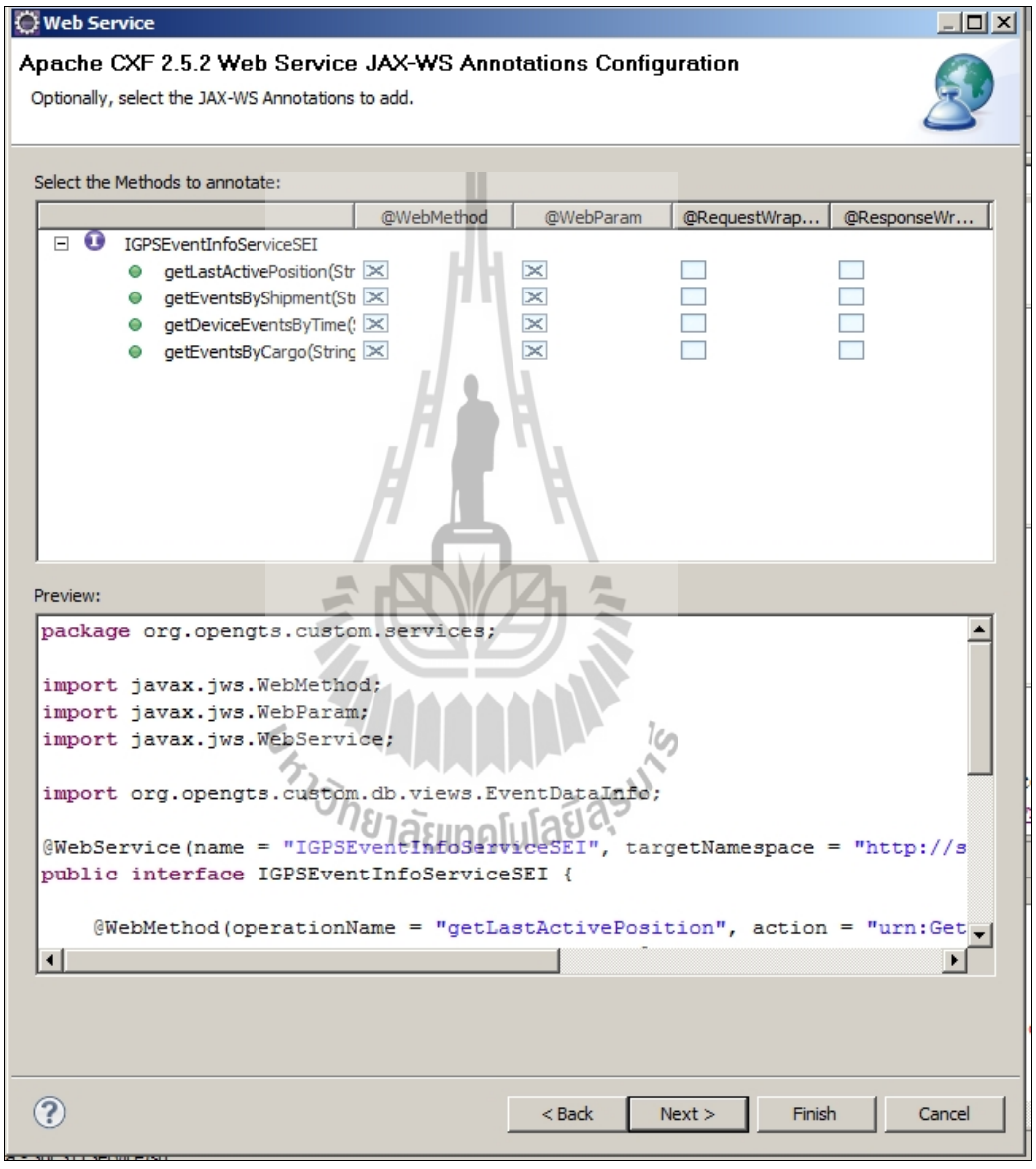
4.3 ขั้นตอนในการทดสอบ

การทดสอบแบ่งเป็น 3 รูปแบบดังนี้

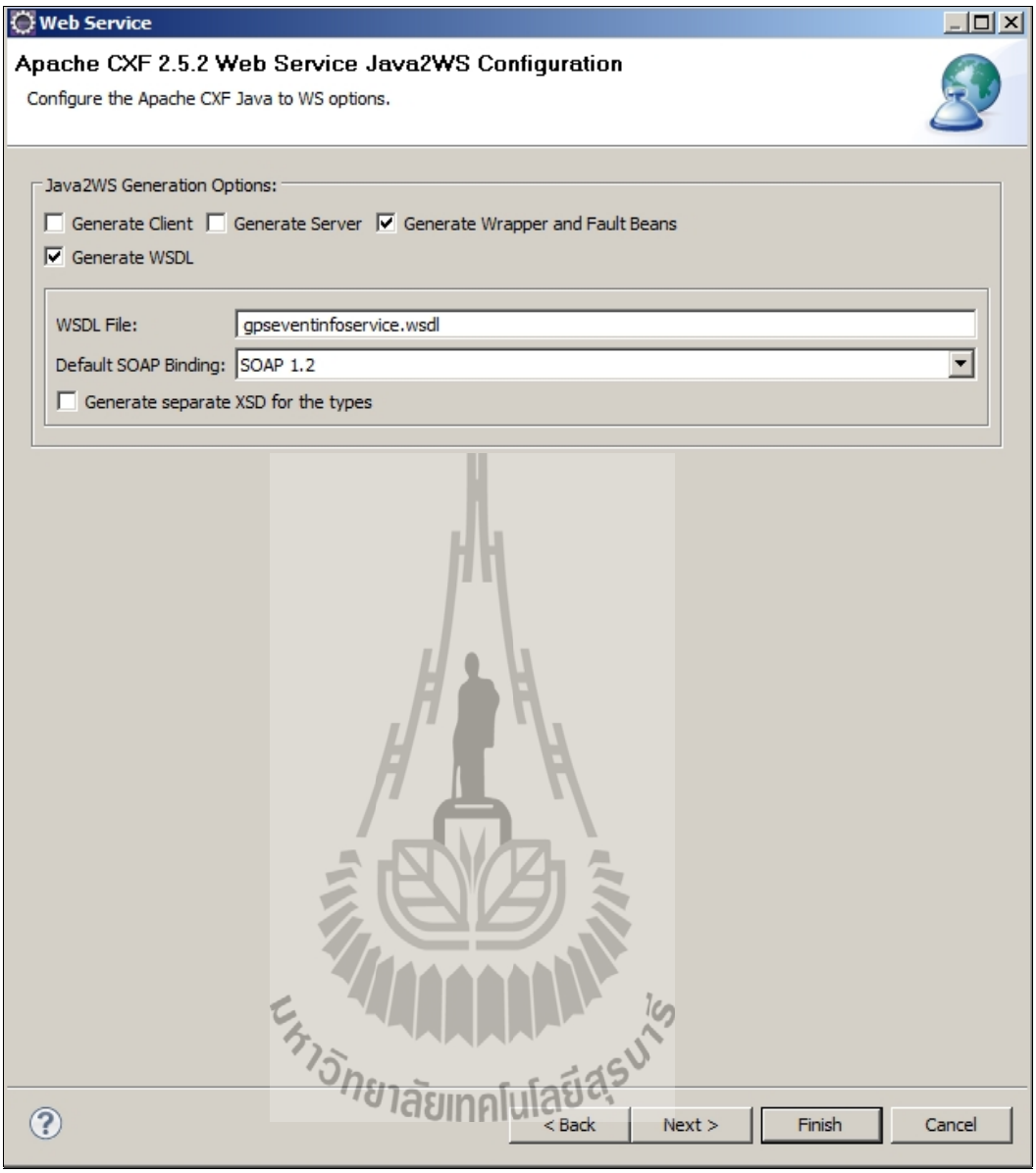
1. การทดสอบส่วนอธิบายบริการ(Service Description)โดยการทำ .NET Service Reference
2. การทดสอบความถูกต้องของการทำงานของ Service Operation ด้วยการทำ Unit Testing
3. การนำ SBFSTT Framework มาสร้างระบบตัวอย่าง

4.3.1 การการทดสอบส่วนอธิบายบริการโดยการทำ .NET Service Reference

ในการทดสอบนี้ ผู้วิจัยได้ทำการ สร้างเว็บเซอร์วิสด้วย Apache CXF 2.5.2 มีการใช้ Service Endpoint Interface และกำหนดให้ใช้มาตรฐานของ SOAP 1.2 ดังแสดงในรูปที่ 4.4 และ 4.5



รูปที่ 4.4 แสดงการสร้างเว็บเซอร์วิสซึ่งมีการประกาศ Service Endpoint Interface เพื่อเป็น API สำหรับเข้าถึง Method ในเซอร์วิส



รูปที่ 4.5 แสดงการกำหนดมาตรฐานให้ใช้ Soap Binding เป็น SOAP 1.2

ซึ่งเมื่อเสร็จสิ้นกระบวนการ จะได้เว็บเซอร์วิสซึ่งมีตัวอย่างเอกสาร WSDL แสดงได้ดังนี้

ตัวอย่างเอกสาร WSDL ของ gpseventinfoservice.wsdl

```
 <?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions name="GPSEventInfoServiceService"
    targetNamespace="http://services.custom.opengts.org/"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:tns="http://services.custom.opengts.org/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<wsdl:types>
```

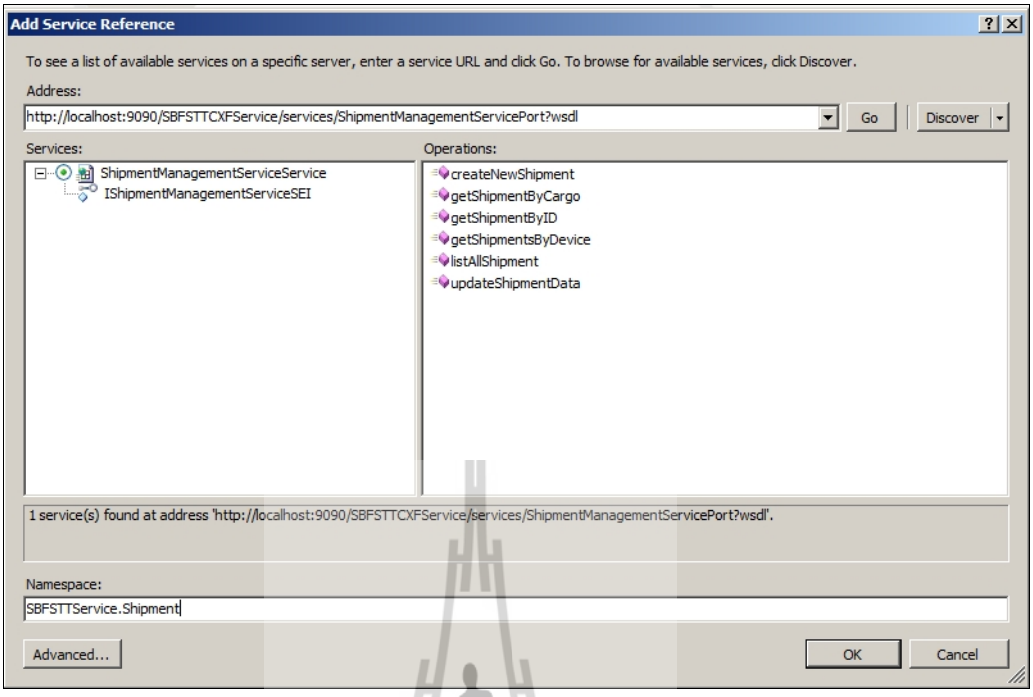
```
<xs:schema targetNamespace="http://services.custom.opengts.org/" version="1.0"
    xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:tns="http://services.custom.opengts.org/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xs:element name="getDeviceEventsByTime" type="tns:getDeviceEventsByTime" />
<xs:element name="getDeviceEventsByTimeResponse"
    type="tns:getDeviceEventsByTimeResponse" />
<xs:element name="getEventsByCargo" type="tns:getEventsByCargo" />
<xs:element name="getEventsByCargoResponse" type="tns:getEventsByCargoResponse" />
<xs:element name="getEventsByShipment" type="tns:getEventsByShipment" />
<xs:element name="getEventsByShipmentResponse" type="tns:getEventsByShipmentResponse"
    />
<xs:element name="getLastActivePosition" type="tns:getLastActivePosition" />
<xs:element name="getLastActivePositionResponse" type="tns:getLastActivePositionResponse"
    />
<xs:complexType name="getLastActivePosition">
<xs:sequence>
<xs:element minOccurs="0" name="accID" type="xs:string" />
<xs:element minOccurs="0" name="deviceID" type="xs:string" />
<xs:element name="timeStamp" type="xs:long" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getLastActivePositionResponse">
<xs:sequence>
<xs:element minOccurs="0" name="return" type="tns:eventDataInfo" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="eventDataInfo">
<xs:sequence>
<xs:element minOccurs="0" name="accountID" type="xs:string" />
<xs:element minOccurs="0" name="address" type="xs:string" />
<xs:element name="altitude" type="xs:double" />
<xs:element minOccurs="0" name="dataSource" type="xs:string" />
<xs:element minOccurs="0" name="deviceID" type="xs:string" />
<xs:element name="gpsAge" type="xs:long" />
<xs:element name="heading" type="xs:double" />
<xs:element name="latitude" type="xs:double" />
<xs:element name="longitude" type="xs:double" />
<xs:element name="speedKPH" type="xs:double" />
<xs:element name="statusCode" type="xs:long" />
<xs:element name="timestamp" type="xs:long" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getEventsByCargo">
<xs:sequence>
<xs:element minOccurs="0" name="cargoID" type="xs:string" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getEventsByCargoResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="return" type="tns:eventDataInfo" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getDeviceEventsByTime">
<xs:sequence>
<xs:element minOccurs="0" name="deviceID" type="xs:string" />
<xs:element minOccurs="0" name="accID" type="xs:string" />
<xs:element name="startTimestamp" type="xs:long" />
<xs:element name="endTimestamp" type="xs:long" />
  </xs:sequence>
  </xs:complexType>
```

```
<xs:complexType name="getDeviceEventsByTimeResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="return" type="tns:eventDataInfo" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getEventsByShipment">
<xs:sequence>
<xs:element minOccurs="0" name="shipmentID" type="xs:string" />
  </xs:sequence>
  </xs:complexType>
<xs:complexType name="getEventsByShipmentResponse">
<xs:sequence>
<xs:element maxOccurs="unbounded" minOccurs="0" name="return" type="tns:eventDataInfo" />
  </xs:sequence>
  </xs:complexType>
  </xs:schema>
  </wsdl:types>
<wsdl:message name="getLastActivePosition">
<wsdl:part element="tns:getLastActivePosition" name="parameters" />
  </wsdl:message>
<wsdl:message name="getLastActivePositionResponse">
<wsdl:part element="tns:getLastActivePositionResponse" name="parameters" />
  </wsdl:message>
<wsdl:message name="getDeviceEventsByTime">
<wsdl:part element="tns:getDeviceEventsByTime" name="parameters" />
  </wsdl:message>
<wsdl:message name="getEventsByShipmentResponse">
<wsdl:part element="tns:getEventsByShipmentResponse" name="parameters" />
  </wsdl:message>
<wsdl:message name="getEventsByShipment">
<wsdl:part element="tns:getEventsByShipment" name="parameters" />
  </wsdl:message>
<wsdl:message name="getEventsByCargoResponse">
<wsdl:part element="tns:getEventsByCargoResponse" name="parameters" />
  </wsdl:message>
<wsdl:message name="getDeviceEventsByTimeResponse">
<wsdl:part element="tns:getDeviceEventsByTimeResponse" name="parameters" />
  </wsdl:message>
<wsdl:message name="getEventsByCargo">
<wsdl:part element="tns:getEventsByCargo" name="parameters" />
  </wsdl:message>
<wsdl:portType name="IGPSEventInfoServiceSEI">
<wsdl:operation name="getLastActivePosition">
<wsdl:input message="tns:getLastActivePosition" name="getLastActivePosition" />
<wsdl:output message="tns:getLastActivePositionResponse"
   name="getLastActivePositionResponse" />
  </wsdl:operation>
<wsdl:operation name="getEventsByCargo">
<wsdl:input message="tns:getEventsByCargo" name="getEventsByCargo" />
<wsdl:output message="tns:getEventsByCargoResponse" name="getEventsByCargoResponse"
   />
  </wsdl:operation>
<wsdl:operation name="getDeviceEventsByTime">
<wsdl:input message="tns:getDeviceEventsByTime" name="getDeviceEventsByTime" />
<wsdl:output message="tns:getDeviceEventsByTimeResponse"
   name="getDeviceEventsByTimeResponse" />
  </wsdl:operation>
<wsdl:operation name="getEventsByShipment">
<wsdl:input message="tns:getEventsByShipment" name="getEventsByShipment" />
<wsdl:output message="tns:getEventsByShipmentResponse"
   name="getEventsByShipmentResponse" />
  </wsdl:operation>
  </wsdl:portType>
```

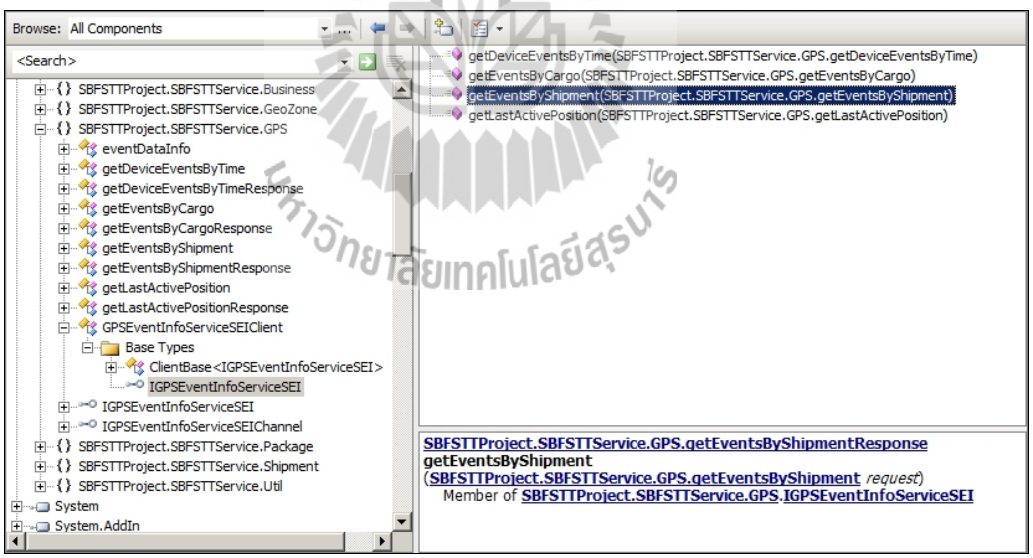
```
<wsdl:binding name="GPSEventInfoServiceServiceSoapBinding"
    type="tns:IGPSEventInfoServiceSEI">
<soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="getLastActivePosition">
<soap12:operation soapAction="urn:GetLastActivePosition" style="document" />
<wsdl:input name="getLastActivePosition">
<soap12:body use="literal" />
  </wsdl:input>
<wsdl:output name="getLastActivePositionResponse">
<soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="getEventsByCargo">
<soap12:operation soapAction="urn:GetEventsByCargo" style="document" />
<wsdl:input name="getEventsByCargo">
<soap12:body use="literal" />
  </wsdl:input>
<wsdl:output name="getEventsByCargoResponse">
<soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="getDeviceEventsByTime">
<soap12:operation soapAction="urn:GetDeviceEventsByTime" style="document" />
<wsdl:input name="getDeviceEventsByTime">
<soap12:body use="literal" />
  </wsdl:input>
<wsdl:output name="getDeviceEventsByTimeResponse">
<soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="getEventsByShipment">
<soap12:operation soapAction="urn:GetEventsByShipment" style="document" />
<wsdl:input name="getEventsByShipment">
<soap12:body use="literal" />
  </wsdl:input>
<wsdl:output name="getEventsByShipmentResponse">
<soap12:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
<wsdl:service name="GPSEventInfoServiceService">
<wsdl:port binding="tns:GPSEventInfoServiceServiceSoapBinding"
    name="GPSEventInfoServicePort">
<soap12:address
    location="http://localhost:9090/SBFSTTCXFService/services/GPSEventInfoServicePort" />
  </wsdl:port>
  </wsdl:service>
  </wsdl:definitions>
```

เอกสาร WSDL ที่ได้มาทั้งหมด 6 เซอร์วิส นั้น จะต้องสามารถถูก reference ได้ด้วย .NETService Reference

เครื่องมือที่ผู้วิจัยได้นำมาใช้ในกระบวนการทดสอบ .NET Service Reference นั้นคือ Visual Studio 2008 professional ซึ่งจะทำการแปลงเอกสาร WSDL ให้อยู่ในรูปแบบ C# class โดยมีขั้นตอนการทดสอบดังแสดงตัวอย่างในรูป 4.6 และ 4.7 ดังต่อไปนี้



รูปที่ 4.6 แสดงหน้าจอของการทำ Service Reference ด้วย visual studio 2008 Professional



รูปที่ 4.7 แสดง Object Browser ของ Visual Studio 2008 ซึ่งแสดงให้เห็น Service method ที่ได้ทำการ Reference จากเซอร์วิส

ตัวอย่างบางส่วนของ XML Configuration (app.config) ซึ่งแสดง Endpoint address ของเซอร์วิสแต่ละตัว

```
<client>
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/BusinessDataImportS
ervicePort"
                binding="customBinding"
bindingConfiguration="BusinessDataImportServiceServiceSoapBinding"

contract="SBFSTTService.Business.IBusinessDataImportServiceSEI"
                name="BusinessDataImportServicePort" />
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/GPSEventInfoService
Port"
                binding="customBinding"
bindingConfiguration="GPSEventInfoServiceServiceSoapBinding"
                contract="SBFSTTService.GPS.IGPSEventInfoServiceSEI"
name="GPSEventInfoServicePort" />
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/GeoZoneInfoServiceP
ort"
                binding="customBinding"
bindingConfiguration="GeoZoneInfoServiceServiceSoapBinding"
                contract="SBFSTTService.GeoZone.IGeoZoneInfoServiceSEI"
name="GeoZoneInfoServicePort" />
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/PackageManagementSe
rvicePort"
                binding="customBinding"
bindingConfiguration="PackageManagementServiceServiceSoapBinding"

contract="SBFSTTService.Package.IPackageManagementServiceSEI"
                name="PackageManagementServicePort" />
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/SystemUtilServicePo
rt"
                binding="customBinding"
bindingConfiguration="SystemUtilServiceServiceSoapBinding"
                contract="SBFSTTService.Util.UtilityServiceSEI"
name="SystemUtilServicePort" />
            <endpoint
address="http://localhost:9090/SBFSTTCXFService/services/ShipmentManagementS
ervicePort"
                binding="customBinding"
bindingConfiguration="ShipmentManagementServiceServiceSoapBinding"

contract="SBFSTTService.Shipment.IShipmentManagementServiceSEI"
                name="ShipmentManagementServicePort" />
</client>
```

จากการทดสอบพบว่าไม่พบข้อผิดพลาดใดๆ ในการทำ .NET Service Reference ซึ่ง Visual Studio 2008 สามารถทำ Service Reference จาก Web Service Class ที่อิมพลีเมนต์ ด้วยภาษา Java มาสู่ .NET Component ได้อย่างสมบูรณ์ทั้งในส่วนของ Service Operation และส่วนของ Service Parameter และสามารถแสดงผลเป็น Object ใน Object Browser ได้อย่างถูกต้องในทุกเซอร์วิส

4.3.2 การทดสอบความถูกต้องของการทำงานของ Service Operation ด้วยการทำ Unit Testing

ในการทดสอบนี้ ผู้วิจัยได้เลือกใช้ Unit Testing Project ของ Visual Studio 2008 Professional โดยจะทำการสร้าง Project ขึ้นเพื่อทดสอบในทุก Service Operation ในขั้นตอนของการทดสอบ โดยมีขั้นตอนในการทดสอบดังนี้

1. จำลองข้อมูลสำหรับการทดสอบ

ผู้วิจัยได้ทำการจำลองข้อมูลขึ้นมาเพื่อเป็นข้อมูลตัวอย่างสำหรับป้อนเข้าระบบในกา
ทดสอบซึ่งสามารถแสดงในรูปแบบของตารางได้ดังนี้

ตารางที่ 4.2 แสดงข้อมูล ItemInfo สำหรับกำหนดให้ตัวแปรใน Unit Test

ID	name	baseUnit	weight	desc
IX01	Test Item	box	5	Item Description

ตารางที่ 4.3 แสดงข้อมูล CargoInfo สำหรับกำหนดให้ตัวแปรใน Unit Test

ID	name	shipmentID	baseUnit	weight	loadTimestamp	unloadTimestamp	desc
CX01	Test Cargo	SHX01	box	20	1329879704	1329879944	Cargo Description

ตารางที่ 4.4 แสดงข้อมูล SenderInfo สำหรับกำหนดให้ตัวแปรใน Unit Test

ID	name	phone	address	desc
SX01	Test Sender	0000000000	My Address	Sender Description

ตารางที่ 4.5 แสดงข้อมูล ReceiverInfo สำหรับกำหนดให้ตัวแปรใน Unit Test

ID	name	phone	address	desc
RX01	Test Receiver	0000000000	My Address	Receiver Description

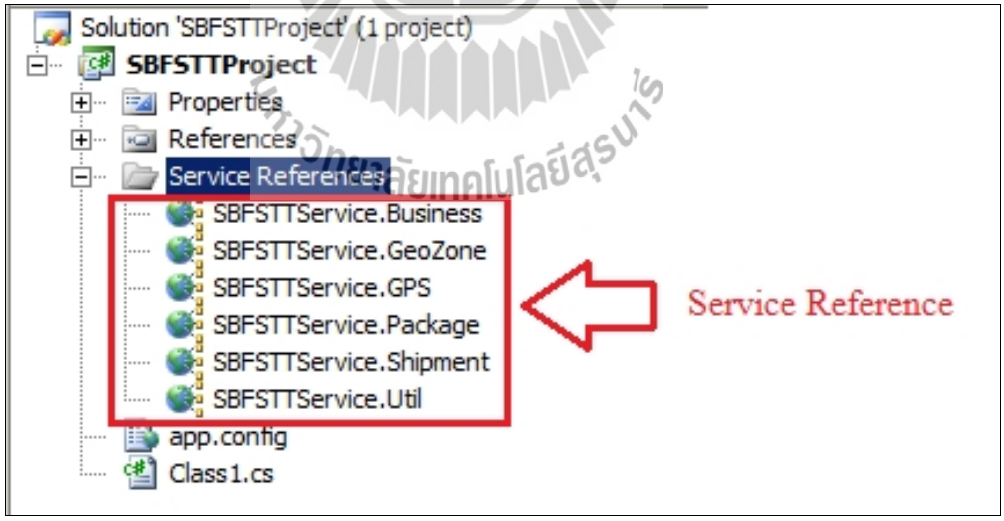
ตารางที่ 4.6 แสดงข้อมูล Cargo Item สำหรับกำหนดให้ตัวแปรใน Unit Test

CargoID	ItemID	Qty
CX01	IX01	5

ตารางที่ 4.7 แสดงข้อมูล ShipmentInfo สำหรับกำหนดให้ตัวแปรใน Unit Test

ID	SHX01
accountID	Sysadmin
deviceID	0837357404
senderID	SX01
receiverID	RX01
desc	Shipment Description
loadApprovedTimestamp	1329879704
receiveApprovedTimestamp	1329879944
departEventTimestamp	1329879704
arriveEventTimestamp	1329879944

2. ทำการสร้างโปรเจ็คใน Visual Studio และทำการ Reference Service ทุกเซอร์วิสเข้ามาในโปรเจ็คโดยใช้ขั้นตอนและวิธีการเช่นเดียวกับที่แสดงในข้อ 4.3.1



รูปที่ 4.8 แสดง C# Class library Project ซึ่งทำ Service Reference เรียบร้อยแล้ว

3. สร้าง Unit Test เพื่อทำการ Test Project ที่สร้างขึ้น ในขั้นตอนนี้ ผู้วิจัยได้ทำการสร้าง Unit Test ขึ้นโดยใช้เครื่องมือของ Visual Studio 2008 ในการสร้างโค๊ดสำหรับการทดสอบขึ้นมาเพื่อใช้สำหรับการทดสอบ Service Operation แต่ละตัว

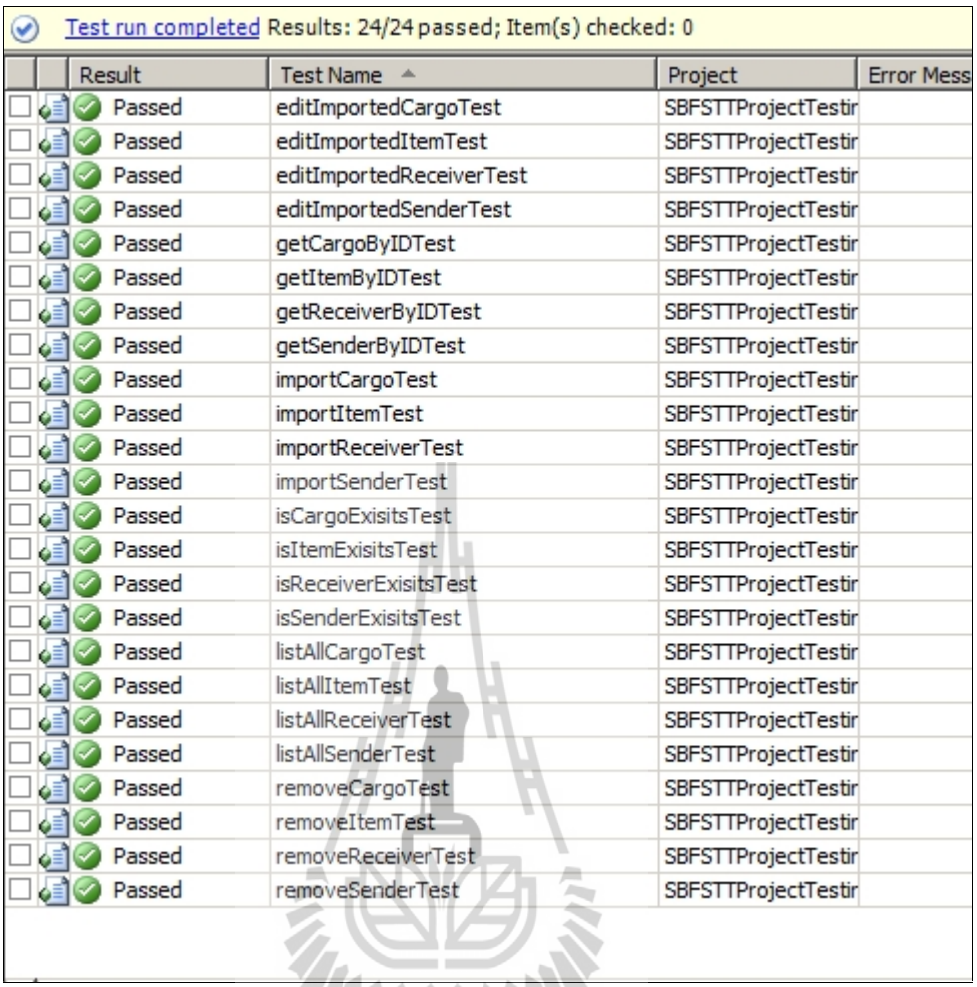
ตัวอย่างบางส่วนของซอสโค๊ด Unit Test ซึ่งเขียนขึ้นด้วยภาษา C#

```
[Priority(1), TestMethod()]
public void importCargoTest()
{
   BusinessDataImportServiceSEIClient target =
      new  BusinessDataImportServiceSEIClient();
   cargoInfo inf = new cargoInfo()
   {
      ID = "CX01",
      name = "Test Cargo",
      shipmentID = "SHX01",
      baseUnit = "box",
      weight = 20,
      loadTimestamp = 1329879704,
      unloadTimestamp = 1329879944,
      desc = "Cargo Description"
   };
   bool expected = true;
   bool actual;
   target.Open();
   actual = target.importCargo(inf);
   target.Close();
   Assert.AreEqual(expected, actual);
}


[Priority(3), TestMethod()]
public void listAllCargoTest()
{
   BusinessDataImportServiceSEIClient target =
      new BusinessDataImportServiceSEIClient();
   cargoInfo[] actual;
   target.Open();
   actual = target.listAllCargo();
   target.Close();
   Assert.IsTrue((actual.Length >= 1));
}
```

แสดงตัวอย่างข้อมูลที่ส่งกลับมาจากเว็บเซอร์วิส

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
    <soap:Body>
        <ns2:getGeoZoneResponse
        xmlns:ns2="http://services.custom.opengts.org/">
            <return>
                <accountID>sysadmin</accountID>
                <description>SUT</description>
                <displayName></displayName>
                <geozoneID>gx01</geozoneID>
                <latitude1>14.86882</latitude1>
                <latitude2>14.86769</latitude2>
                <latitude3>14.86683</latitude3>
                <latitude4>14.86592</latitude4>
                <latitude5>14.86554</latitude5>
                <latitude6>14.86563</latitude6>
                <latitude7>14.86662</latitude7>
                <latitude8>14.86846</latitude8>
                <longitude1>102.05195</longitude1>
                <longitude2>102.05286</longitude2>
                <longitude3>102.0525</longitude3>
                <longitude4>102.05184</longitude4>
                <longitude5>102.0515</longitude5>
                <longitude6>102.04973</longitude6>
                <longitude7>102.049</longitude7>
                <longitude8>102.04964</longitude8>
                <maxLatitude>14.86882</maxLatitude>
                <maxLongitude>102.05286</maxLongitude>
                <minLatitude>14.86554</minLatitude>
                <minLongitude>102.049</minLongitude>
                <radius>500</radius>
                <reverseGeoCode>true</reverseGeoCode>
                <shapeColor>#dd0000</shapeColor>
                <zoneType>3</zoneType>
            </return>
        </ns2:getGeoZoneResponse>
    </soap:Body>
</soap:Envelope>
```



Test run completed Results: 24/24 passed; Item(s) checked: 0

Result	Test Name	Project	Error Mess
Passed	editImportedCargoTest	SBFSTTProjectTestir	
Passed	editImportedItemTest	SBFSTTProjectTestir	
Passed	editImportedReceiverTest	SBFSTTProjectTestir	
Passed	editImportedSenderTest	SBFSTTProjectTestir	
Passed	getCargoByIDTest	SBFSTTProjectTestir	
Passed	getItemByIDTest	SBFSTTProjectTestir	
Passed	getReceiverByIDTest	SBFSTTProjectTestir	
Passed	getSenderByIDTest	SBFSTTProjectTestir	
Passed	importCargoTest	SBFSTTProjectTestir	
Passed	importItemTest	SBFSTTProjectTestir	
Passed	importReceiverTest	SBFSTTProjectTestir	
Passed	importSenderTest	SBFSTTProjectTestir	
Passed	isCargoExisitsTest	SBFSTTProjectTestir	
Passed	isItemExisitsTest	SBFSTTProjectTestir	
Passed	isReceiverExisitsTest	SBFSTTProjectTestir	
Passed	isSenderExisitsTest	SBFSTTProjectTestir	
Passed	listAllCargoTest	SBFSTTProjectTestir	
Passed	listAllItemTest	SBFSTTProjectTestir	
Passed	listAllReceiverTest	SBFSTTProjectTestir	
Passed	listAllSenderTest	SBFSTTProjectTestir	
Passed	removeCargoTest	SBFSTTProjectTestir	
Passed	removeItemTest	SBFSTTProjectTestir	
Passed	removeReceiverTest	SBFSTTProjectTestir	
Passed	removeSenderTest	SBFSTTProjectTestir	

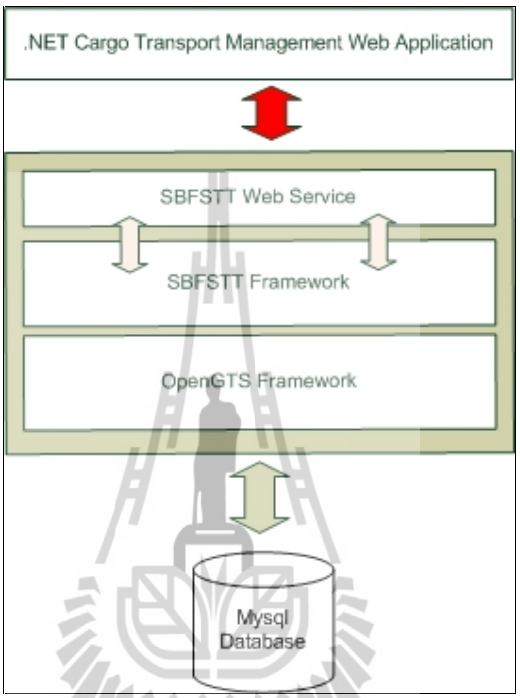
รูปที่ 4.9 แสดงตัวอย่างหน้าจอแสดงผลการทดสอบเว็บเซอร์วิสด้วย Visual Studio 2008 Unit Testing

มหาวิทยาลัยเทคโนโลยีสุรนารี

จากการทดสอบด้วย Unit Testing พบว่า Service Operations ทั้งหมดสามารถผ่านการทดสอบได้ทั้งหมด ไม่พบข้อผิดพลาดใดๆ ซึ่งจากผลการทดสอบ สามารถสรุปได้ว่าเว็บเซอร์วิสที่ถูกอิมพลีเมนต์มาจากจาก SBFSTT Framework สามารถถูกเรียกใช้และทำงานได้อย่างถูกต้องทังหมด ในทุก Service Operations ที่ทำการทดสอบ

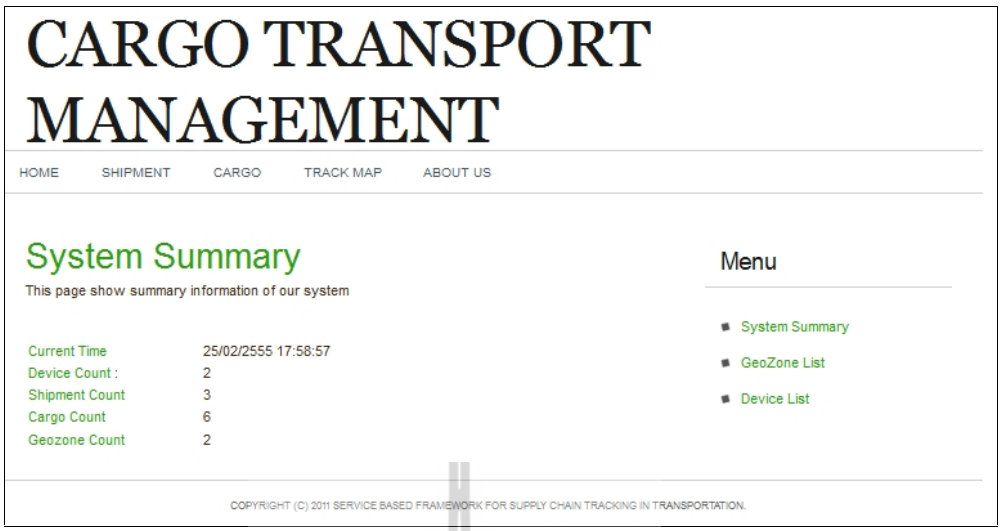
4.3.3 การนำ SBFSTT Framework มาสร้างระบบตัวอย่าง

ในการทดสอบนี้ ผู้วิจัยได้พัฒนาระบบตัวอย่างซึ่งเรียกใช้เว็บเซ วิสจาก SBFSTT Framework เพื่อประมวลผลข้อมูล และแสดงผลข้อมูลด้วย Service Client Application ซึ่งพัฒนาภายใต้เทคโนโลยี ASP.NET โดยมีการออกแบบสถาปัตยกรรมของระบบดังนี้

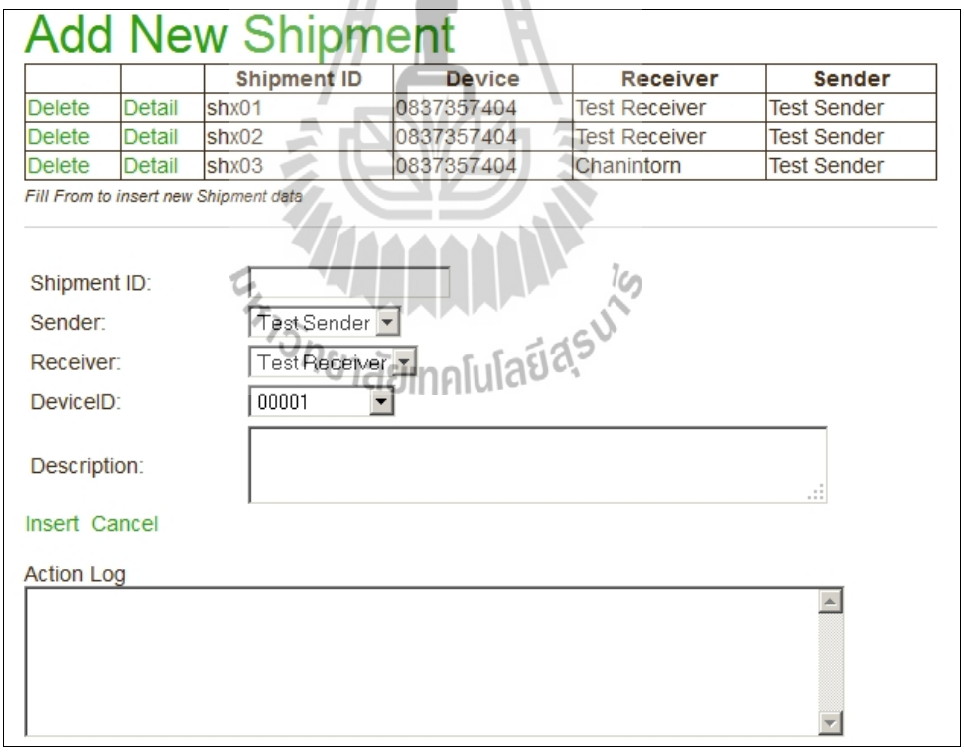


รูปที่ 4.10 แสดงสถาปัตยกรรมของระบบ Cargo Transport Management Web Application ซึ่งเป็นระบบตัวอย่างที่สร้างจากการอิมพลีเมนต์ SBFSTT Framework

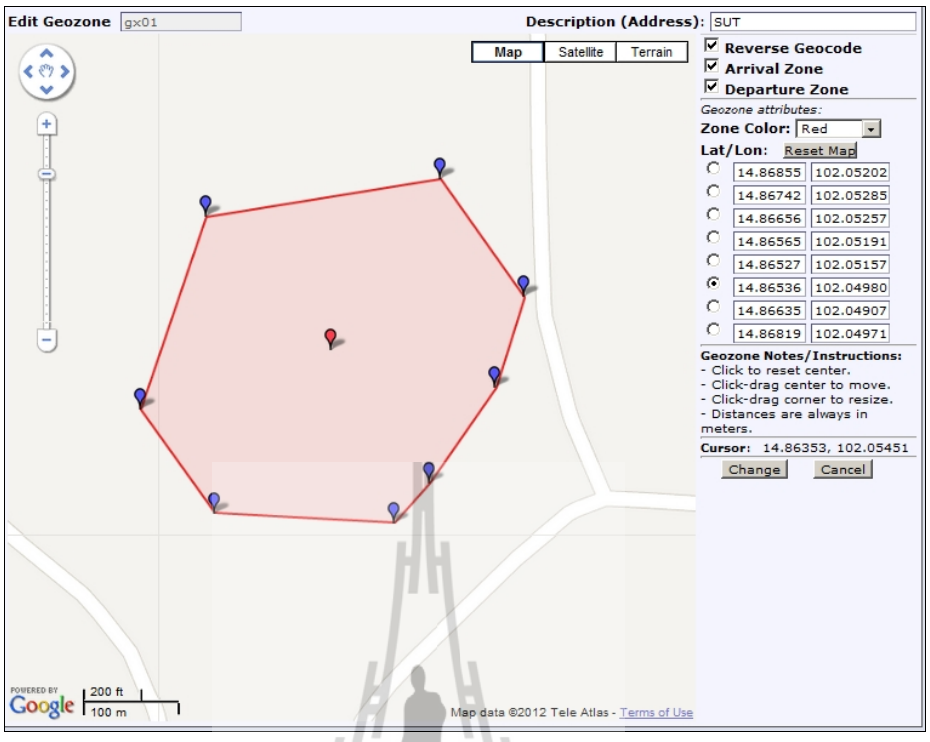
จากรูปที่ ซึ่งแสดงถึงสถาปัตยกรรมของระบบ ซึ่งจะประกอบด้วย 2 ส่วนหลักๆ ซึ่งทำงานอยู่ต่างแพลตฟอร์มกันโดยที่ Service Engine ซึ่งพัฒนาขึ้นจากการอิมพลีเมนต์ SBFSTT Framework ซึ่งอยู่ในแพลตฟอร์ม Java เป็นหัวใจหลักในการทำงานส่วนใหญ่ โดยส่วนแสดงผล (User Interface) ผู้วิจัยได้เลือกใช้เทคโนโลยีบนแพลตฟอร์ม .net คือ ASP.NET และใช้ภาษา C# ในการพัฒนาโดย ส่วนแสดงผลทำหน้าที่เพียงเรียกใช้เว็บเซอร์วิสจากฝั่ง Java เพื่อเป็นส่วนติดต่อผู้ใช้งานสำหรับกรอกข้อมูลที่จำเป็นและเรียกข้อมูลจากเว็บเซอร์วิสไปแสดงผล



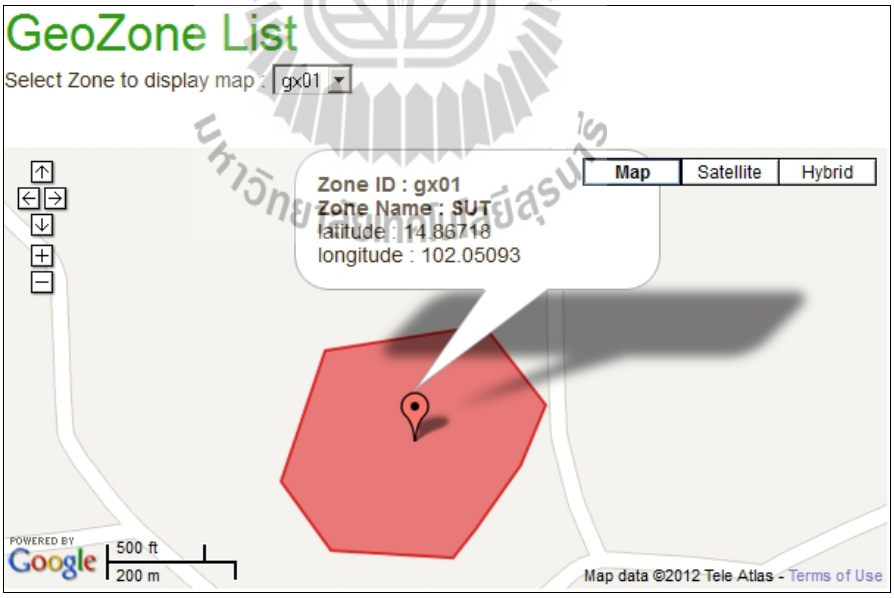
รูปที่ 4.11 แสดงหน้าแรกของตัวอย่างโปรแกรมซึ่งพัฒนาขึ้นจากเว็บเซอร์วิสที่สร้างขึ้นจาก SBFSTT Framework



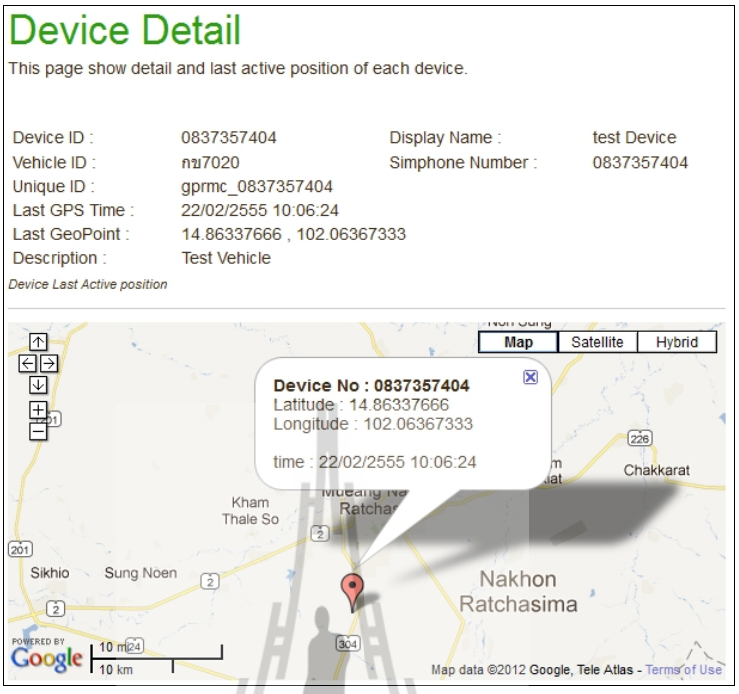
รูปที่ 4.12 แสดงหน้าจอกรอกข้อมูลซึ่งเรียกใช้เว็บเซอร์วิสจาก SBFSTT Framework



รูปที่ 4.13 แสดงการกำหนดจุดสำคัญ (Geo Zone) ด้วยระบบ OpenGTS

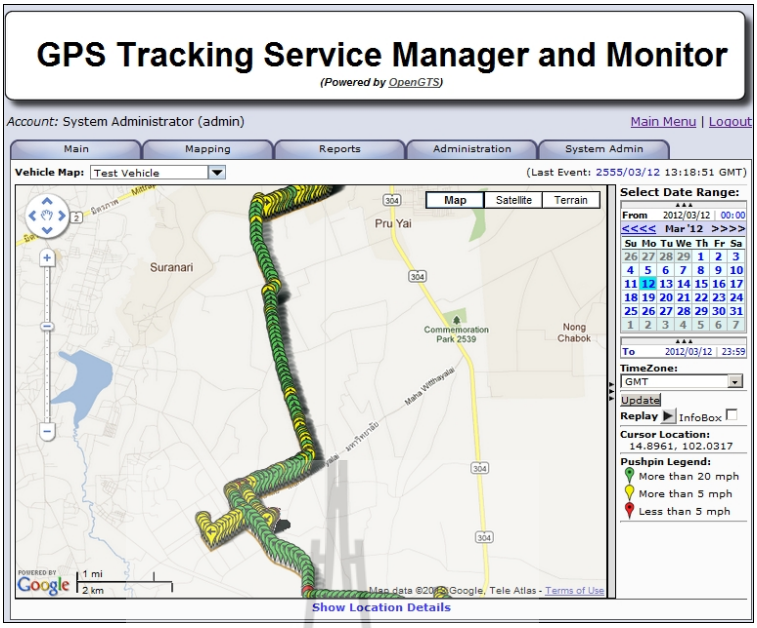


รูปที่ 4.14 แสดงตัวอย่างหน้าจอแสดงผลในส่วนของจุดสำคัญ (Geo Zone) อาศัยการดึงข้อมูลจากเซอร์วิสของ SBFSTT Framework ซึ่งสามารถนำไปใช้กำหนดจุดผ่านของการขนส่งสินค้าได้



รูปที่ 4.15 แสดงตัวอย่างหน้าจอแสดงผลของ Device ซึ่งข้อมูลที่ได้เกิดจากการดึงข้อมูลโดยผ่านเว็บเซอร์วิสของ SBFSTT Framework

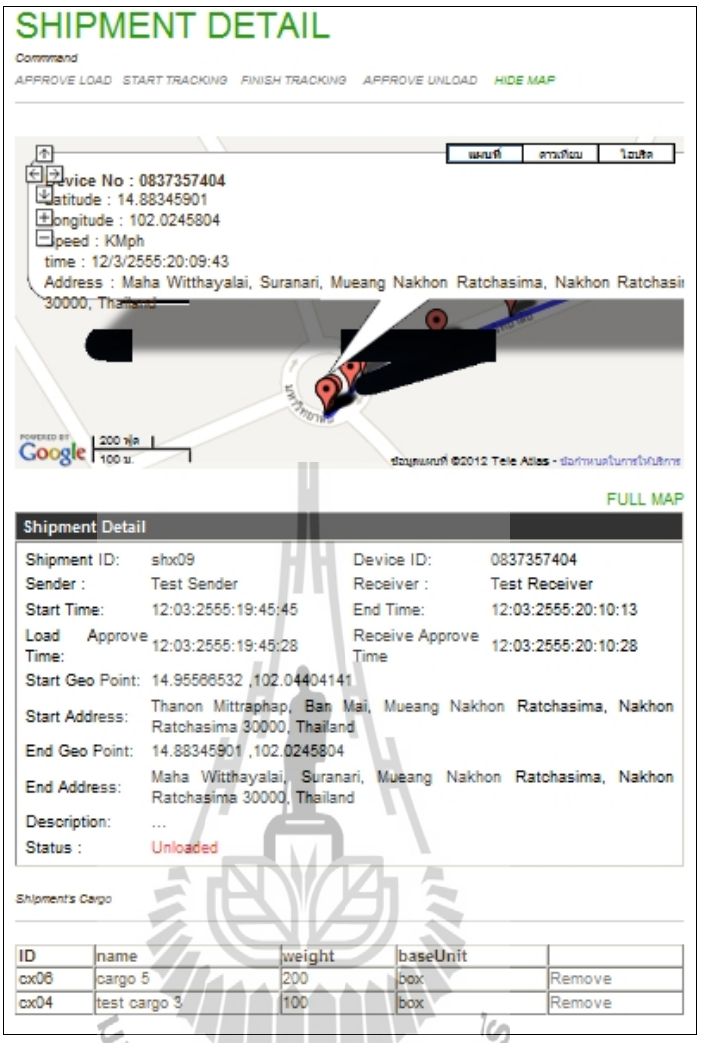
รูปที่ 4.16 แสดงหน้าจอโปรแกรมซึ่งติดตั้งบน Smart Phone ที่จะคอยรายงานตำแหน่งพิกัด GPS มายังเซิฟเวอร์



รูปที่ 4.17 แสดงหน้าจอ Track map ในส่วนของ GPS Tracking Service Manager and Monitor ซึ่งผู้วิจัยเลือกดัดแปลงระบบของ OpentGTS มาใช้โดยไม่ต้องทำการพัฒนาขึ้นใหม่

จากรูปที่ 4.17 จะสังเกตได้ว่า ข้อมูลที่ได้จากการ Track ด้วย Gps Tracking เพียงอย่างเดียว มีความซับซ้อน เข้าใจยาก และไม่สามารถบอกรายละเอียดการขนส่งได้อย่างชัดเจน

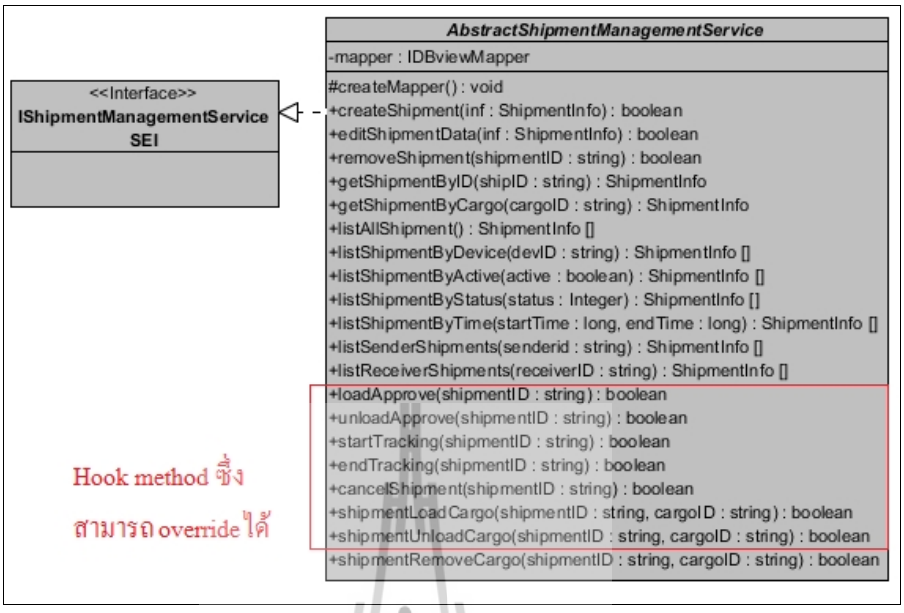
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 4.18 แสดงหน้าจอโปรแกรมแสดงผลในส่วนของการดูประวัติเส้นทางการขนส่งสินค้าซึ่งประมวลผลมาจากข้อมูลที่เก็บจาก GPS

รูปที่ 4.18 เป็นการนำข้อมูลชุดเดียวกับที่ได้ในรูปที่ 4.17 มาประมวลผลแยกตามรายการขนส่งโดยอาศัยเซอร์วิสของ SBFSTT Framework ซึ่งช่วยลดความซับซ้อนของข้อมูล แปลความหมายได้สะดวกกว่าข้อมูลดิบเพียงอย่างเดียว ซึ่งจะเอื้อประโยชน์ในการนำไปอิมพลีเมนต์เป็นระบบขนส่งสินค้าได้เป็นอย่างมาก

ในส่วนของความหลากหลายในการประยุกต์ใช้กรอบงานนั้น SBFSTT Framework ได้ถูกออกแบบให้มี interface ซึ่งสามารถทำการเขียนทับ (Override) พฤติกรรมบางพฤติกรรมซึ่งอาจมีความแตกต่างด้านการใช้งานในแต่ละรูปแบบการขนส่งซึ่งสามารถทำการเปลี่ยนแปลงพฤติกรรมได้โดยเทคนิคการ Override method ซึ่งสามารถได้ใน method ต่อไปนี้



รูปที่ 4.19 แสดง hook method ซึ่งสามารถทำการ override ได้

```
@Override
public boolean unloadApprove(String shipmentID){
      ShipmentInfo inf = super.getShipmentByID(shipmentID);
      GeoZoneInfoService geo = new GeoZoneInfoService();
      if(geo.containPoint(inf.getAccountID(),
      inf.getDestinationGeoZoneID(),
      inf.getDestinationLatitude(),
      inf.getDestinationLongitude())){
            return super.unloadApprove(shipmentID);
      }
      return false;
}
```

ตัวอย่างซอสโค๊ดซึ่งมีการ Override method unloadApprove ซึ่งมีการเปลี่ยนแปลงพฤติกรรมให้ต้องทำการตรวจสอบว่าสินค้าอยู่ในโซนที่กำหนดก่อนจึงจะทำการยืนยันการขนถ่ายได้

ผู้วิจัยได้นำเฟรมเวิร์คที่พัฒนาขึ้นไปให้กับโปรแกรมเมอร์ทดลองทำการอิมพลีเมนต์ เฟรมเวิร์คเพื่อใช้งาน โดยทำการทดสอบในส่วนของการอิมพลีเมนต์เฟรมเวิร์คเพื่อสร้างเว็บเซอร์วิส ขึ้นตามรูปแบบการพัฒนาที่วางเอาไว้โดยในการทดลองครั้งนี้ได้รับความอนุเคราะ
โปรแกรมเมอร์ 3 ท่านซึ่งเป็นบุคลากรฝ่ายพัฒนาสารสนเทศของ บริษัท ศรีสะเกษทีทีมอเตอร์ จำกัด ร่วมทำการทดสอบ โดยหลังจากอธิบายขั้นตอนและวิธีการแล้วได้ให้โปรแกรมเมอร์ทั้ง 3 ท่านทำการอิม- พลีเมนต์ เฟรมเวิร์คเพื่อสร้างเว็บเซอร์วิสขึ้น และได้เก็บข้อมูลในรูปแบบของแบบสอบถาม โดยมีรูปแบบดังนี้

แบบสอบถามโครงการวิจัย เรื่อง กรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่ง

การทดสอบประยุกต์ใช้ Framework

คำถาม			
1. เข้าใจรูปแบบการ implement framework	เข้าใจยาก	พอใช้	เข้าใจง่าย
2. เข้าใจกระบวนการแลกเปลี่ยนข้อมูลกับระบบ Web service	เข้าใจยาก	พอใช้	เข้าใจง่าย
3. ความรู้ความเข้าใจเกี่ยวกับระบบ GPS tracking และการประยุกต์ใช้งานก่อนการใช้งาน framework	มาก	ปานกลาง	น้อย
4. ความยาก/ง่ายในการประยุกต์ใช้งาน framework ในการพัฒนาระบบเมื่อเปรียบเทียบกับการพัฒนาขึ้นเองทั้งหมด	ยาก	ปานกลาง	น้อย
5. ข้อเสนอแนะ …… …… …… …… …… …… …… …… …… …… ……			

มหาวิทยาลัยเทคโนโลยีสุรนารี

ซึ่งเมื่อให้โปรแกรมเมอร์ทั้ง 3 คนทำการทดสอบและกรอกแบบสอบถามสามารถสรุปได้ดังนี้

โปรแกรมเมอร์คนที่ 1

คำถาม			
1. เข้าใจรูปแบบการ implement framework	เข้าใจยาก	/	เข้าใจง่าย
2. เข้าใจกระบวนการแลกเปลี่ยนข้อมูลกับระบบ Web service	เข้าใจยาก	พอใช้	/
3. ความรู้ความเข้าใจเกี่ยวกับระบบ GPS tracking และการประยุกต์ใช้งานก่อนการใช้งาน framework	มาก	/	น้อย
4. ความยาก/ง่ายในการประยุกต์ใช้งาน framework ในการพัฒนาระบบเมื่อเปรียบเทียบกับการพัฒนาขึ้นเองทั้งหมด	ยาก	/	น้อย
5. ข้อเสนอแนะ เป็นระบบที่น่าสนใจ และน่านำมาพัฒนาต่อเพื่อใช้งานจริงๆ แต่ก็อาจมีข้อจำกัดในเรื่องของอุปกรณ์ GPS และการเชื่อมต่อแบบ real-time อาจทำให้ระบบไม่สามารถใช้งานได้ตลอดเวลา อาจมีการเพิ่มจุดเด่น เพื่อให้ผู้ใช้งานในเชิงธุรกิจ เห็นว่ามีประโยชน์ต่อธุรกิจของเค้าได้จริง			

โปรแกรมเมอร์คนที่ 2

คำถาม			
1. เข้าใจรูปแบบการ implement framework	เข้าใจยาก	พอใช้	/
2. เข้าใจกระบวนการแลกเปลี่ยนข้อมูลกับระบบ Web service	เข้าใจยาก	พอใช้	/
3. ความรู้ความเข้าใจเกี่ยวกับระบบ GPS tracking และการประยุกต์ใช้งานก่อนการใช้งาน framework	มาก	/	น้อย
4. ความยาก/ง่ายในการประยุกต์ใช้งาน framework ในการพัฒนาระบบเมื่อเปรียบเทียบกับการพัฒนาขึ้นเองทั้งหมด	ยาก	/	น้อย

มหาวิทยาลัยเทคโนโลยีสุรนารี

5. ข้อเสนอแนะ อาจมีการเพิ่มเติมในส่วนของการสร้างเครื่องมือมาใช้ในการ generate ข้อมูลที่จำเป็นบางอย่าง เช่นข้อมูลกลุ่ม information จะช่วยให้ประหยัดเวลา และทำความเข้าใจกับ framework ได้ง่ายขึ้น

โปรแกรมเมอร์คนที่ 3

คำถาม			
1. เข้าใจรูปแบบการ implement framework	เข้าใจยาก	/	
2. เข้าใจกระบวนการแลกเปลี่ยนข้อมูลกับระบบ Web service	เข้าใจยาก	พอใช้	/
3. ความรู้ความเข้าใจเกี่ยวกับระบบ GPS tracking และการประยุกต์ใช้งานก่อนการใช้งาน framework	มาก	/	น้อย
4. ความยาก/ง่ายในการประยุกต์ใช้งาน framework ในการพัฒนาระบบเมื่อเปรียบเทียบกับการพัฒนาขึ้นเองทั้งหมด	ยาก	/	น้อย
5. ข้อเสนอแนะ เป็นระบบที่น่าสนใจ แต่ก็ยังมีข้อจำกัดในหลายๆ ด้าน อาจต้องมีการปรับปรุงใ ง่ายต่อความเข้าใจ และออกแบบวิธีการพัฒนาให้ง่ายขึ้น			

จากการที่ผู้วิจัยได้เข้าพูดคุยและสังเกตการณ์ทดสอบพร้อมทั้งสัมภาษณ์นั้น สามารถสรุปได้ดังนี้ โปรแกรมเมอร์ทั้ง 3 ท่าน มีความชำนาญในตัวภาษา java ซึ่งเป็นภาษาที่ต้องใช้ในการอิมพลีเมนต์เฟรมเวิร์คค่อนข้างน้อยและขาดความคุ้นเคยกับตัวเครื่องมือที่ใช้ในการพัฒนาระบบคือ Eclipse ทำให้ทำความเข้าใจกับการอิมพลีเมนต์ระบบได้ค่อนข้างลำบาก แต่ก็สามารถทำการอิมพลีเมนต์เฟรมเวิร์คสำเร็จในที่สุด และเมื่อได้เว็บเซอร์วิสแล้ว ผู้วิจัยได้ให้โปรแกรมเมอร์ทั้ง 3 ท่าน ทดสอบการสร้างส่วนติดต่อผู้ใช้งานแบบง่ายๆ สำหรับเรียกใช้ operation ของเว็บเซอร์วิสที่สร้างขึ้นจากตัวเฟรมเวิร์คซึ่งโปรแกรมเมอร์ทั้ง 3 ท่านสามารถทำได้อย่างรวดเร็วเนื่องจากส่วนใหญ่คุ้นเคยกับการเรียกใช้งานเว็บเซอร์วิสจาก asp.net เป็นอย่างดี แต่ด้วยองค์ประกอบอย่างอื่นของการพัฒนาระบบซึ่งยากต่อการศึกษาและเรียกใช้งานเช่น การเรียกใช้ Google map API เพื่อนำมาช่วย

แสดงผลข้อมูล ซึ่งมีความซับซ้อน และระบบ GPS Tracking เป็นระบบที่แปลกใหม่สำหรับทีมพัฒนาซอฟต์แวร์ชุดนี้ จึงทำให้การทดสอบไม่คล่องตัวนัก แต่เมื่อสอบถามโดยมองจากมุมมองของการพัฒนาระบบเทียบกับการพัฒนาขึ้นมาทั้งหมดเองก็ได้รับความเห็นที่ตรงกันว่าการใช้เฟรมเวิร์คมาช่วยในการพัฒนานั้นสะดวกกว่ามาก และใช้เวลาน้อยกว่ามาก

4.4 อภิปรายผล

จากการทดสอบหลากหลายรูปแบบ และการพัฒนาระบบบริหารการขนส่งสินค้าโดยใช้เว็บเซอร์วิสที่พัฒนาขึ้นจากเฟรมเวิร์คที่ได้สร้างขึ้น พบว่าสามารถเรียกใช้และทำงานได้อย่างมีประสิทธิภาพ และจากการนำเฟรมเวิร์คไปให้ทีมพัฒนาซอฟต์แวร์ที่ขาดทั้งในส่วนความคุ้นเคยในเรื่องของภาษาที่ใช้ในการพัฒนาคือ Java และความเข้าใจในเรื่องของการพัฒนาระบบติดตามการขนส่งสินค้า แต่ก็สามารถทำความเข้าใจและนำมาใช้ได้อย่างรวดเร็ว

ด้วยการออกแบบเฟรมเวิร์คที่มีความยืดหยุ่นในการปรับแต่งและแก้ไขคุณสมบัติของข้อมูลและพฤติกรรมของระบบโดยการ Override และการออกแบบให้เป็นเว็บเซอร์วิสซึ่งค่อนข้างเป็นอิสระต่อกัน ทำให้สามารถนำไปพัฒนาเป็นระบบสนับสนุนการขนส่งสินค้าได้หลากหลายประเภทเช่นการขนส่งแบบเที่ยวเดียว, การขนส่งแบบหลายสถานีโดยที่ไม่จำเป็นต้องพัฒนาขึ้นมาใหม่เองทั้งหมด หรืออาจนำไปประยุกต์ใช้ร่วมกับระบบลายเซ็นอิเล็กทรอนิกส์เพื่อควบคุมการขนถ่ายสินค้าได้เช่นกัน อีกทั้งยังสามารถเลือกใช้ภาษาที่จะนำมาใช้ในการพัฒนาต่อยอดจากเว็เซอร์วิสได้อย่างอิสระ โดยไม่ต้องกังวลในเรื่องของแพลตฟอร์มที่จะนำมาเชื่อมต่อเพื่อแลกเปลี่ยนนำข้อมูลไปใช้ประโยชน์อีกด้วย

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 5
บทสรุป

งานวิจัยนี้ มีจุดมุ่งหมายที่จะพัฒนากรอบงานขึ้น เพื่อที่จะลดขั้นตอนการพัฒนาโปรแกรมระบบการติดตามการขนส่งสินค้าด้วยเทคโนโลยี GPS(Global Positioning System) โดยการนำซอฟต์แวร์โอเพ่นซอสซึ่งสามารถทำงานด้านการติดตามยานพาหนะด้วย GPS ที่ได้รับการยอมรับอย่างแพร่หลายมาปรับปรุงให้รองรับการติดตามการขนส่งและพัฒนาขีดความสามารถทางด้านการแลกเปลี่ยนข้อมูลกับระบบภายนอก โดยนำเอาแนวคิดทางด้านการพัฒนาเฟรมเวิร์คแ SOA (Service Oriented Architecture) มาวิเคราะห์และประยุกต์ใช้ในการพัฒนาให้อยู่ในรูปแบบของเฟรมเวิร์คที่เหมาะสมกับการอิมพลีเมนต์ให้อยู่ในรูปเว็บเซอร์วิส เพื่อนำไปใช้พัฒนาระบบการติดตามการขนส่งสินค้าและบริหารห่วงโซ่อุปทานที่ต้องมีความเกี่ยวข้องและใช้ข้อมูลร่วมกับระบบอื่นๆ อีกหลายระบบ ทำให้เกิดความสะดวกและสามารถใช้ประโยชน์จากข้อมูลของการติดตา ยานพาหนะด้วย GPS ได้มากขึ้น และด้วยแนวคิดเชิงบริการของ SOA ทำให้ระบบเว็บเซอร์วิสที่ได้จากการอิมพลีเมนต์เฟรมเวิร์คสามารถทำการแลกเปลี่ยนข้อมูลกับระบบอื่นโดยไม่ขึ้นกั แพลตฟอร์มและภาษาในการพัฒนาส่งผลให้ลดขีดจำกัดในเรื่องความเข้ากันได้ระหว่างแพลตฟอร์มในการพัฒนาระบบ ซึ่งเฟรมเวิร์คที่พัฒนาขึ้นมานี้ทำให้ลดขั้นตอน ต้นทุนในการพัฒนา รวมทั้งข้อผิดพลาดที่เกิดขึ้นจากบุคคล(Human error) สามารถนำไปประยุกต์ใช้ในการพัฒนาระบบได้อย่างรวดเร็ว

ขั้นตอนการพัฒนางานวิจัยชิ้นนี้ เริ่มจากการศึกษางานและกระบวนการขนส่ กระบวนการทำงานของการตรวจสอบและติดตามตำแหน่งยานพาหนะด้วย GPS โดยการทดลองใช้ซอฟต์แวร์สำหรับติดตามตำแหน่งยานพาหนะด้วย GPS ซึ่งเป็นโอเพ่นซอสคือ OpenGTS (Open GPS Tracking System :http://opengts.sourceforge.net/) หลังจากนั้นก็ได้เริ่มศึกษาโครงสร้างของซอสโค๊ดของ OpenGTSเพื่อหาช่องทางในการปรับปรุงให้รองรับการขนส่งสินค้าและคิดค้นวิธีการพัฒนาเพื่อนำข้อมูลที่ได้จากการเก็บข้อมูลของระบบ OpenGTSมาใช้ประโยชน์โดยการวิเคราะห์ร่วมกับแนวคิดของการสร้างเว็บเซอร์วิสภายใต้พื้นฐานของสถาปัตยกรรมเชิงบริการ(SOA:Service Oriented Architecture) โดยมีขอบเขตที่จะทำให้สามารถแลกเปลี่ยนข้อมูลที่เกี่ยวข้องกับการขนส่งสินค้ากับระบบอื่นได้ และได้พัฒนาให้อยู่ในรูปแบบของเฟรมเวิร์คเพื่อให้เกิดเป็นกรอบในกา

พัฒนาที่มีความยืดหยุ่นในการนำไปประยุกต์ใช้ได้หลากหลาย ผู้วิจัยได้ตัดสินใจใช้ภาษา JAVA ในการพัฒนาและได้ทดสอบเฟรมเวิร์คกับเครื่องมือพัฒนาเว็บเซอร์วิสหลายตัวเช่น Apache AXIS, Apache AXIS2, Apache CXF 2.x และได้เลือกใช้ Apache CXF2.5.2 เป็นเครื่องมือพัฒนาเว็บเซอร์วิสที่เหมาะสมกับเฟรมเวิร์คเนื่องจากเป็นเครื่องมือพัฒนาเว็บเซอร์วิสที่สามารถสร้างเว็บเซอร์วิสที่เข้ากันได้กับแพลตฟอร์มอื่นๆ ได้ดีที่สุดและมีการใช้ Service Endpoint Interface ซึ่งเหมาะสำหรับใช้ในเฟรมเวิร์คมากกว่าเครื่องมือตัวอื่น

สำหรับขั้นตอนในการทดสอบระบบนั้น ผู้วิจัยได้ออกแบบการทดสอบไว้ 3 รูปแบบคือ การทดสอบส่วนอธิบายบริการ(Service Description) โดยการทดลองทำ .net Service Reference, การทดสอบความถูกต้องของการทำงานของ Service Operation ด้วยการทำ Unit Testing ด้วย Visual Studio Unit Testing และการทดลองพัฒนาระบบด้วยเฟรมเวิร์คซึ่งจากการทดสอบทั้ง 3 รูปแบบก็ได้ผลเป็นที่น่าพอใจ และหลังจากนั้นก็ได้นำเฟรมเวิร์คไปให้นักพัฒนาซอฟต์แวร์จำนวน 3 ท่านทดลองทำการอิมพลีเมนต์เว็บเซอร์วิสสำหรับระบบการติดตามการขนส่งสินค้าขึ้นจากเฟรมเวิร์คและทดสอบการสร้างส่วนแสดงผลขึ้นด้วยเทคโนโลยีที่นักพัฒนาซอฟต์แวร์แต่ละคนถนัด ซึ่งก็ได้ผลตอบรับเป็นที่น่าพอใจ

5.1 สรุปผลการวิจัย

จากการวิจัยสามารถสรุปผลได้ดังนี้

การพัฒนากรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการขนส่งซึ่งผู้วิจัยได้ให้ชื่อย่อของกรอบงานว่า SBFSTT Framework (Service Based Framework For Supply Chain Tracking in Transportation) โดยผู้วิจัยได้ทำการพัฒนาต่อยอดจากระบบ OpenGTS(Open GPS Tracking System) ซึ่งจากการวิเคราะห์ ศึกษา พัฒนา และทดสอบ เฟรมเวิร์คถึง 3 รูปแบบ รวมทั้งจากการนำไปทดสอบความเข้าใจและแนวคิดในการอิมพลีเมนต์ระบบด้วยบุคลากรของหน่วยงานภาคเอกชนซึ่งเป็นนักพัฒนาซอฟต์แวร์ สามารถสรุปผลการวิจัยได้ว่าการพัฒนางานวิจัยชิ้นนี้ ซึ่งมีชื่อว่า กรอบงานเชิงบริการสำหรับการติดตามห่วงโซ่อุปทานในการข (Service Based Framework For Supply Chain Tracking in Transportation) สามารถสร้างเฟรมเวิร์คสำหรับการพัฒนาระบบเชิงบริการสำหรับการติดตามการขนส่งสินค้าขึ้นมาได้ ซึ่งมีส่วนช่วยในการพัฒนาระบบการติดตามการขนส่งสินค้า คือ ลดต้นทุน ความซับซ้อนในการพัฒนาระบบ รวมข้อผิดพลาดที่อาจเกิดขึ้นจากบุคคลในขั้นตอนการพัฒนาระบบการติดตามการขนส่งสินค้า และเฟรมเวิคสามารถนำไปสร้างเป็นเว็บเซอร์วิสบนพื้นฐานของสถาปัตยกรรมเชิงบริการซึ่งความสามารถในการแลกเปลี่ยนข้อมูลจากระบบติดตามการขนส่งสินค้าด้วย GPS กับระบบอื่นโดย

ไม่ขึ้นกับแพลตฟอร์มได้ ซึ่งผลงานจากการวิจัยสามารถนำไปประยุกต์ใช้ในการสร้างระบบก
สนับสนุนและติดตามการขนส่งสินค้าได้สะดวกกว่าการพัฒนาขึ้นเองทั้งหมด พร้อมกันนั้น
เนื่องจากการเลือกระบบ OpenGTS เป็นแกนหลักในการพัฒนาเฟรมเวิร์คในส่วนของการเก็บข้อมูล
ตำแหน่งดาวเทียมด้วย GPS ส่งผลให้สามารถรองรับมาตรฐานและเข้ากันได้กับอุปกรณ์ GPS ได้
หลากหลายอีกด้วย

5.2 การประยุกต์งานวิจัย

งานวิจัยชิ้นนี้สามารถนำไปประยุกต์ใช้ในการพัฒนาระบบเว็บเซอร์วิสเพื่อสนับสนุนกา
พัฒนาระบบการติดตามและสนับสนุนการขนส่งสินค้า ซึ่งเป็นระบบที่สามารถมีส่วนช่วยในการ
ติดตามห่วงโซ่อุปทานและสามารถประยุกต์ใช้กับระบบลอจิสติกส์อื่นๆ ได้ง่าย เนื่องจากออกแบบ
อยู่บนพื้นฐานแนวคิดของ SOA นักพัฒนาซอฟต์แวร์ที่ต้องการนำเฟรมเวิร์คนี้ไปใช้สามาร
ปรับแต่งเพื่อให้เหมาะสมกับระบบและลักษณะงานที่ต้องการนำไปประยุกต์ใช้ได้อย่างสะดวก
พร้อมกันนั้นยังสามารถเชื่อมต่อกับแอพพลิเคชันอื่นๆ ในการบริหารห่วงโซ่อุปทานและลอจิสติกส์
ได้

5.3 แนวคิดในการพัฒนาต่อ

เนื่องจากเฟรมเวิร์คนี้มีขอบข่ายการพัฒนาเพียงในส่วนของการสร้างเว็บเซอร์วิสดังนั้น
เพียงแค่การอิมพลีเมนต์เฟรมเวิร์คยังไม่สามารถช่วยให้ระบบทำงานได้ จำเป็นต้องมีส่วนติดต่อ
ผู้ใช้เหมาะสมและสามารถที่จะรองรับและเรียกใช้เว็บเซอร์วิสที่สร้างขึ้น ดังนั้นแนวคิดในการ
พัฒนาต่ออาจมีการสร้างเฟรมเวิร์คในส่วนของส่วนติดต่อผู้ใช้งานขึ้นมารองรับการทำงานร่วมกั
เฟรมเวิร์คนี้ในอนาคต

รายการอ้างอิง

Anderson J. (1983). **The Architecture of Cognition.** Harvard University Press, Cambridge, Massachusets.

Andrej Krevl, Mojca Ciglari. (2006). **A Framework for Developing Distributed Location Based Applications.** 1-4244-0054-6/06 IEEE.

Beck K. and Johnson R. (1994). **Patterns Generate Architectures.** In Proceedings of the 8th European Conference on Object-Oriented Programming, Bologna, Italy.

Dan Gisolfi. (2001). **Web Services Architecture, Part 1 : An introduction to dynamic e-business by Dan Gisolfi Solution Architect.** IBM, Start Emerging Technologies, Arpil 2001.

INTERTRANSPORT LOGISTICS ปีที่ 3 ฉบับที่ 63 วันที่ 1-15 มิถุนายน 2546 หน้า 4 /

Johnson, R. E. and Foote, J. (1988). **Designing Reusable Classes.** Journal of Object Oriented Programming, 1(2):22-35.

GeoTelematic Solutions, Inc. **The OpenGTS Project**. http://opengts.sourceforge.net access on : 10 January 2012.

K. Hausknecht, D. ŠimuniĆ, S. Afridi. (2011). **Real-time Urban Transportation Service Tracking and Monitoring.** MIPRO 2011, May 23-27, 2011, Opatija, Croatia.

Taligent. (1994). **Building Object-Oriented Frameworks,** A Taligent White Paper. Technical report, Taligent Inc.

Thomas Erl. (2006). **Service-Oriented Architecture Concepts, Technology and Design.** Prentice Hall.

Timothy C. Lethbridge, Robert Laganiere (2005). **Object-Oriented Software Engineering Practice Software Development using UML and JAVA.** McGRAW-HILL International.

วิศิษฏ์ วัฒนานุกูล. (2552) **การจัดการไอทีลอจิสติกส์.** กรุงเทพฯ. ซีเอ็ดยูเคชั่น.

สรวิศ นฤปิติ. **ระบบสารสนเทศภูมิศาสตร์ในงานระบบขนส่งอัจริยะ (ITS)Geographical Information System (GIS) in Intelligent Transport System (ITS)**

สุธี พงศาสสกุลชัย. (2550). **การพัฒนาระบบด้วยสถาปัตยกรรมเชิงบริการบนเทคโนโลยีของ Web Service.** สำนักพิมพ์เคทีพี.

ภาคผนวก ก

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างศึกษา

CHANINTORN RUANGUDOMSAKUL, PICHAYOTAI MAHATHANAPIWAT (2011). **USING OPENGTS FOR GPS TRACKING TO SUPPORT DEVELOPMENT OF CARGO TRANSPORT SYSTEM.** การประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 37 "วิทยาศาสตร์สร้างสรรค์ เพื่อสรรค์สร้างอนาคต" สมาคมวิทยาศาสตร์แห่งประเทศไทยในพระบรมราชูปถัมภ์ ร่วมกับ คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล, Centara Grand & Bangkok Convention Center at Central World กรุงเทพมหานคร ระหว่างวันที่ 10 – 12 ตุลาคม พ.ศ. 2554.

มหาวิทยาลัยเทคโนโลยีสุรนารี

USING OPENGTS FOR GPS TRACKING TO SUPPORT DEVELOPMENT OF CARGO TRANSPORT SYSTEM

Chanintorn Ruangudomsakul *, Pichayotai Mahathanapiwat

Department of computer engineering Suranaree University of Technology Nakornratchasima 3000, THAILAND
*e-mail: ch_ruang@hotmail.com

Abstract: This study aims to modify Open GPS Tracking System (OpenGTS) project to enhance usability and make possibility to connect and exchange data via web service. This project focuses on modification of original OpenGTS source code with added web service to provide SOAP interface for supply method and development of cargo transport tracking application. According to this study, we are able to customize original OpenGTS source code to add our propose module to support development of cross platform transport management.

Introduction: Global Positioning System or GPS is the technology that gives you the possibility to find an object location anywhere on the globe. The fundamental technique of GPS is to measure the ranges between the receiver and a few simultaneously observed satellites. The positions of the satellites are forecasted and broadcasted along with the GPS signal to the user. The Global Positioning System was designed and built, and is operated and maintained by the U.S. Department of Defense. GPS was conceived as a ranging system from known positions of satellites in space to unknown positions on land and sea, as well as in air and space. The most important applications of the GPS are positioning and navigating [1, 2].

In the field of supply chain management (SCM), GPS tracking technology has roles and more important in this time. That can be seen from cargo transport processes that have been integrated GPS tracking device for use of real-time tracking and tracing transport vehicle [2, 4].

OpenGTS™ ("Open GPS Tracking System") is the first available open source project designed specifically to provide web-based GPS tracking services for a "fleet" of vehicles. OpenGTS features were designed to fill requirement of entry-level fleet tracking system. But it is also very highly configurable and scalable to larger system as well.

According to information above and, this study aims to modify and develop OpenGTS extension module to provide service based API for support cargo transport system and makes OpenGTS to be able to exchange data with other application via web service. This will make OpenGTS more applicable and easy to apply for development system [5].

Methodology: We use OpenGTS version 2.3.2 as original system and Eclipse as IDE. OpenGTS itself is written entirely in Java, using technologies such as Apache Tomcat for web service deployment, and MySQL for the data store. But it can also configure to use other DBMS as data store too.

Tools is used for the system development are: MySQL database, Original OpenGTS 2.3.2 source code, Apache Tomcat 6.0 Servlet container, Apache Axis2, SOAP UI 4.0, Eclipse IDE.

Pre configuration and test pre configuration phrase:

This study configures OpenGTS to use MySQL for the data store and Google map as map provider. The study uses Windows mobile 6.5 smart phone with build-in GPS receiver as GPS device. Our mobile application designed in simple ways to send latitude/longitude, speed, time, and heading data to server through HTTP based protocols. Figure 1 shows OpenGTS UI with pre configuration and test tracking result.

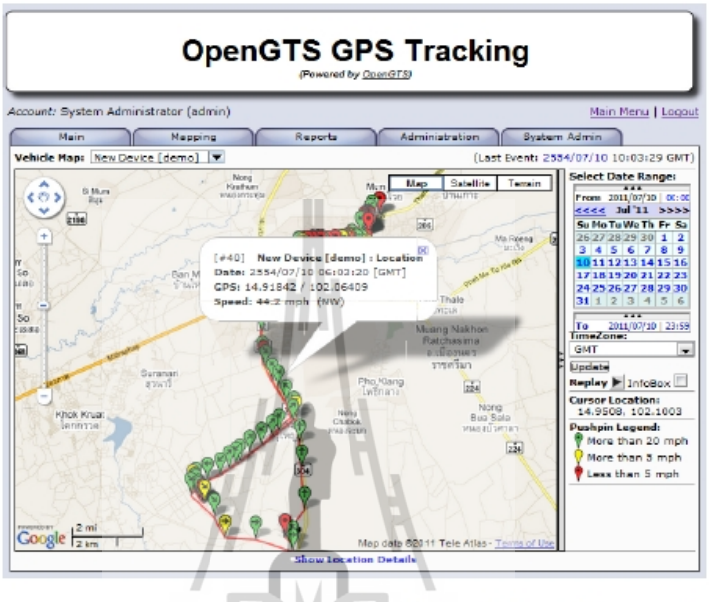


Figure 1. OpenGTS map user interface with tracking result

Cargo transport:

First we were looking for how to send cargo from sender to receiver. Figure 2 illustrates the typical operating processes for door-to door services offered by railway logistics enterprises. The process begins at the client's warehouse, where shipments are loaded onto a truck. Next, the shipments are transported to a consolidation center, where they are combined. Then they are delivered to the railway freight station and are shipped out by train. After arriving at the destination terminal, shipments are unloaded and reloaded onto a truck, and then are delivered to the local distribution center for holding and storage. From there, they are distributed to sales outlets by delivery trucks or vans [3].

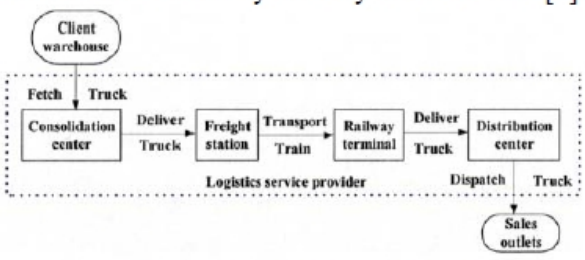


Figure 2. logistic service process [3].

From the logistic process above, we can summarize the general step of standard transport system for design tracking module. Table 1 shows general step of transport system. The process start with preparation step, which cargo labeled, packaged and placed in pallet. Then they will be loaded into container track or transport vehicle for shipping out to receiver. After track or transport vehicle arrive to receiver, station or distribution center. The cargo will be unloaded.

Table 1. Central step of cargo transport system

Business Step	Action
Preparation	label cargo, packing and placed in pallet
Loaded	Load products pallet into Container track or transport vehicle.
Shipping out	Transport vehicle send cargo to recover
Unload	When product arrive to station or distribution center. Unload cargo from vehicle

OpenGTS:

Figure 3 describes the basic system architecture of the **OpenGTS** system. The various device communication servers (the modules which listen for incoming data from the remote GPS tracking devices) run as separate processes on top of Java. The Track servlet (ie. The web-interface), as well as other servlets (including any http-based device communication server), run within a Servlet Container, such as Apache Tomcat.

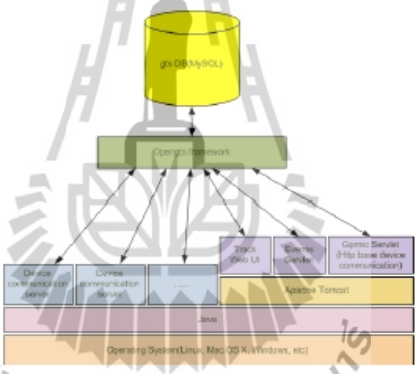


Figure 3. Basic system architecture of the OpenGTS system

Main resource of this study is the OpenGTS framework. The framework has module, pattern and basic of implementation such as class for define table record, utility, data access module etc. We can access to use for implementing our cargo transport system.

Implementation phase:

According to OpenGTS framework, we can use it to implement database table class for generate database table in MySQL. Figure 4 shows our purposed system block diagram. In this study, we implement extension module that contain additional data table class to support above simple cargo transport process. And then, we implement web service wrapper class and service parameter class to wrap original and custom additional data access layer of OpenGTS framework. We build and deploy web service using apache axis2. Figure 5 shows an example of class diagram of cargo transport. It shows basic of implementation to create table and access data using pattern and base class of OpenGTS framework. Figure 6 shows an example of our purposed web service test by SOAP UI.

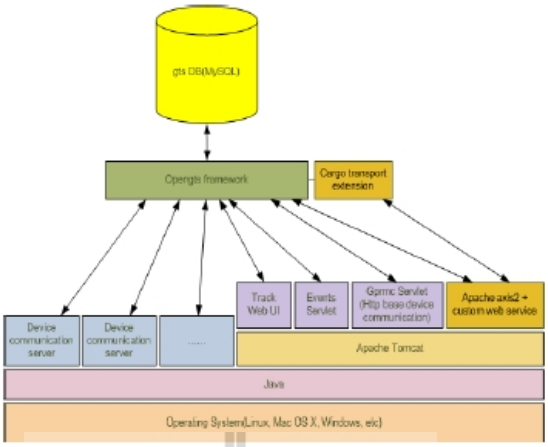


Figure 4. Our purpose system block diagram

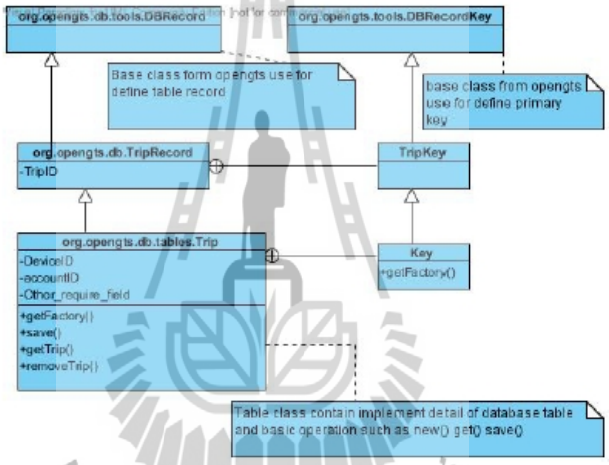


Figure 5. Example of class diagram of cargo transport

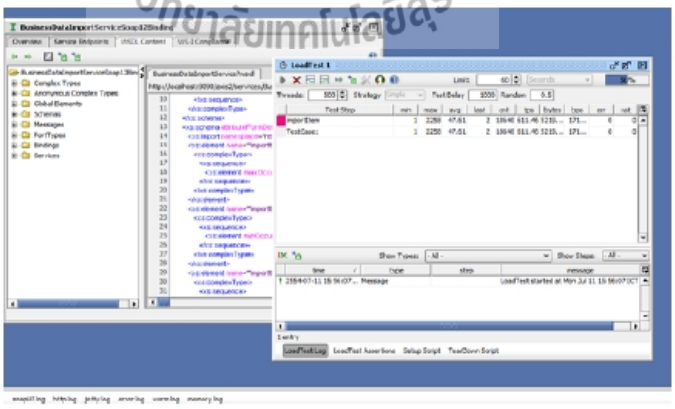


Figure 6. Example of our purposed web service test by SOAP UI.

Results, Discussion and Conclusion: After investigating test result and code review result. We found that our modified OpenGTS works correctly and can integrate original vehicle

GPS tracking with our additional cargo transport data. Our service can be consumed by other client. At the moment, the system is being tested, correcting and improving in order to support development of cargo transport system based on open source platform. Our study can be shown to developer how to modify and apply open source GPS tracking platform such as OpenGTS to use with business supply chain management and logistic. That will reduce cost, time and error in development and give chance to business to have self GPS tracking system with low cost.

References:

1. Noppadol Chadil Apirak Russameesawang Phongsak Keeratiwintakorn. Real-Time Tracking Management System Using GPS, GPRS and Google Earth. Proceedings of ECTI-CON 2008.
2. W. He, E. L. Tan, E. W. Lee, T. Y. Li. A solution for Integrated Track and Trace in Supply Chain based on RFID &GPS. 2009.
3. Yujie Wang, Yaoqiu Wang. Application Model of Small Containerized Cargo Units in Integrated Transportation: A Case Study. 2008.
4. *Yingli Wang, Andrew Potter*. The application of real time tracking technologies in freight transport. Third International IEEE Conference on Signal-Image Technologies and Internet-Based System.
5. YaoJuan, Wang Hongxia. Study on E-business Logistics System Based On SOA.

Keywords: openGTS, logistic, GPS, GPS tracking, open source

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้เขียน

นายชนินทร เรืองอุดมสกุล เกิดวันที่ 18 มีนาคม พ.ศ. 2527 ที่ อำเภอเมือง จังหวัดศรีสะเกษ เข้ารับการศึกษาในระดับชั้น อนุบาล 1 ที่โรงเรียนค่ายนารายศึกษา อำเภอเมือง จังหวัดลพบุรี และได้ย้ายมาเข้ารับการศึกษาชั้นอนุบาล 2 ประถมศึกษาปีที่ 6 ที่โรงเรียนอนุบาลศรีสะเกษ อำเภอเมือง จังหวัดศรีสะเกษ ระดับมัธยมศึกษาตอนต้น – ปลาย ที่โรงเรียนศรีสะเกษวิทยาลัย อำเภอเมือง จังหวัดศรีสะเกษ ในปีการศึกษา 2548 ได้เข้าศึกษาต่อในระดับปริญญาตรี ที่มหาวิทยาลัยราชภัฏศรีสะเกษ ในสาขาวิทยาการคอมพิวเตอร์ และสำเร็จการศึกษาด้วยเกียตินิยมอันดับ 1 ในปีการศึกษา 2551 ภายหลังสำเร็จการศึกษาระดับปริญญาตรี ได้เข้าศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยี-สุรนารี ในปีการศึกษา 2552

ในระหว่างการศึกษา ได้รับความอนุเคราะห์อย่างดียิ่งจากอาจารย์ที่ปรึกษา คณาจารย์ในสาขาวิชา และได้รับความไว้วางใจให้เป็นผู้ช่วยสอนในรายวิชา Object Oriented Technology, System Analysis, Software Engineering และ Event Driven Programming และได้มีบทความวิจัยนำเสนอในการประชุมวิชาการวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย ครั้งที่ 37 “วิทยาศาสตร์สร้างสรรค์ เพื่อสรรค์สร้างอนาคต” ในหัวข้อเรื่อง “USING OPENGTS FOR GPS TRACKING TO SUPPORT DEVELOPMENT OF CARGO TRANSPORT SYSTEM” ซึ่งจัดโดย สมาคมวิทยาศาสตร์แห่งประเทศไทยในพระบรมราชูปถัมภ์ ร่วมกับ คณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล ณ Centara Grand & Bangkok Convention Center at Central World กรุงเทพมหานคร ในระหว่างวันที่ 10 – 12 ตุลาคม พ.ศ. 2554