

ระบบจัดเก็บชิ้นส่วนโครงแบบของทรัพยากรการบริการไอทีโดยใช้อาร์ดีเอฟ



นายนราธิช ฌ ลำปาง

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)  
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2560

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A System for Collecting Configuration Items of IT Service Resources using RDF



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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2017

Copyright of Chulalongkorn University

|                                 |   |
|---------------------------------|---|
| หัวข้อวิทยานิพนธ์               | ระบบจัดเก็บชิ้นส่วนโครงสร้างของทรัพยากรการบริการไอ<br>ทีโดยใช้อาร์ดีเอฟ |
| โดย                             | นายนราทิช ณ ลำปาง   |
| สาขาวิชา                        | วิศวกรรมซอฟต์แวร์   |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ                                     |

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์  
.....ประธานกรรมการ  
(รองศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์)  
.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)  
.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์)  
.....กรรมการภายนอกมหาวิทยาลัย  
(ผู้ช่วยศาสตราจารย์ ดร.มณฑุปายาส ทองมาก)

นราทิช ณ ลำปาง : ระบบจัดเก็บชิ้นส่วนโครงสร้างของทรัพยากรการบริการไอทีโดยใช้อาร์ดีเอฟ (A System for Collecting Configuration Items of IT Service Resources using RDF) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 85 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอทางเลือกของการจัดเก็บชิ้นส่วนโครงสร้างของทรัพยากรการบริการไอที เพื่อให้การจัดการบริการเทคโนโลยีสารสนเทศมีประสิทธิภาพมากขึ้นโดยเฉพาะอย่างยิ่งการนำมาใช้กับการจัดการโครงสร้าง ซึ่งผู้บริหารจำเป็นต้องมีการวิเคราะห์ที่เหมาะสมและถูกต้องเพื่อการตัดสินใจ โดยนำเสนอการเก็บข้อมูลลงฐานความรู้ เพื่อให้มีความยืดหยุ่นในการรับมือกับปัญหาเชิงกลยุทธ์ ซึ่งโครงสร้างออนโทโลยีได้รับการออกแบบเพื่อพัฒนาโครงสร้างของทรัพยากรการบริการเทคโนโลยีสารสนเทศในรูปแบบของอาร์ดีเอฟเอ็นทีริปเปิล คลาสและลำดับชั้นของโครงสร้างถูกออกแบบให้แสดงความสัมพันธ์ของซีไอได้อย่างเหมาะสม การออกแบบการสืบค้นใช้ภาษาสเปคควอลเพื่อรองรับบริการเทคโนโลยีสารสนเทศ เช่น การคำนวณงบประมาณในการบำรุงรักษาประจำปี เป็นต้น โปรแกรมโพรเซสเซอร์ถูกเลือกมาเพื่อใช้ในการออกแบบโครงสร้างออนโทโลยี จากนั้นพัฒนาเว็บแอปพลิเคชันด้วยภาษาจาวาที่ใช้กรอบงานของจินาผ่านทางอาปาเซจินาฟูเซกิเซิร์ฟเวอร์ และเว็บแอปพลิเคชันสามารถทำงานได้สำเร็จตามแนวทางการใช้ออนโทโลยี

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2560

# # 5770998921 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: ONTOLOGY / ITSM / CONFIGURATION MANAGEMENT / CMDB

NARATICH NA-LAMPANG: A System for Collecting Configuration Items of IT Service Resources using RDF. ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, Ph.D., 85 pp.

This thesis proposes an alternative of a collecting configuration items of IT service resources . To make the IT service management more effective especially the configuration management, the manager needs the appropriate and ad-hoc analysis for decision making. Our approach proposes to stored data to knowledge base in order to flexibly cope with the strategic problems. An ontology schema is designed to develop configuration management of IT service resources in term of RDF N-triple. The classes and hierarchy of the schema are provided along with the appropriate property relations among the configuration items. We design the queries by SPARQL for IT service functions, such as how to allocate the budget for the annual maintenance fee, etc. The Protege software is used during our ontology schema design. We develop a Java web application with Jena Framework bundled in the Jena Fuseki ontology server, to successfully demonstrate our ontology-based approach.

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

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

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

Academic Year: 2017

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณ รศ. ดร.วิวัฒน์ วัฒนาวุฒิ ที่กรุณาให้คำปรึกษาและคำแนะนำในการทำวิจัยรวมทั้งการเขียนวิทยานิพนธ์และส่งผลงานทางวิชาการ แม้ข้าพเจ้าจะติดขัดหลายประการระหว่างทำวิจัยแต่ท่านก็สละเวลามาช่วยแก้ไขผลงานให้เสมือนจนช่วยผลักดันให้วิทยานิพนธ์ฉบับนี้สำเร็จได้

ขอกราบขอบพระคุณ รศ. ดร.ธราทิพย์ สุวรรณศาสตร์, ผศ. ดร.อาทิตย์ ทองทักษ์ และ ผศ. ดร.มธุปายาส ทองมาก ที่ช่วยสละเวลาให้คำแนะนำงานวิจัยตั้งแต่เริ่มการสอบโครงร่างและยังสอบถามด้วยความห่วงใยเสมอมาตลอด ซึ่งเป็นประโยชน์สำคัญในการทำวิจัยและทำให้งานวิทยานิพนธ์นี้สำเร็จได้



## สารบัญ

หน้า

|  |    |
|--|----|
| บทคัดย่อภาษาไทย.....   | ง  |
| บทคัดย่อภาษาอังกฤษ.....  | จ  |
| กิตติกรรมประกาศ.....   | ฉ  |
| สารบัญ.....  | ช  |
| สารบัญตาราง.....   | ฅ  |
| สารบัญภาพ .....  | ญ  |
| บทที่ 1 บทนำ .....   | 1  |
| 1.1 ความเป็นมาและความสำคัญของปัญหา.....  | 1  |
| 1.2 วัตถุประสงค์ของการวิจัย .....  | 2  |
| 1.3 ขอบเขตงานวิจัย .....   | 2  |
| 1.4 ขั้นตอนงานวิจัย .....  | 3  |
| 1.5 บทความงานวิจัยที่ได้รับการตีพิมพ์ .....  | 3  |
| บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....  | 4  |
| 2.1 การจัดการบริการเทคโนโลยีสารสนเทศ ตามกรอบความคิดไอทิล [1].....                  | 4  |
| 2.2 ซีไอและซีไอแอกกริเกต [3].....  | 7  |
| 2.3 การจัดการโครงสร้างและ ฐานข้อมูลการจัดการโครงสร้างตามกรอบความคิดไอทิล [4] ..... | 7  |
| 2.3 ออนโทโลยี และ ภาษาอวาล์ (OWL: Web Ontology Language).....                      | 11 |
| 2.4 โปรแกรมโพรเทจ (Protégé Program).....   | 14 |
| 2.5 ภาษาสพาควอล (SPARQL) .....   | 16 |
| 2.6 เอกสารการทดสอบ [12].....   | 17 |
| 2.7 ระบบสารสนเทศ (Information system).....   | 19 |
| 2.8 งานวิจัยที่เกี่ยวข้อง.....   | 22 |

|   |    |
|---|----|
| บทที่ 3 การวิเคราะห์และออกแบบโครงร่างออนไลน์ของซีไอ .....                   | 27 |
| 3.1 การวิเคราะห์และกำหนดซีไอ .....  | 28 |
| 3.2 การออกแบบแผนภาพคลาส .....   | 32 |
| 3.3 การออกแบบโครงร่างพื้นฐานออนไลน์ .....                                   | 54 |
| บทที่ 4 การพัฒนาโปรแกรมสำหรับการจัดการโครงแบบ .....                         | 58 |
| 4.1 การออกแบบแผนภาพยูสเคส.....  | 58 |
| 4.2 การออกแบบสถาปัตยกรรมของระบบการจัดการโครงแบบ .....                       | 58 |
| 4.3 การออกแบบแผนภาพคลาสและแผนภาพลำดับ .....                                 | 60 |
| 4.4 การออกแบบแผนผังเว็บแอปพลิเคชันการจัดการโครงแบบ.....                     | 63 |
| 4.5 การสร้างฐานข้อมูลออนไลน์ .....  | 64 |
| บทที่ 5 กรณีศึกษาสำหรับระบบการจัดการโครงแบบ .....                           | 67 |
| 5.1 ออกแบบกรณีศึกษาสร้างข้อมูลซอฟต์แวร์ที่ให้บริการทางธุรกิจของธนาคาร ..... | 67 |
| 5.2 ทดสอบใช้งานโปรแกรมการจัดการโครงแบบ .....                                | 68 |
| บทที่ 6 สรุปผลงานวิจัยและข้อเสนอแนะ .....                                   | 80 |
| 6.1 สรุปผลงานวิจัย .....  | 80 |
| 6.2 ประโยชน์ที่ได้รับ .....   | 81 |
| 6.3 ข้อจำกัดของระบบ .....   | 81 |
| 6.4 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ .....                                  | 81 |
| รายการอ้างอิง .....   | 82 |
| ประวัติผู้เขียนวิทยานิพนธ์ .....  | 85 |



## สารบัญตาราง

|   |    |
|---|----|
| ตารางที่ 2.1 คุณสมบัติทั่วไปของซีไอ [4].....                                      | 11 |
| ตารางที่ 3.1 ประเภทซีไอที่กำหนด .....   | 31 |
| ตารางที่ 3.2 ซีไอที่กำหนด.....  | 31 |
| ตารางที่ 3.3 ความสัมพันธ์ระหว่างคลาส.....   | 33 |
| ตารางที่ 3.4 คุณสมบัติของคลาส ITInfrastructure.....                               | 37 |
| ตารางที่ 3.5 คุณสมบัติของคลาส ChangeRequest.....                                  | 38 |
| ตารางที่ 3.6 ผลสรุปคุณสมบัติเพิ่มเติมของคลาส Hardware และซัพคลาสของ Hardware..... | 39 |
| ตารางที่ 3.7 ผลสรุปคุณสมบัติเพิ่มเติมของคลาส Network และซัพคลาสของ Network.....   | 40 |
| ตารางที่ 3.8 คุณสมบัติเพิ่มเติมของคลาส Software และซัพคลาสของ Software.....       | 42 |
| ตารางที่ 3.9 คุณสมบัติเพิ่มเติมของคลาส Document และซัพคลาสของคลาส Document.....   | 46 |

## สารบัญภาพ

|   |    |
|---|----|
| รูปที่ 2.1 แผนภาพกระบวนการในการสนับสนุนการบริการ [2] .....                                    | 6  |
| รูปที่ 2.2 ตัวอย่างซีไอและซีไอแอกกรีเกต [3] .....   | 7  |
| รูปที่ 2.3 แผนภาพกิจกรรมในระบบการจัดการโครงแบบ [4].....                                       | 8  |
| รูปที่ 2.4 ตัวอย่างโครงสร้างแบบแยกย่อย [5].....   | 10 |
| รูปที่ 2.5 ตัวอย่างการออกแบบออนโทโลยีในการให้บริการเว็บเซอร์วิส [9].....                      | 13 |
| รูปที่ 2.6 ตัวอย่างข้อมูลภาษาอาวล์ในการออกแบบการให้บริการเว็บเซอร์วิส [9].....                | 13 |
| รูปที่ 2.7 ตัวอย่างการสร้างข้อมูลภาษาอาวล์เพื่อสร้างข้อมูลออนโทโลยีของพีชซ่า [10].....        | 15 |
| รูปที่ 2.8 ส่วน "Classes" บนโปรแกรมโพรเทจ [10].....   | 15 |
| รูปที่ 2.9 รายละเอียดของปุ่มในส่วน "Classes" บนโปรแกรมโพรเทจ [10] .....                       | 16 |
| รูปที่ 2.10 หน้ากรอกข้อมูลการสร้างคลาสในส่วน "Classes" บนโปรแกรมโพรเทจ [10].....              | 16 |
| รูปที่ 2.11 ตัวอย่างข้อมูลคลาสที่ได้เมื่อสร้างข้อมูลออนโทโลยีพีชซ่าบนโปรแกรมโพรเทจ [10] ..... | 16 |
| รูปที่ 2.12 ตัวอย่างการเขียนภาษาสพาควอล [11].....   | 17 |
| รูปที่ 2.13 ความสัมพันธ์ของเอกสารการทดสอบ [12] .....  | 18 |
| รูปที่ 2.14 ตัวอย่างเครือข่ายและอุปกรณ์ในการเชื่อมต่อเบื้องต้น [14].....                      | 20 |
| รูปที่ 2.15 ตัวอย่างเครือข่ายบริเวณกว้าง [14].....  | 20 |
| รูปที่ 2.16 ชั้นของซอฟต์แวร์ประเภทต่างๆ [16] .....  | 21 |
| รูปที่ 2.17 แผนภาพยูสเคสเพื่อการออกแบบฐานข้อมูลการจัดการโครงแบบ [17] .....                    | 22 |
| รูปที่ 2.18 โมเดลของการบริหารจัดการบริการเทคโนโลยีสารสนเทศ [18].....                          | 23 |
| รูปที่ 2.19 โครงสร้างสถาปัตยกรรมของอาปาเซจึนา [19].....                                       | 24 |
| รูปที่ 2.20 แผนภาพดีพลอยเมนต์ชั้นฟาสาดสำหรับอาปาเซจึนา [19].....                              | 25 |
| รูปที่ 2.21 แนวคิดการพัฒนาเว็บเชิงความหมายด้วยกรอบงานอาปาเซจึนา [19].....                     | 25 |
| รูปที่ 3.1 ภาพรวมการพัฒนาระบบจัดการโครงแบบด้วยออนโทโลยี .....                                 | 27 |

|  |    |
|--|----|
| รูปที่ 3.2 โครงสร้างแบบแยกย่อยของโครงสร้างพื้นฐานระบบงานเทคโนโลยีสารสนเทศ .....                                    | 30 |
| รูปที่ 3.3 แผนภาพคลาสเบื้องต้นของระบบ.....   | 36 |
| รูปที่ 3.4 แผนภาพคลาสแบบละเอียดของ คลาส ITInfrastructure, คลาส ChangeRequest, คลาส Hardware และ คลาส Network ..... | 40 |
| รูปที่ 3.5 แผนภาพคลาสแบบละเอียดของคลาส Software และซัพคลาสของคลาส Software.....                                    | 42 |
| รูปที่ 3.6 แผนภาพคลาสแบบละเอียดของคลาส Document และซัพคลาสของคลาส Document.....                                    | 45 |
| รูปที่ 3.7 แผนภาพวัตถุส่วนของเอกสาร .....  | 49 |
| รูปที่ 3.8 แผนภาพวัตถุแสดงตัวอย่างเอกสารก่อนการเปลี่ยนเวอร์ชัน .....   | 50 |
| รูปที่ 3.9 แผนภาพวัตถุแสดงตัวอย่างเอกสารหลังมีการขอเปลี่ยนแปลง .....   | 50 |
| รูปที่ 3.10 แผนภาพวัตถุแสดงตัวอย่างซอฟต์แวร์เมื่อมีการอัปเดต .....   | 52 |
| รูปที่ 3.11 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเบื้องต้น .....  | 53 |
| รูปที่ 3.12 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเมื่อมีการเพิ่มฮาร์ดแวร์ .....                             | 53 |
| รูปที่ 3.13 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเมื่อมีการเปลี่ยนแปลงค่า .....                             | 54 |
| รูปที่ 3.14 การออกแบบออนโทโลยีคลาสบนโปรแกรมโปรเจกต์ .....  | 55 |
| รูปที่ 3.15 ตัวอย่างการออกแบบคุณสมบัติแบบชนิดข้อมูล .....  | 56 |
| รูปที่ 3.16 ผลการออกแบบคุณสมบัติแบบชนิดข้อมูลของคลาส IT Infrastructure .....                                       | 56 |
| รูปที่ 3.17 ตัวอย่างการสร้างคุณสมบัติแบบวัตถุชื่อ HasDocument .....  | 57 |
| รูปที่ 4.1 แผนภาพยูสเคสระบบจัดการโครงแบบ.....  | 59 |
| รูปที่ 4.2 แผนภาพสถาปัตยกรรมของระบบ .....  | 60 |
| รูปที่ 4.3 แผนภาพคลาสหลักของโปรแกรมจัดการโครงแบบ .....   | 61 |
| รูปที่ 4.4 แผนภาพคลาสของโปรแกรมจัดการโครงแบบ .....   | 62 |
| รูปที่ 4.5 แผนภาพลำดับการใช้งานแต่ละฟังก์ชัน .....   | 63 |
| รูปที่ 4.6 แผนผังเว็บไซต์การจัดการโครงแบบ .....  | 64 |

|   |    |
|---|----|
| รูปที่ 4.7 ภาพการบันทึกข้อมูลจากโปรแกรมโพรเทจ .....   | 64 |
| รูปที่ 4.8 หน้าจอเมนูส่วนการจัดการชุดข้อมูลบนอาปาเซจินาฟูเซกิ .....   | 65 |
| รูปที่ 4.9 หน้าจอเมนูส่วนการจัดการชุดข้อมูลบนอาปาเซจินาฟูเซกิ .....   | 65 |
| รูปที่ 4.10 หน้าจอแสดงผลข้อมูลภายในชุดข้อมูล “ds” .....   | 66 |
| รูปที่ 5.1 แผนภาพแบบวัตถุของระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจ Loan ส่วนของการทดสอบ .....                       | 67 |
| รูปที่ 5.2 แผนภาพแบบวัตถุของระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจ Loan ส่วนของเครือข่ายฮาร์ดแวร์และซอฟต์แวร์ ..... | 68 |
| รูปที่ 5.3 หน้าจอหน้าแรกของโปรแกรมการจัดการโครงแบบ .....  | 69 |
| รูปที่ 5.4 หน้าจอ New Business System .....   | 69 |
| รูปที่ 5.5 หน้าจอเพิ่มข้อมูลฮาร์ดแวร์ .....   | 70 |
| รูปที่ 5.6 หน้าจอกรอกข้อมูลเซิร์ฟเวอร์ .....  | 70 |
| รูปที่ 5.7 หน้าจอผลของการกรอกข้อมูลส่วนฮาร์ดแวร์ .....  | 71 |
| รูปที่ 5.8 หน้าจอส่วนของการเพิ่มข้อมูลเอกสาร .....  | 71 |
| รูปที่ 5.9 หน้าจอส่วนของการกรอกข้อมูลเอกสารแผนการทดสอบ .....  | 72 |
| รูปที่ 5.10 หน้าจอส่วนของการกรอกข้อมูลเอกสารความต้องการของผู้ใช้งาน .....                                       | 72 |
| รูปที่ 5.11 หน้าจอส่วนของการกรอกข้อมูลเอกสารข้อกำหนดความต้องการ .....   | 73 |
| รูปที่ 5.12 หน้าจอส่วนของการเพิ่มข้อมูลซอฟต์แวร์และเครือข่าย .....  | 73 |
| รูปที่ 5.13 หน้าจอส่วนของการกำหนดความสัมพันธ์ของซีไอ .....  | 74 |
| รูปที่ 5.14 หน้าจอหลังจากสร้าง Business System สำเร็จ .....   | 75 |
| รูปที่ 5.15 หน้าจอหน้า View Existing Business System .....  | 75 |
| รูปที่ 5.16 หน้าจอแสดงข้อมูลของ Business System .....   | 76 |
| รูปที่ 5.17 ตัวอย่างกราฟที่แสดงข้อมูลองค์ประกอบต่างๆของ Business System .....                                   | 76 |
| รูปที่ 5.18 ตัวอย่างหน้า Inquiry Data .....   | 77 |

|  |    |
|--|----|
| รูปที่ 5.19 ตัวอย่างการส่งออกข้อมูลซีไอเป็นแฟ้มเอกซ์เซล.....         | 77 |
| รูปที่ 5.20 หน้าจอ Template CI.....                                  | 77 |
| รูปที่ 5.21 ตัวอย่างเพิ่มเติมแบบเอกซ์เซลซีไอ Server.....             | 78 |
| รูปที่ 5.22 ตัวอย่างเพิ่มเอกซ์เซลซีไอ Server ที่กรอกข้อมูลแล้ว ..... | 78 |
| รูปที่ 5.23 หน้าจอแสดงผลการใช้งานฟังก์ชัน Import CI .....            | 78 |
| รูปที่ 5.24 หน้าจอ List CI.....                                      | 79 |



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีสารสนเทศถือเป็นสิ่งจำเป็นสำหรับองค์กร ซึ่งองค์กรต่างๆ ได้มีการนำเทคโนโลยีเข้ามาช่วยให้การทำงานในองค์กรเป็นไปอย่างสะดวกและรวดเร็วขึ้น เนื่องจากการนำเทคโนโลยีสารสนเทศมาใช้โดยไม่มีการจัดการที่ดีอาจทำให้เกิดปัญหาและความไม่เป็นระเบียบของระบบงาน ฉะนั้นจึงต้องมีแนวทางการบริหารจัดการตามกรอบงานของไอทิล (ITIL: Information Technology Infrastructure Library) มาช่วยในการจัดการบริการเทคโนโลยีสารสนเทศ (IT Service Management) โดยเฉพาะอย่างยิ่งกับองค์กรที่ต้องการเสถียรภาพของระบบสูง เช่น องค์กรที่เกี่ยวกับด้านการเงิน การธนาคารซึ่งองค์กรเหล่านี้ความผิดพลาดในระบบหรือความผิดพลาดของโปรแกรมประยุกต์ใช้งานต้องมีการจัดการควบคุมให้เกิดความผิดพลาดน้อยที่สุด รวมทั้งหากเกิดปัญหา (Incident) ต้องสามารถจัดการแก้ปัญหาได้รวดเร็วและมีการจัดการที่เป็นแบบแผน นอกจากนี้ระบบงานที่มีการใช้งานข้อมูลหรือบริการใดร่วมกันก็มีความเสี่ยงที่หากมีการแก้ไขข้อมูลหรือเปลี่ยนแปลงข้อมูลจากฝ่ายใดฝ่ายหนึ่งจะมีผลกระทบต่อระบบโดยรวม ดังนั้นการเปลี่ยนแปลงใดๆ เหล่านี้ จึงจำเป็นต้องมีการควบคุม [1] ซึ่งไอทิลนั้นมีกระบวนการ การจัดการการเปลี่ยนแปลง (Change Management) และการจัดการโครงสร้าง (Configuration Management) ทำหน้าที่ควบคุมการเปลี่ยนแปลงข้อมูลและบริหารจัดการข้อมูลต่าง ๆ ของระบบไว้ และจัดเก็บลงในฐานข้อมูลการจัดการโครงสร้าง (CMDB: Configuration Management Database) ซึ่งโดยทั่วไปต้องออกแบบโครงสร้างฐานข้อมูล (Database Schema) ให้ครอบคลุมชิ้นงานในโดเมนทั้งหมดและหากต้องการแก้ไขเปลี่ยนแปลงโครงสร้างฐานข้อมูลจะใช้เวลาและอาจมีค่าใช้จ่าย

ผู้วิจัยได้เสนอแนวคิดที่จะพัฒนาระบบการจัดการโครงสร้างโดยเก็บข้อมูลพื้นฐานที่สำคัญลงฐานความรู้ในรูปแบบของข้อมูลออนโทโลยี (Ontology) ซึ่งออนโทโลยีนั้นมีจุดเด่นที่สามารถอธิบายความหมายและความสัมพันธ์ระหว่างสิ่งต่างๆ ได้ครอบคลุมและมีความยืดหยุ่นกว่าโครงสร้างฐานข้อมูลทั่วไปสามารถแก้ไขโครงสร้างได้ง่าย ขั้นตอนการพัฒนาจะเริ่มจากการระบุซีไอ (CI: Configuration Item) โดยซีไอตามความหมายของไอทิลคือชิ้นงานที่มีความสำคัญ ซึ่งซีไอนั้นสามารถเก็บคุณลักษณะต่างๆ และความสัมพันธ์ไปยังซีไออื่นๆ อีกทั้งมีความยืดหยุ่นในการเพิ่มหรือปรับความสัมพันธ์ได้ นอกจากนี้ยังสามารถใช้ประโยชน์ในการจัดการบริการเทคโนโลยีสารสนเทศ หรือนำข้อมูลไปวิเคราะห์ต่อไป เช่น ทำนายผลกระทบที่จะเกิดขึ้นในระบบจากการเปลี่ยนแปลงซีไอ เป็นต้น

## 1.2 วัตถุประสงค์ของการวิจัย

- 1) ออกแบบโครงร่างออนไลน์เพื่อสนับสนุนการจัดการโครงแบบ
- 2) พัฒนาโปรแกรมสนับสนุนการจัดการโครงแบบ

## 1.3 ขอบเขตงานวิจัย

- 1) ออกแบบโครงร่างข้อมูลออนไลน์เพื่อสนับสนุนการจัดการโครงแบบดังนี้
  - 1.1) ระบุซีไอจากกระบวนการของไอทิล และวิธีปฏิบัติในซีเอ็มเอ็มไอพื้นที่กระบวนการการจัดการโครงแบบ
  - 1.2) ใช้ภาษาอวาล์ ในการกำหนดโครงร่างข้อมูลออนไลน์ของซีไอ
  - 1.3) โครงร่างข้อมูลออนไลน์ประกอบด้วย
    - 1.3.1) ซีไอหลักคือ ซีไอระบบงาน ซีไอซอฟต์แวร์ระบบ ซีไอซอฟต์แวร์ประยุกต์ ซีไอเซิร์ฟเวอร์ ซีไอข้อกำหนดการทดสอบ ซีไอเอกสารความต้องการ
    - 1.3.2) ความสัมพันธ์ระหว่างซีไอแบ่งได้เป็น 6 ประเภทเป็นอย่างน้อย เช่น "Is a component of", "Is associated with", "Uses", "Is a characteristic of", "Is a new version of"
  - 1.4) ซีไอแอกกรีเกตมีระดับชั้นไม่เกิน 3 ชั้น
  - 1.5) รองรับการเก็บเวอร์ชันได้อย่างน้อย 2 เวอร์ชัน
- 2) โปรแกรมเพื่อสนับสนุนการจัดการโครงแบบ มีขอบเขตดังนี้
  - 2.1) สามารถส่งออกข้อมูลซีไอเป็นแฟ้มเอกซ์เซล
  - 2.2) สามารถนำเข้าข้อมูลซีไอจากแฟ้มเอกซ์เซลในรูปแบบเดียวกับข้อมูลที่ส่งออกตามหัวข้อที่ 2.1 ได้
  - 2.3) สามารถทำงานในการสืบค้นข้อมูลและการจัดการซีไอ ซึ่งครอบคลุม กิจกรรมในการจัดการโครงแบบ ดังนี้
    - 2.3.1) สร้างซีไอ
    - 2.3.2) เปลี่ยนแปลงซีไอ
    - 2.3.3) ลบซีไอ
    - 2.3.4) สืบค้นข้อมูลซีไอ
  - 2.4) ใช้ภาษาสพาควอลในการเข้าถึงฐานข้อมูลออนไลน์

3) กรณีศึกษาของระบบการจัดการโครงแบบ เป็นระบบสารสนเทศขององค์กรเทคโนโลยีสารสนเทศที่มีประเภทของ ซีโอและซีไอแอกกรีเกตที่มีระดับขั้นและเวอร์ชันครบถ้วนตามที่ได้ ออกแบบไว้

#### 1.4 ขั้นตอนงานวิจัย

- 1) ศึกษาข้อมูลและทำความเข้าใจการจัดการโครงแบบ
- 2) ศึกษาข้อมูลและทำความเข้าใจออนโทโลยีและอาปาเซจินา
- 3) วิเคราะห์และกำหนดขอบเขตของงานวิจัย
- 4) เลือกวิธีการและเครื่องมือที่เหมาะสมกับงานวิจัย
- 5) ออกแบบและพัฒนาฐานข้อมูลการจัดการโครงแบบ
- 6) ออกแบบและพัฒนาโปรแกรมสำหรับการจัดการโครงแบบ
- 7) ประเมินผลงานวิจัย
- 8) ตีพิมพ์ผลงานทางวิชาการ
- 9) สรุปผลการวิจัย และจัดทำวิทยานิพนธ์

#### 1.5 บทความงานวิจัยที่ได้รับการตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความวิชาการ คือเรื่อง “Development of an Ontology-Based Configuration Management System” โดย นราทิช ฌ ลำปาง และ วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ INTERNATIONAL CONFERENCE 8th Edition Electronics, Computers and Artificial Intelligence (ECAI 2016) จัดโดย UNIVERSITY OF PITESTI และ PETROLEUM-GAS UNIVERSITY OF PLOIESTI เมื่อวันที่ 30 มิถุนายน – 2 กรกฎาคม พ.ศ. 2559 ณ ประเทศโรมาเนีย



## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะครอบคลุมเนื้อหาเกี่ยวกับทฤษฎีที่เกี่ยวข้องกับงานวิจัยได้แก่ การจัดการบริการ เทคโนโลยีสารสนเทศตามกรอบความคิดของไอทิล ซีไอ การจัดการโครงแบบ ออนโทโลยี โปรแกรม โปรเทจ ภาษาสเปคควอล เอกสารการทดสอบ ระบบสารสนเทศ ซอฟต์แวร์ รวมทั้งงานวิจัยที่เกี่ยวข้อง ดังรายละเอียดในหัวข้อย่อยต่อไปนี้

#### 2.1 การจัดการบริการเทคโนโลยีสารสนเทศ ตามกรอบความคิดไอทิล [1]

ไอทิลเป็นแหล่งความรู้เปิด (Open Knowledge) ที่มีการรวบรวมเป็นเอกสารที่อธิบายถึงแนวทางที่ดีที่สุด (Best Practices) ในด้านการบริหารจัดการโครงสร้างของหน่วยงานด้านเทคโนโลยีสารสนเทศ (IT Infrastructure) ไอทิลถูกพัฒนาโดย Office for Government Commerce (OGC) ของประเทศอังกฤษ ต่อมาได้มีการประกาศเป็นมาตรฐานสากลชื่อว่า ISO-20000:2005 ไอทิลมีการพัฒนาอย่างต่อเนื่องนับจากที่ได้มีการเปิดตัวไอทิลเวอร์ชัน 1 เมื่อปี ค.ศ. 1990 ผ่านมาจนถึงปี ค.ศ. 2007 ได้เพิ่มเติมและปรับปรุงเป็น ไอทิลเวอร์ชัน 2 และล่าสุดในปี ค.ศ. 2011 มีการประกาศเปิดตัว ไอทิล เวอร์ชัน 3 ขึ้นมา เพื่อให้สอดคล้องกับแนวเทคโนโลยีและโลกธุรกิจที่เปลี่ยนแปลงไป โดยในเอกสารของไอทิลจะเป็นการรวบรวมแนวทางปฏิบัติจากหน่วยงานทางด้านเทคโนโลยีสารสนเทศต่างๆ ทั่วโลก เพื่อนำมากำหนดเป็นแนวทาง ซึ่งจะกำหนดให้เทคโนโลยีสารสนเทศ เป็นบริการที่ต้องตอบสนองต่อความต้องการของธุรกิจ ผู้ใช้ รวมถึงลูกค้า ไอทิลมีแนวทางปฏิบัติในส่วนที่ดูแลและพัฒนาทางด้านเทคโนโลยีสารสนเทศให้ตรงกับความต้องการของผู้ใช้งานและยังมีส่วนการสนับสนุนการบริการ (Service Support) เป็นแนวทางปฏิบัติในส่วนที่จัดการเกี่ยวกับงานระดับปฏิบัติการประจำวัน เช่น การบริหารงานศูนย์คอมพิวเตอร์ การดูแลเครือข่ายหรือโปรแกรมประยุกต์ซึ่งจะดูแลและควบคุมกระบวนการต่างๆ ให้เป็นไปอย่างมีระบบและมีประสิทธิภาพ ไอทิลจะถูกใช้เป็นแนวทางในการจัดการด้านเทคโนโลยีสารสนเทศ ที่ให้แต่แนวทางแต่ไม่ได้กล่าวถึงวิธีการดำเนินการ

แนวทางปฏิบัติของไอทิลที่จะนำเทคโนโลยีสารสนเทศมาช่วยดำเนินการอย่างไรเพื่อให้ธุรกิจบรรลุถึงเป้าหมายได้ ซึ่งจะแบ่งแนวทางปฏิบัติได้เป็น 2 ส่วนคือ

- 1) การจัดส่งการบริการ (Service Delivery) เป็นแนวทางปฏิบัติในส่วนที่ต้องติดต่อกับผู้ใช้หรือลูกค้า หาวิธีการแก้ปัญหา (Solution) ให้ผู้ใช้หรือลูกค้า ดูแลและพัฒนาทางด้านเทคโนโลยีสารสนเทศให้ตรงกับความต้องการของผู้ใช้งาน

2) การสนับสนุนการบริการ (Service Support) เป็นแนวทางปฏิบัติในส่วนที่จัดการเกี่ยวกับงานระดับปฏิบัติการประจำวัน (Day-To-Day Operation) เช่น การบริหารงานศูนย์คอมพิวเตอร์ การดูแลเครือข่ายหรือโปรแกรมประยุกต์ซึ่งจะดูแลและควบคุมกระบวนการต่างๆ ให้เป็นไปอย่างมีระบบและมีประสิทธิภาพ

การสนับสนุนการบริการ มีกระบวนการทำงานที่หลากหลาย รวมทั้งมีอุปกรณ์และเอกสารจำนวนมากจึงจำเป็นต้องมีการจัดเก็บข้อมูลที่ดี และมีการบริหารจัดการกระบวนการที่เป็นระบบ

การสนับสนุนการบริการ มีทั้งหมด 6 กระบวนการ ได้แก่

1) เซอร์วิสเดส (Service Desk)

เซอร์วิสเดสทำหน้าที่เป็นศูนย์กลางในการติดต่อระหว่างผู้ให้บริการและผู้ใช้บริการ เมื่อผู้ใช้งานพบปัญหาการใช้งานจะแจ้งปัญหาไปยังเซอร์วิสเดส

2) การจัดการการเปลี่ยนแปลง (Change Management)

การจัดการการเปลี่ยนแปลงเป็นกระบวนการที่ช่วยประเมินผลกระทบของการเปลี่ยนแปลง มีกระบวนการอนุมัติคำร้องขอเปลี่ยนแปลง (Request For Change; RFC) ก่อนที่จะทำการเปลี่ยนแปลงหรือดีพลอยเมนต์

3) การจัดการรีลีส (Release Management)

การจัดการรีลีสเป็นส่วนการจัดการที่ช่วยกำหนดวันดีพลอยเมนต์เพื่อช่วยวางแผนการดีพลอยเมนต์ให้ไม่กระทบกับการใช้งานของผู้ใช้และแอปพลิเคชันอื่นๆ

4) การจัดการโครงสร้าง

การจัดการโครงสร้างคือการสร้างและบำรุงรักษาความเข้ากันได้ของชิ้นงานในระบบ โดยมีกระบวนการที่ช่วยจัดเก็บและจัดการข้อมูลซีไอให้เป็นปัจจุบันอยู่เสมอ

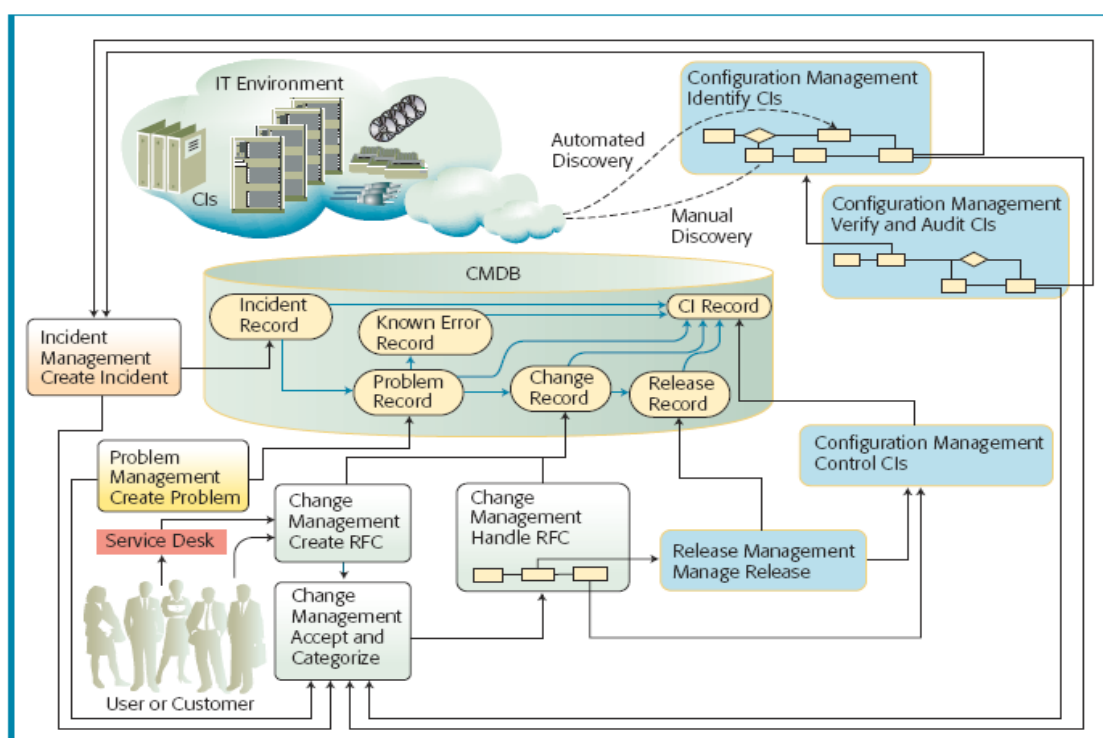
5) การจัดการอุบัติการณ์ (Incident Management)

การจัดการอุบัติการณ์ คือ กระบวนการนี้ช่วยในการสนับสนุนการแก้ไขอุบัติการณ์ (Incident) หรือแก้ไขเหตุการณ์ผิดปกติที่ทำให้การบริการต้องหยุดชะงัก โดยมุ่งหมายไปที่การแก้ไขปัญหาให้เร็วที่สุดโดยไม่ต้องสนใจสาเหตุที่แท้จริงก่อน

6) การจัดการปัญหา (Problem Management)

การจัดการปัญหา คือ กระบวนการที่ช่วยในการแก้ไขปัญหาจากสาเหตุที่แท้จริง ซึ่งต่างกับการจัดการอุบัติการณ์ที่มุ่งเน้นไปที่การกู้คืนสิ่งปัญหาให้กลับมาพร้อมใช้งานให้เร็วที่สุด

โดยกระบวนการนี้สามารถดำเนินการได้ทั้งเชิงรุกและเชิงรับ คือสามารถวางแผนป้องกัน  
 ปัญหาล่วงหน้าหรือการค้นหาสาเหตุที่แท้จริงของปัญหาและทำการป้องกันไม่ให้เกิดซ้ำอีก  
 กระบวนการในการสนับสนุนการบริการ แสดงในรูปที่ 2.1



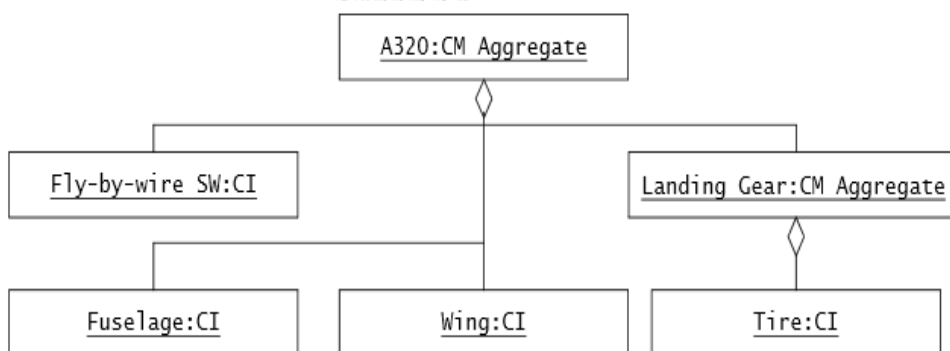
รูปที่ 2.1 แผนภาพกระบวนการในการสนับสนุนการบริการ [2]

จากรูปที่ 2.1 เมื่อผู้ใช้งานพบปัญหาการใช้งานจะแจ้งปัญหาไปยังเซอร์วิสเดสค์ หรือหากเกิด  
 ปัญหาขึ้นในระบบหากตรวจพบจากการมอนิเตอร์ จะควบคุมโดยการจัดการอุบัติการณ์เพื่อแก้ไข  
 โดยเร็วให้ระบบใช้งานได้ตามปกติ ซึ่งการแก้ไขระบบหรือกรณีผู้ใช้งานต้องการปรับปรุงระบบจะต้อง  
 ผ่านการจัดการการเปลี่ยนแปลง ซึ่งจะมีการสร้างคำร้องขอเปลี่ยนแปลง มีขั้นตอนการอนุมัติ ซึ่งเมื่อ  
 ผ่านการอนุมัติก็จะไปสู่การจัดการริสค์เพื่อกำหนดวันที่พลอยเมนต์ ส่วนการจัดการปัญหาจะสรุป  
 วางแผนป้องกันปัญหาจากอุบัติการณ์ที่เกิดขึ้นไม่ให้เกิดซ้ำอีก ผลการเปลี่ยนแปลงที่เกิดขึ้นและ  
 การบันทึกข้อมูลต่างๆ จะจัดบริหารโดยการจัดการโครงสร้างซึ่งจะจัดเก็บข้อมูลไว้ที่ซีไอ

โดยในงานวิจัยนี้จะนำทฤษฎีของไอทิลในด้านการจัดการโครงสร้างมาเป็นแนวทางในการ  
 ออกแบบฐานข้อมูลการจัดการโครงสร้าง (CMDB) คัดเลือกซีไอ และจัดการซีไอ

## 2.2 ซีโอและซีไอแอกกรีเกต [3]

ซีไอคือชิ้นงาน (Work Product) ที่เราพัฒนาขึ้นมาหรือเป็นหน่วยโครงสร้างพื้นฐานของระบบเทคโนโลยีสารสนเทศที่ดูแลอยู่ ซึ่งเราต้องการควบคุมหากเกิดการเปลี่ยนแปลงขึ้นกับซีไอนั้น ซีไออาจเป็นชิ้นงานเพียงชิ้นเดียวหรือประกอบขึ้นจากหลายๆ ชิ้นงานก็ได้ซึ่งเรียกซีไอที่รวมกันจากหลายชิ้นงานว่าซีไอแอกกรีเกต ยกตัวอย่างซีไอและซีไอแอกกรีเกต ดังรูปที่ 2.4 ซีไอแอกกรีเกต ของเครื่องบิน เอ320 จะประกอบไปด้วย ซีไอของชิ้นส่วนเครื่องบิน , ซีไอของซอฟต์แวร์การบิน , ซีไอของปีก , ซีไอแอกกรีเกตของอุปกรณ์ลงจอด ซึ่ง ซีไอแอกกรีเกตของอุปกรณ์ลงจอด ก็จะประกอบด้วย ซีไอล้อ เป็นต้น



รูปที่ 2.2 ตัวอย่างซีไอและซีไอแอกกรีเกต [3]

## 2.3 การจัดการโครงแบบและ ฐานข้อมูลการจัดการโครงแบบตามกรอบความคิดไอทิล [4]

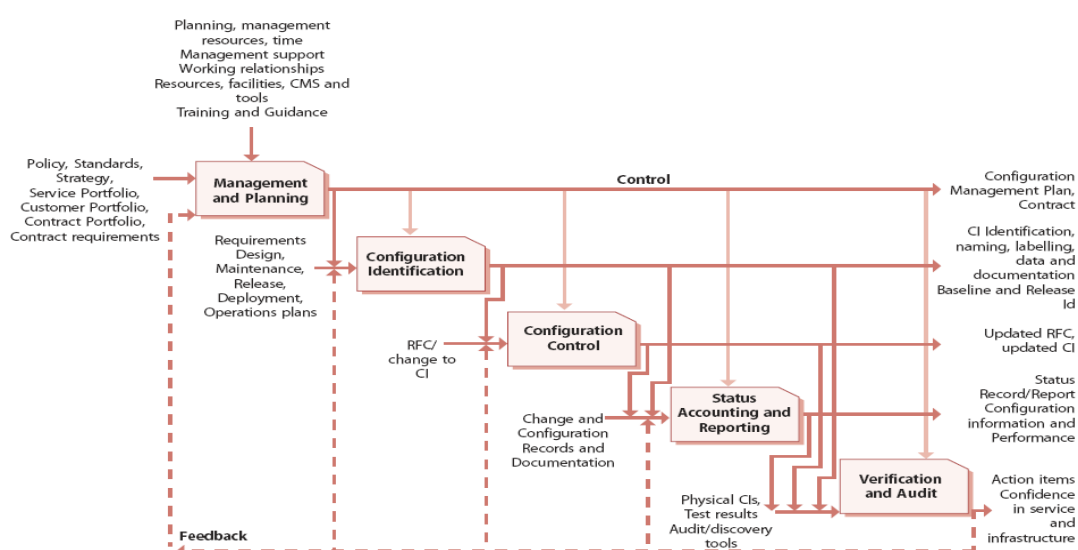
การจัดการโครงแบบคือการสร้างและบำรุงรักษาความเข้ากันได้ของชิ้นงานในระบบ โดยมีกระบวนการที่ช่วยจัดเก็บและจัดการข้อมูลซีไอให้เป็นปัจจุบันอยู่เสมอ โดยมีการจัดการและวางแผนมีการควบคุมเวอร์ชันการเปลี่ยนแปลงรวมทั้งตรวจสอบความถูกต้อง โดยข้อมูลต่างๆ จะรวบรวมและเก็บข้อมูลในรูปแบบของซีไอไว้ในฐานข้อมูลการจัดการโครงแบบ โดยที่ซีไอนั้นคือหน่วยโครงสร้างพื้นฐานของฐานข้อมูลการจัดการโครงแบบ

ฐานข้อมูลการจัดการโครงแบบเป็นฐานข้อมูลหลักของการจัดการด้านเทคโนโลยีสารสนเทศ ซึ่งเก็บรวบรวมรายละเอียดขององค์ประกอบขององค์กรที่ใช้ในการจัดการบริการเทคโนโลยีสารสนเทศและการสนับสนุนการบริการ เช่น ข้อมูลฮาร์ดแวร์ ซอฟต์แวร์ เอกสารและข้อมูลบุคคล เป้าหมายสำคัญของฐานข้อมูลการจัดการโครงแบบคือการช่วยให้องค์กรเข้าใจความสัมพันธ์ระหว่างองค์ประกอบต่างๆ และติดตามการกำหนดค่าของซีไอได้

ไอทีลได้แนะนำให้บูรณาการกระบวนการในการจัดการเปลี่ยนแปลงเข้ากับการจัดการโครงแบบ โดยทุกๆ คำร้องขอการเปลี่ยนแปลงต้องทำการบันทึกและเมื่อดำเนินการเปลี่ยนแปลงสำเร็จต้องทำการอัปเดตซีไอที่แก้ไขลงในฐานข้อมูลการจัดการโครงแบบด้วย

## 2.2.1 ระบบการจัดการโครงแบบ [4]

ระบบการจัดการโครงแบบ ประกอบไปด้วยกิจกรรมต่างๆ คือ การจัดการและวางแผน การระบุซีไอ การควบคุม การทำบัญชีโครงแบบและรายงาน การตรวจสอบและสอบทาน ดังรูปที่ 2.3 มีรายละเอียดดังนี้



รูปที่ 2.3 แผนภาพกิจกรรมในระบบการจัดการโครงแบบ [4]

จุฬาลงกรณ์มหาวิทยาลัย

1) การจัดการและวางแผน - เป็นการวางแผนครอบคลุมกลยุทธ์นโยบายขอบเขตวัตถุประสงค์บทบาทและความรับผิดชอบกระบวนการจัดการโครงแบบ รวมทั้ง กิจกรรมและขั้นตอนการสร้างฐานข้อมูลการจัดการโครงแบบ ความสัมพันธ์กับกระบวนการอื่นๆ กำหนดจำนวนประเภท ชิ้นโครงแบบ กำหนดขอบเขต และรายละเอียดของข้อมูลชิ้นโครงแบบและความลึกของข้อมูล

2) การระบุซีไอ - ส่วนนี้จะทำการคัดเลือก (Selection) ทำการยืนยัน (Identification) และ ทำฉลาก (Labeling) บนชิ้นโครงแบบทั้งหมด ซึ่งจะครอบคลุมถึงการบันทึกข้อมูลเกี่ยวกับชิ้นโครงแบบรวมทั้งฮาร์ดแวร์และซอฟต์แวร์ รุ่นเอกสาร การเป็นเจ้าของ และตัวบ่งชี้ที่ไม่ซ้ำกันอื่นๆ ชิ้นโครงแบบควรได้รับการบันทึกไว้ในระดับของรายละเอียดตามความต้องการทางธุรกิจโดยทั่วไปให้อยู่ในระดับของ "การเปลี่ยนแปลงที่เป็นอิสระ" ซึ่งรวมถึงการกำหนดความสัมพันธ์ของชิ้นโครงแบบในระบบ

กิจกรรมการระบุซีไออ้างอิงขั้นตอนตามกรอบงานของไอทิล [4] ระบุไว้ดังนี้

- 1) กำหนดเงื่อนไขสำหรับการคัดเลือกซีไอ
- 2) ระบุซีไอที่เกี่ยวข้องทั้งหมดและคัดกรองซีไอจากเงื่อนไขที่กำหนดไว้
- 3) กำหนดเลขยืนยันตนเฉพาะให้แก่ละซีไอ
- 4) กำหนดค่าคุณสมบัติให้แก่ละซีไอ
- 5) กำหนดว่าแต่ละซีไอจะเริ่มเข้าสู่การจัดการโครงแบบเมื่อไหร่
- 6) ยืนยันผู้รับผิดชอบแต่ละซีไอ

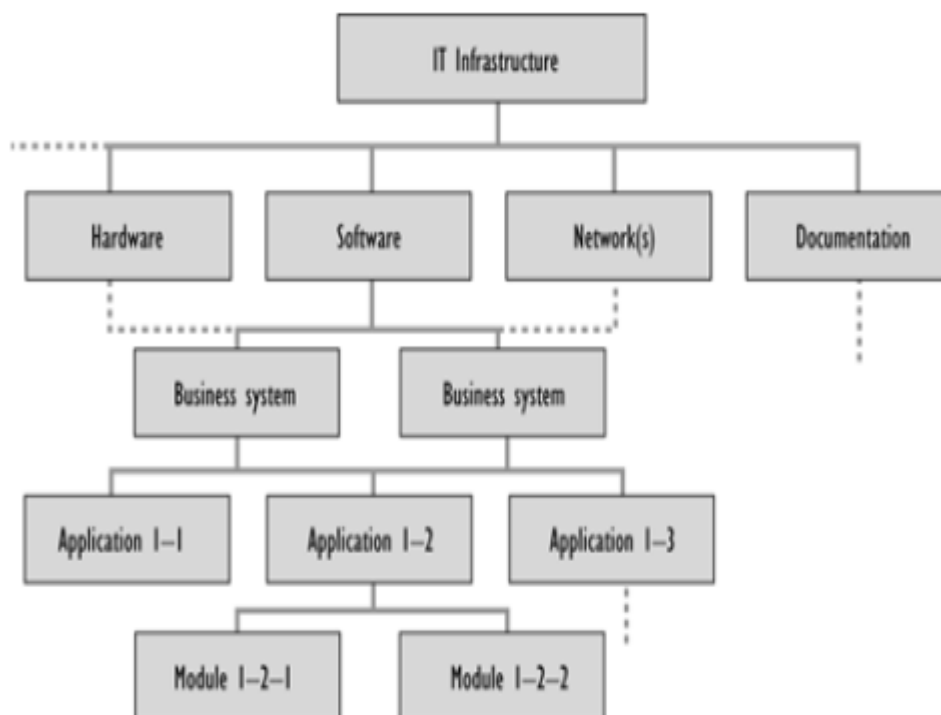
3) การควบคุม - ส่วนกิจกรรมนี้จะช่วยให้ความมั่นใจว่าต้องได้รับการอนุญาตที่สามารถระบุตัวตนก่อนที่ชั้นโครงแบบจะได้รับการบันทึกหรือเปลี่ยนแปลง มีเอกสารการควบคุมที่เหมาะสม ข้อกำหนดการปรับปรุงชั้นโครงแบบทั้งหมดจะอยู่ภายใต้การจัดการการเปลี่ยนแปลง

4) การทำบัญชีโครงแบบและรายงาน - ส่วนกิจกรรมนี้จะติดตามการเปลี่ยนแปลงชั้นโครงแบบตลอดวงจรผ่านการบันทึกสถานะต่างๆ และการแสดงผลสถานะเพื่อรายงาน

5) การตรวจสอบและสอบทาน - คือการตรวจสอบการดำรงอยู่ของชั้นโครงแบบ และตรวจสอบว่าชั้นโครงแบบจะถูกบันทึกไว้อย่างถูกต้องในฐานข้อมูลการจัดการโครงแบบ ซึ่งจะรวมถึงขั้นตอนของการตรวจสอบการจัดการรีลีส และเอกสารการจัดการโครงแบบก่อนที่จะมีการเปลี่ยนแปลง

2.2.2 โครงสร้างแบบแยกย่อยของโครงแบบ (Configuration Breakdown Structure) และการคัดเลือกซีไอ [5]

ไอทิลได้แนะนำว่าการเลือกซีไอควรเลือกโดยการใช้กระบวนการจำแนกย่อยจากรายการระดับบนสุดลงมาโดยขึ้นอยู่กับความต้องการทางธุรกิจและการบริการ โดยการทำโครงสร้างแบบแยกย่อยของโครงแบบ ซึ่งเป็นการวิเคราะห์ข้อมูลพื้นฐานที่อยู่ในโครงแบบเพื่อกำหนดส่วนประกอบพื้นฐาน ดังตัวอย่างรูปที่ 2.4 โครงสร้างแบบแยกย่อยของโครงสร้างพื้นฐานในระบบเทคโนโลยีสารสนเทศ



รูปที่ 2.4 ตัวอย่างโครงสร้างแบบแยกย่อย [5]

จากรูปที่ 2.4 โครงสร้างแบบแยกย่อยในชั้นแรกจะประกอบไปด้วย Hardware, Software, Network และ Document ในชั้นที่ 2 จะประกอบไปด้วย Business System ซึ่งเป็นลูก (child) ของ Software ในรูปจะมี 2 Business System และ เส้นประในรูปแสดงถึงการมีความสัมพันธ์ระหว่างกัน ซึ่งตามรูปตัวอย่าง Business System จะมีความสัมพันธ์กับ Hardware และ Network ในงานวิทยานิพนธ์นี้จะนำวิธีการสร้างโครงสร้างแบบแยกย่อยของโครงแบบมาประยุกต์ใช้เพื่อการคัดเลือกซีไอเพื่อให้มีความสามารถหลายมุมมองและจะช่วยปรับปรุงการวิเคราะห์ระบบได้ดีขึ้น

สำหรับคุณสมบัติทั่วไปของซีไอ ไอตีลได้แนะนำให้มีความสมบัติดังแสดงตัวอย่างในตารางที่ 2.1 จะมีคุณสมบัติเช่น Unique identifier คือข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน Name คือชื่อของซีไอ เช่น ซีไอชื่อซอฟต์แวร์วินโดว 10 Description คือรายละเอียดของซีไอ เช่น เป็นซอฟต์แวร์ระบบปฏิบัติการ Location คือตำแหน่งหรือสถานที่ Version คือเวอร์ชันของซีไอ Status คือสถานะของซีไอ เช่น ใช้งานอยู่หรือไม่ใช้งานแล้ว และ Related document masters คือมีความสัมพันธ์ไปยังเอกสารใดๆ เป็นต้น ซึ่งในงานวิจัยนี้จะเลือกนำเอาคุณสมบัติทั่วไปของซีไอตามทีไอตีลแนะนำมาเป็นคุณสมบัติพื้นฐานของซีไอ

ตารางที่ 2.1 คุณสมบัติทั่วไปของซีไอ [4]

| คุณลักษณะ                | ความหมาย   |
|--------------------------|--|
| Unique identifier        | ข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน                                 |
| CI Type                  | ประเภทของซีไอ  |
| Name                     | ชื่อของ ซีไอ   |
| Description              | รายละเอียด   |
| Location                 | ตำแหน่งหรือสถานที่   |
| Supply date              | วันที่ให้บริการ  |
| Version                  | เวอร์ชัน   |
| License detail           | รายละเอียดของลิขสิทธิ์   |
| Owner/custodian          | เจ้าของหรือผู้ดูแล   |
| Status                   | สถานะ  |
| Supplier/source          | ซัพพลายเออร์หรือแหล่งข้อมูล  |
| Related document masters | เอกสารที่เกี่ยวข้อง  |
| Related software masters | ซอฟต์แวร์ที่เกี่ยวข้อง   |
| Historical data          | ข้อมูลบันทึกประวัติ  |
| Relationships type       | ประเภทของความสัมพันธ์  |
| Applicable SLA.          | เงื่อนไขข้อตกลงการให้บริการ (Service level) (Agreement) ที่เหมาะสม |

ในส่วนของ Software Configuration Management [6] ได้ระบุว่าซีไอคือชิ้นส่วนต่างๆในกระบวนการของวิศวกรรมซอฟต์แวร์ เช่น เอกสารหรือซอร์สโค้ด ยกตัวอย่างเช่น เอกสารการทดสอบ เอกสารการออกแบบ โปรแกรม เป็นต้น คุณสมบัติเบื้องต้นที่ควรมีของซีไอ เช่น ชื่อซีไอ รายละเอียดของซีไอ รายการต่างๆ ของทรัพยากร ประเภทซีไอว่าเป็นเอกสาร โปรแกรม หรือข้อมูล หมายเลขเฉพาะเจาะจงที่ไม่ซ้ำกัน ข้อมูลเวอร์ชัน เป็นต้น

### 2.3 ออนโทโลยี และ ภาษาอาวล์ (OWL: Web Ontology Language)

ออนโทโลยี [7] คือการนิยามหรือกำหนดรูปแบบโครงสร้างของสิ่งที่สนใจ เช่น วัตถุ สิ่งของ หรือสิ่งมีชีวิต ให้มีความหมายตามขอบเขตขององค์ความรู้ ซึ่งออนโทโลยีได้ถูกกล่าวถึงในหลายองค์กร โดยเฉพาะด้านปัญญาประดิษฐ์นั้นมีการใช้งานออนโทโลยีมานานแล้ว โดยในปัจจุบันได้ถูกนำมาประยุกต์ใช้ในมาตรฐานของการออกแบบจำลองโครงสร้างของเอกซ์เอ็มแอล (XML) และการนิยามรูปแบบแนวคิดของโครงสร้างของฐานข้อมูล ออนโทโลยีเป็นเทคโนโลยีทางการพัฒนาภาษาเชิง



ความหมายสมัยใหม่ โดยเป็นภาษาที่เครื่องคอมพิวเตอร์สามารถตีความหมายและทำตามคำสั่งได้ โดยลักษณะของออนโทโลยี คือการนิยามรูปแบบ (model) ภายในขอบเขตขององค์ความรู้เพื่ออธิบายสิ่งที่สนใจในโดเมน ให้ได้ใจความและถูกต้องมากที่สุด

ภาษาอวาล์ [8] เป็นภาษาหนึ่งที่ใช้สร้างข้อมูลออนโทโลยี ภาษาอวาล์ถูกสร้างโดย W3C Web Ontology Working Group ถูกพัฒนาขึ้นเพื่อเป็นส่วนขยายต่อจากภาษาอาร์ดีเอฟ (RDF: Resource Description Framework) และสืบทอดมาจากภาษาตราปา (DRAPA) ภาษาอวาล์จัดเป็นองค์ประกอบหนึ่งในงานเว็บเชิงความหมาย (Semantic Web) ที่ใช้ในการบรรยายข้อมูลเชิงความหมาย สามารถกำหนดโครงสร้างข้อมูลในลักษณะลำดับชั้น และอธิบายข้อมูลที่มีความสัมพันธ์ในระบบฐานข้อมูลได้รวมทั้งสามารถรองรับการบรรยายข้อมูลเชิงตรรกะ ชนิดข้อมูล และตัวบ่งปริมาณได้ ทำให้ข้อมูลที่ถูกแทนที่นั้นมีความหมายมากยิ่งขึ้น

ลักษณะการบรรยายจะอยู่ในรูปของคลาส และคุณสมบัติของคลาส เพื่ออธิบายเอนทิตีและความสัมพันธ์ (Relationship) ต่าง ๆ ที่เกิดขึ้น ยกตัวอย่าง เว็บไซต์ซึ่งเป็นแหล่งรวบรวมข้อมูลต่างๆ ต้องใช้คำสำคัญในการค้นคว้า ซึ่งข้อมูลที่เก็บอยู่ในปัจจุบันนั้นยังขาดการจัดการที่ดีเนื่องจากไม่มีเครื่องมือที่จะช่วยสนับสนุน เพื่อใช้ในการเชื่อมโยงข้อมูลให้อยู่ในรูปแบบที่กำหนดตัวกลางที่จะใช้ในการคำนวณนี้ ต้องการเครื่องมือที่สามารถอ่านคำอธิบายของหัวข้อต่างๆ ในเว็บที่เป็นทรัพยากรและคำอธิบายที่ได้มานี้จะต้องอยู่ในรูปแบบที่มนุษย์สามารถเข้าใจได้ด้วย

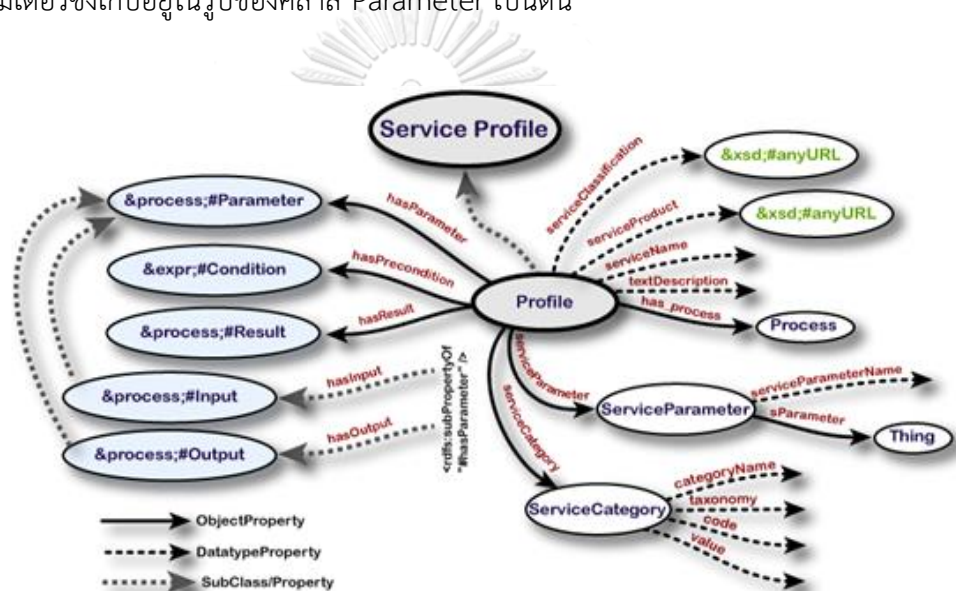
ภาษาอวาล์ถูกสร้างขึ้นมาเพื่อที่สามารถอธิบายคลาสและความสัมพันธ์ของคลาสที่อยู่ในข้อมูลเว็บ และโปรแกรมประยุกต์ออนไลน์ ต่างๆ โดยสามารถที่จะสร้างข้อมูลภาษาอวาล์ได้ดังนี้ คือจัดรูปแบบของโดเมนโดยการสร้างคลาสและคุณสมบัติของคลาส สร้างเอกลักษณ์เฉพาะและคุณลักษณะเฉพาะของแต่ละคลาส ให้เหตุผลเกี่ยวกับแต่ละคลาสและแต่ละเอกลักษณ์เพื่อสร้างลำดับชั้น โดยความหมายที่เป็นทางการของภาษาอวาล์จะเน้นถึงการทำให้ได้มาซึ่งหลักตรรกะแต่ถูกวางเงื่อนไขโดยใช้ความหมาย ซึ่งเงื่อนไขนี้อาจจะได้มาจากเอกสารใดเอกสารหนึ่งหรือเอกสารที่อยู่กันอย่างกระจัดกระจายหลายๆ ที่ก็เป็นไปได้

โครงสร้างคลาสภาษาอวาล์จะมีประเภทของคุณสมบัติดังนี้

- 1) คุณสมบัติแบบชนิดข้อมูล (Datatype Property) จะเก็บคุณสมบัติในรูปแบบของชนิดข้อมูล เช่น ข้อมูลแบบสตริง (String) ข้อมูลแบบเลขจำนวนเต็ม (Integer) เป็นต้น
- 2) คุณสมบัติแบบวัตถุ (Object Property) จะเก็บคุณสมบัติในรูปแบบของความสัมพันธ์ไปยังคลาสอื่น

ดังรูปที่ 2.5 จะแสดงความสัมพันธ์และความหมายการออกแบบออนโทโลยีในการให้บริการเว็บเซอร์วิส ว่ามีส่วนประกอบอะไรบ้างและเส้นเชื่อมที่จะแสดงคุณสมบัติและความสัมพันธ์ของแต่ละชั้นส่วน โดยมีรายละเอียด เช่น

1. Service Profile จะเป็น คลาสเพิ่มข้อมูลเว็บเซอร์วิส ซึ่งเป็นคลาสหลักที่เก็บรายละเอียดข้อมูลต่างๆ Profile จะเป็น คลาสลูก (Subclass) ของ Service Profile
2. Profile จะมีคุณสมบัติแบบชนิดข้อมูล คือ ServiceClassification, ServiceProduct, ServiceName, TextDescription ไว้เก็บข้อมูลต่างๆ ในรูปข้อมูลตัวอักษร
3. Profile จะมีคุณสมบัติแบบวัตถุ เช่น hasParameter ความหมายคือ คลาส Profile มีค่าพารามิเตอร์ซึ่งเก็บอยู่ในรูปของคลาส Parameter เป็นต้น



รูปที่ 2.5 ตัวอย่างการออกแบบออนโทโลยีในการให้บริการเว็บเซอร์วิส [9]

```

<owl:ObjectProperty rdf:ID="hasParameter">
  <rdfs:domain rdf:resource="#Process"/>
  <rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasInput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
  <rdfs:range rdf:resource="#Input"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasOutput">
  <rdfs:subPropertyOf rdf:resource="#hasParameter"/>
  <rdfs:range rdf:resource="#Output"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="resultForm">
  <rdfs:label>resultForm</rdfs:label>
  <rdfs:domain rdf:resource="#Result"/>
  <rdfs:range rdf:resource="&rdf;#XMLLiteral"/>
</owl:DatatypeProperty>

```

รูปที่ 2.6 ตัวอย่างข้อมูลภาษาอวาลในการออกแบบการให้บริการเว็บเซอร์วิส [9]

จากรูปที่ 2.6 แสดงภาษาอวาล์ที่ได้จากรูปที่ 2.5 โดยลักษณะของภาษาอวาล์มีการเก็บข้อมูลเป็น แสดงการเขียนข้อมูลคุณสมบัติแบบวัตถุชื่อ hasParameter จะมี “domain” เป็นต้นทางของลูกศร เป็นคลาสชื่อ Profile เชื่อมความสัมพันธ์ไปยังปลายลูกศรซึ่งระบุค่าใน “Range” คือคลาสชื่อ Parameter ส่วนคุณสมบัติแบบวัตถุชื่อ hasInput กับ hasOutput จะเป็นคุณสมบัติลูก (SubclassProperty) ของ คุณสมบัติแบบวัตถุชื่อ hasParameter โดยจะสืบทอดคุณสมบัติแบบวัตถุ ของ hasParameter มาทั้งหมดและ มีความสัมพันธ์เพิ่มคือ คุณสมบัติแบบวัตถุชื่อ hasInput มี “domain” คือคลาสชื่อ Profile มี “Range” คือคลาสชื่อ Input และ hasOutput มี “domain” คือคลาสชื่อ Profile มี “Range” คือคลาสชื่อ Output และสุดท้ายคุณสมบัติแบบ ชนิดข้อมูลชื่อ ResultFrom มี “domain” คือคลาสชื่อ Result มี “Range” คือ XMLLiteral คือ เป็นชนิดข้อมูลแบบเอกซ์เอ็มแอล

## 2.4 โปรแกรมโพรเทจ (Protégé Program)

โปรแกรมโพรเทจ [10] พัฒนาโดยมหาวิทยาลัยสแตนฟอร์ด โดยเป็นโอเพนซอร์ส แพลตฟอร์มที่มีชุดของเครื่องมือที่ช่วยในการสร้างแบบจำลองโดเมนและโปรแกรมประยุกต์เชิงความรู้ ให้เป็นภาษาออนโทโลยี และมีเครื่องมือที่สามารถช่วยสร้างโครงสร้างต่างๆ ของภาษาอวาล์ คือ คลาส คุณสมบัติแบบชนิดข้อมูล และ คุณสมบัติแบบวัตถุ ได้

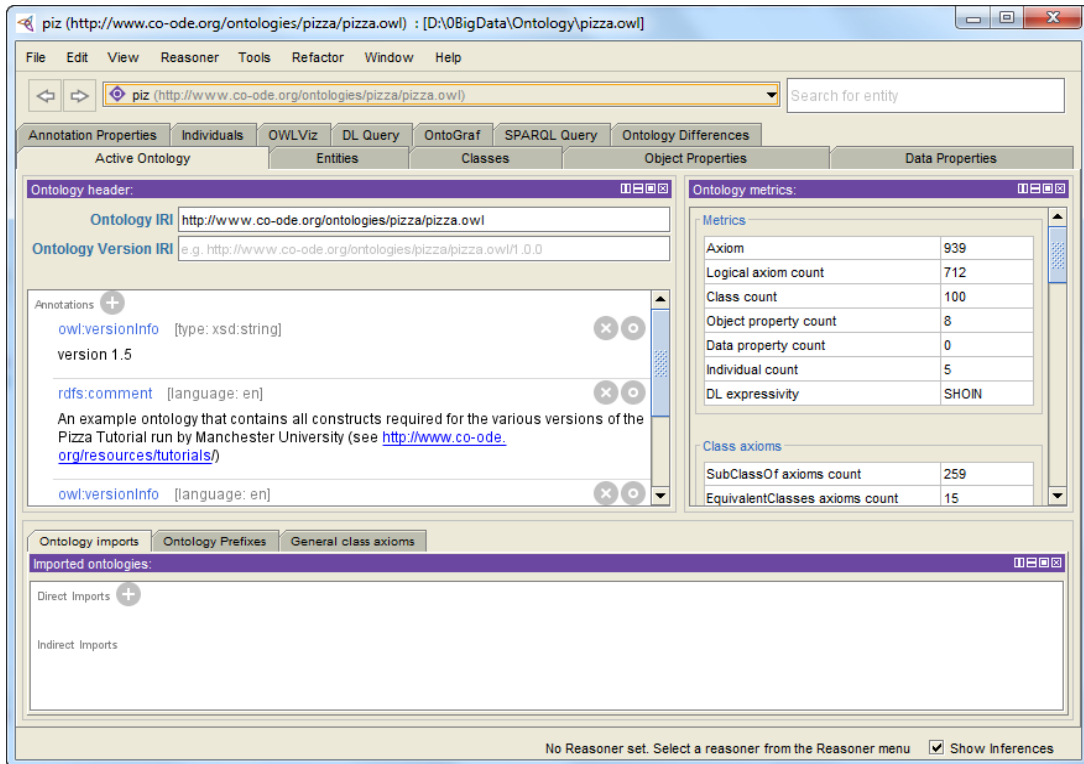
ตัวอย่างการสร้าง ข้อมูลภาษาอวาล์ใหม่เพื่อสร้างข้อมูลออนโทโลยีของพิซซ่า

- 1) เปิดโปรแกรมโพรเทจ
- 2) เมื่อขึ้นข้อความต้อนรับ ให้กดเลือกเมนู "Create New OWL Ontology"
- 3) จะปรากฏข้อความ "Create Ontology URI Wizard" ให้ป้อนชื่อ URI ที่จะตั้งชื่อเช่น "http://www.pizza.com/ontologies/pizza.owl"
- 4) บันทึกออนโทโลยี จะปรากฏหน้าจอดังรูปที่ 2.7

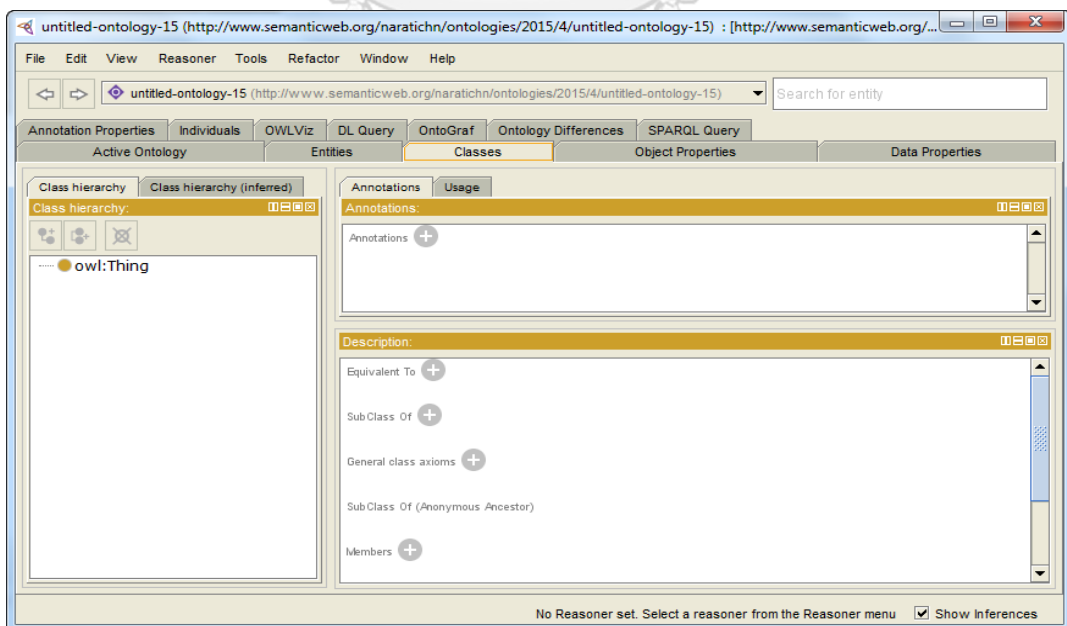
ใน โปรแกรมโพรเทจเวอร์ชัน 4 การแก้ไขคลาสจะอยู่ในส่วนของเมนูชื่อ "Classes" ดังในรูป ที่ 2.8 และรูปที่ 2.9 คลาสของภาษาอวาล์จะนำเสนอถึงเซตของวัตถุ โดยที่คลาส "Thing" จะเป็น เซตหลักของทุกเซต ขั้นตอนการสร้างคลาสทำได้ดังนี้

- 1) เลือกส่วน "Classes" จากเมนูตามรูปที่ 2.8
- 2) ตามรูปที่ 2.9 กดปุ่ม "Add Subclass"
- 3) ตามรูปที่ 2.10 ป้อนชื่อที่ต้องการ

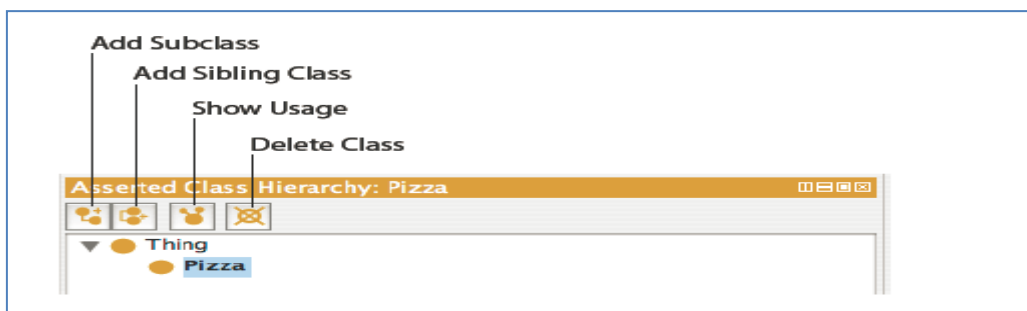
4) วนป้อนคลาสให้ครบจะได้ผลลัพธ์ตามรูปที่ 2.11



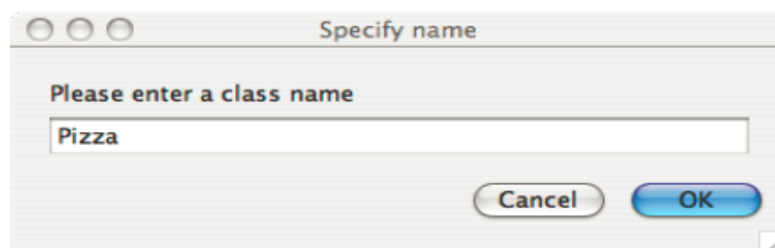
รูปที่ 2.7 ตัวอย่างการสร้างข้อมูลภาษาอวาล์เพื่อสร้างข้อมูลออนโทโลยีของพิซซ่า [10]



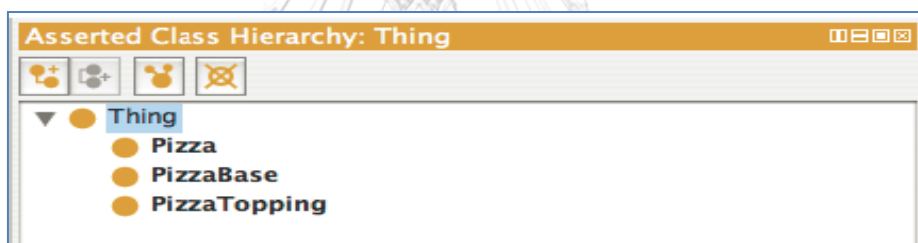
รูปที่ 2.8 ส่วน "Classes" บนโปรแกรมโปรเตจ [10]



รูปที่ 2.9 รายละเอียดของปุ่มใน ส่วน "Classes" บนโปรแกรมโพรเทจ [10]



รูปที่ 2.10 หน้ากรอกข้อมูลการสร้างคลาสในส่วน "Classes" บนโปรแกรมโพรเทจ [10]



รูปที่ 2.11 ตัวอย่างข้อมูลคลาสที่ได้เมื่อสร้างข้อมูล ontology โดยพีชชาบนโปรแกรมโพรเทจ [10]

CHULALONGKORN UNIVERSITY

## 2.5 ภาษาสพาควอล (SPARQL)

ภาษาสพาควอล [11] เป็นภาษาสำหรับดึงข้อมูลมาแสดง (Query Language) ซึ่งอยู่บนพื้นฐานของข้อมูลกราฟ ซึ่งมีลักษณะในรูปแบบของภาษาอาร์ดีเอฟ (RDF) และภาษาอาวด์ กราฟที่มีลักษณะที่ง่ายที่สุดก็ถือเป็นกราฟในรูปแบบของทริเปิ้ล (Triple) ที่ประกอบด้วย ประธาน (Subject) ส่วนขยาย (Predicate) และ วัตถุ (Object)

ตัวอย่างเช่น ต้องการจะค้นหาหัวข้อเรื่องของหนังสือจากข้อมูลที่ให้ในลักษณะของกราฟที่เป็นอาร์ดีเอฟ ซึ่งภาษาจะดึงข้อมูลนี้แบ่งเป็นสองส่วนคือ SELECT และ WHERE ซึ่ง SELECT จะไปอธิบายตัวแปรที่จะไปปรากฏบนผลลัพธ์ และ WHERE จะเป็นเงื่อนไข ดังแสดงในรูปที่ 2.12

**Data:**

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
```

**Query:**

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

**Query Result:**

| title             |
|-------------------|
| "SPARQL Tutorial" |

รูปที่ 2.12 ตัวอย่างการเขียนภาษาสพาควอล [11]

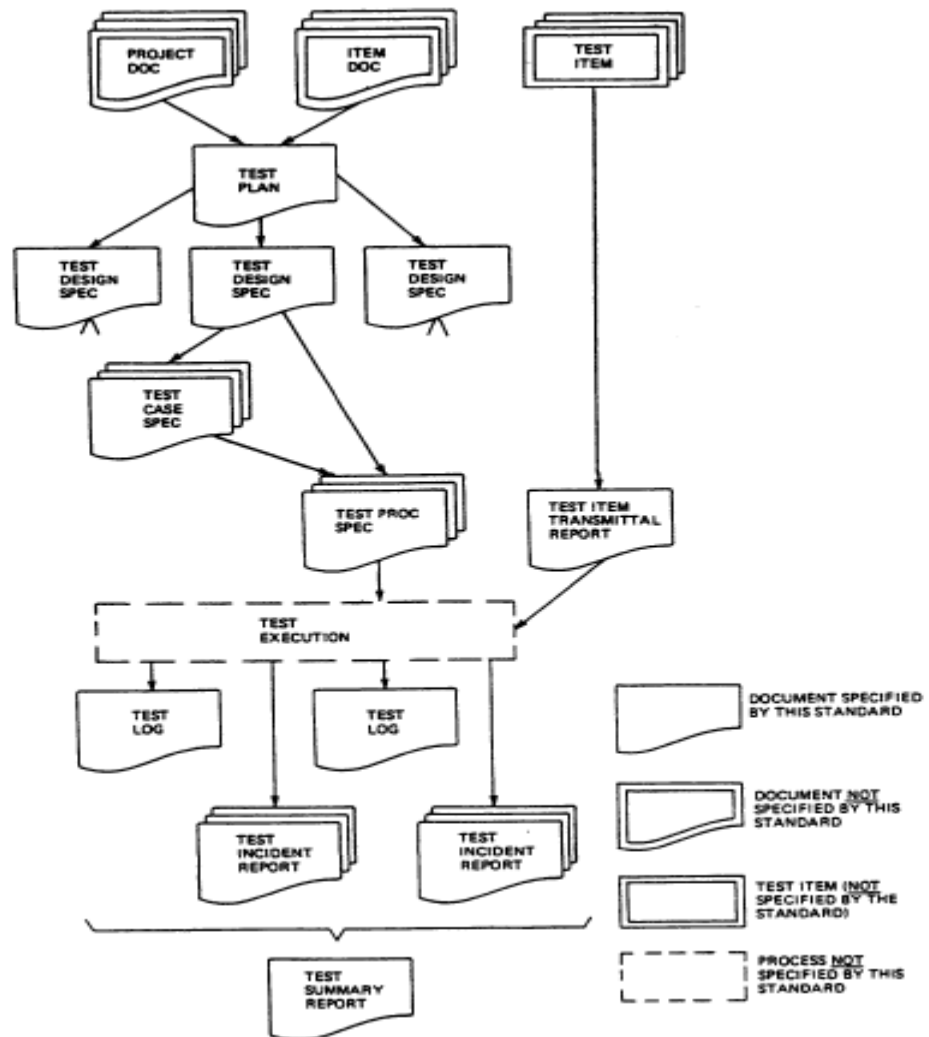
## 2.6 เอกสารการทดสอบ [12]

การทดสอบซอฟต์แวร์ เป็นกระบวนการที่จะช่วยให้ซอฟต์แวร์ที่พัฒนาขึ้นมี ความถูกต้อง ความสมบูรณ์ มีความปลอดภัย และมีคุณภาพที่ดี ซึ่งในกระบวนการพัฒนาซอฟต์แวร์ก่อนที่โปรแกรมจะดีพลอยเม้นท์จำเป็นต้องมีการทดสอบ ซึ่งในงานวิจัยนี้จะเก็บข้อมูลชื่อของเอกสารทดสอบ โดยความสัมพันธ์ของเอกสารการทดสอบแสดงในรูปที่ 2.13 โดยแสดงความสัมพันธ์ของเอกสารการทดสอบซึ่งประกอบไปด้วยเอกสาร คือ

- 1) แผนการทดสอบ (Test Plan) คือมุมมองแบบกว้างที่ระบุว่าการทดสอบจะเกิดขึ้นในลักษณะไหน ใครเป็นคนทำ โครงสร้างของเอกสารประกอบด้วย เช่น หมายเลขไอดีของแผนการทดสอบ (Test plan Identifier) จุดมุ่งหมายของแผน รายการทดสอบ (Test items) คุณลักษณะที่จะนำมาทดสอบ (Features to be tested) คุณลักษณะที่จะไม่นำมาทดสอบ (Features not to be tested) สภาพแวดล้อมที่ต้องการ (Environment Need) เงื่อนไขของการผ่านหรือไม่ผ่านการทดสอบ (Item pass/fail criteria) กำหนดเวลาในการทดสอบ (Schedule) เป็นต้น
- 2) เอกสารการออกแบบการทดสอบ (Test Design Specification) ประกอบด้วย รายละเอียดของเงื่อนไขการทดสอบที่จะนำไปใช้รวมถึงผลที่ต้องการในลักษณะที่เป็นแบบฟอร์มโดยทั่วไป
- 3) เอกสารกรณีทดสอบ (Test Case Specification) เอกสารทดสอบที่ระบุกรณีที่ใช้ทดสอบ โครงสร้างของเอกสารจะประกอบไปด้วย เช่น ข้อมูลไอดีของกรณีทดสอบ (Test case specification identifier) รายการทดสอบ (Test items) ข้อกำหนดข้อมูลนำเข้า (Input Specification) ข้อกำหนดข้อมูลส่งออก (Output Specification) สภาพแวดล้อมที่ต้องการ ขั้นตอนพิเศษของความต้องการ (Special procedural requirements) ข้อพึงพิงระหว่างกัน (Intercase dependencies) เป็นต้น

- 4) เอกสารขั้นตอนการทดสอบ (Test Procedure Specification) – เอกสารเพื่ออธิบายขั้นตอนการทดสอบ ให้ทดสอบได้ทราบว่าจะต้องดำเนินการทดสอบอย่างไรบ้าง รวมถึงการเตรียมสภาพแวดล้อมก่อนเริ่มทำการทดสอบ
- 5) เอกสารบันทึกการทดสอบ (Test Log) เป็นเอกสารที่เก็บผลการทดสอบ
- 6) เอกสารรายงานอุบัติการณ์ของการทดสอบ (Test Incident Report) เป็นเอกสารที่สรุปปัญหาดีเฟกต์ที่ทดสอบพบ เพื่อนำไปวิเคราะห์หาสาเหตุต่อไป
- 7) เอกสารสรุปการทดสอบ (Test Summary Report) เป็นเอกสารที่สรุปผลการทดสอบทั้งหมด ตั้งแต่การวางแผน กิจกรรมต่างๆ ที่ทำ และปัญหาดีเฟกต์ที่พบ และระบุการประเมินจากผลการทดสอบทั้งหมด

จากเอกสารข้างต้นที่กล่าวมานี้ ในวิทยานิพนธ์นี้จะสนใจในส่วนการนำเอกสารแผนการทดสอบ เอกสารข้อกำหนดการทดสอบ มาใช้ในระบบการจัดการโครงแบบเท่านั้น



รูปที่ 2.13 ความสัมพันธ์ของเอกสารการทดสอบ [12]

## 2.7 ระบบสารสนเทศ (Information system)

ระบบสารสนเทศ หมายถึง ระบบที่ประกอบด้วยส่วนต่างๆ ได้แก่ ระบบคอมพิวเตอร์ทั้ง ฮาร์ดแวร์ ซอฟต์แวร์ ระบบเครือข่าย ฐานข้อมูล ผู้พัฒนาระบบ ผู้ใช้ระบบ พนักงานที่เกี่ยวข้อง และผู้เชี่ยวชาญในสาขา ทุกองค์ประกอบนี้ทำงานร่วมกันเพื่อกำหนด รวบรวม จัดเก็บข้อมูล ประมวลผลข้อมูลเพื่อสร้างสารสนเทศ และส่งผลลัพธ์หรือสารสนเทศที่ได้ให้ผู้ใช้เพื่อช่วยสนับสนุนการทำงาน การตัดสินใจ การวางแผน การบริหาร การควบคุม การวิเคราะห์และติดตามผลการดำเนินงานขององค์กร [13]

### 2.7.1 เครือข่ายคอมพิวเตอร์และฮาร์ดแวร์ (Hardware)

เครือข่ายคอมพิวเตอร์ คือ การนำกลุ่มคอมพิวเตอร์และอุปกรณ์ต่าง ๆ มาเชื่อมต่อกันเป็นเครือข่าย โดยใช้สื่อกลางซึ่งเป็นสายเคเบิลหรือคลื่นวิทยุเป็นเส้นทางการลำเลียงข้อมูลเพื่อสื่อสารระหว่างกัน โดยเครือข่ายมีการแบ่งประเภทตามระยะการใช้งานดังนี้

#### 1) เครือข่ายท้องถิ่น (Local Area Network: LAN)

เครือข่ายท้องถิ่นเป็นเครือข่ายส่วนบุคคล ที่มีการลิงค์เชื่อมโยงระหว่างพีซีคอมพิวเตอร์และอุปกรณ์เพื่อการใช้งานร่วมกัน เครือข่ายท้องถิ่นอาจมีเพียงพีซีคอมพิวเตอร์เพียง 2 เครื่องเพื่อใช้งานตามบ้านเรือน หรือเชื่อมโยงพีซีคอมพิวเตอร์เป็นร้อยเครื่องสำหรับองค์กรขนาดใหญ่ โดยจะครอบคลุมระยะทางไม่กี่กิโลเมตร เครือข่ายท้องถิ่นหรือมักเรียกสั้น ๆ ว่า เครือข่ายแลน นั้น ได้รับการออกแบบมาเพื่ออนุญาตให้สามารถแชร์ทรัพยากรบนเครือข่ายร่วมกันได้ เช่น การแชร์ข้อมูล โปรแกรม และเครื่องพิมพ์ เป็นต้น

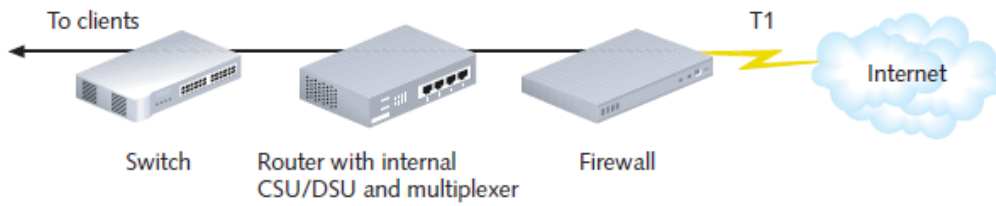
#### 2) เครือข่ายระดับเมือง (Metropolitan Area Network: MAN)

เป็นเครือข่ายที่มีขนาดระหว่างเครือข่ายแลนและเครือข่ายแวน ซึ่งปกติจะครอบคลุมพื้นที่ภายในเมืองหรือจังหวัด โดยเป็นเครือข่ายที่ออกแบบมาเพื่อให้ลูกค้าสามารถเชื่อมต่อใช้งานเพื่อการสื่อสารความเร็วสูง

#### 3) เครือข่ายบริเวณกว้าง (Wide Area Network: WAN)

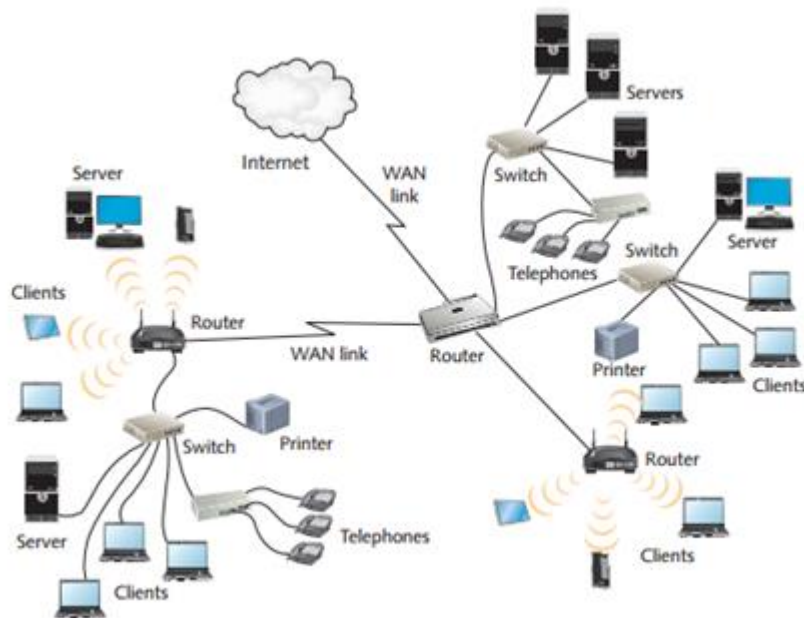
เครือข่ายระดับประเทศหรือเครือข่ายแวนสามารถส่งผ่านข้อมูลได้ระยะไกล สามารถสื่อสารข้ามประเทศหรือข้ามทวีปได้ เครือข่ายแวนอาจมีสายแกนหลักจำนวนมากว่าหนึ่งเส้นที่นำไปใช้เชื่อมต่อเข้ากับอินเทอร์เน็ต โดยส่วนใหญ่แล้วมักนิยมเรียก เครือข่ายระดับเมืองรวมกับเครือข่ายระดับประเทศ เป็น “WAN”





รูปที่ 2.14 ตัวอย่างเครือข่ายและอุปกรณ์ในการเชื่อมต่อเบื้องต้น [14]

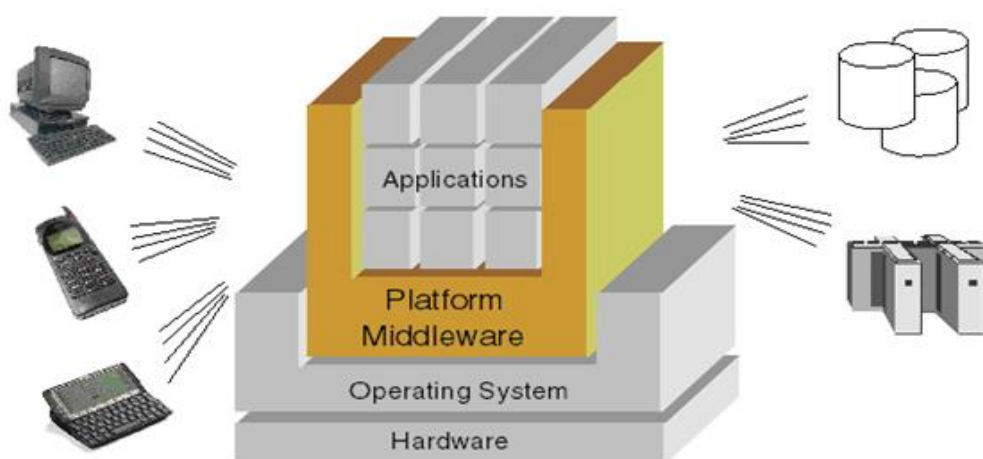
ตัวอย่างรูปที่ 2.14 แสดงอุปกรณ์และการเชื่อมต่อเครือข่ายเบื้องต้น และรูปที่ 2.15 แสดงการเชื่อมต่อเครือข่ายระหว่างกันจากเครือข่ายท้องถิ่นออกเชื่อมต่อกันด้วยเครือข่ายบริเวณกว้าง อุปกรณ์การเชื่อมต่อจะประกอบไปด้วย ไฟร์วอลล์ (Firewall) คืออุปกรณ์ที่ทำหน้าที่ตรวจสอบและกำหนดเงื่อนไขในการไหลผ่านของข้อมูลบนเครือข่ายคอมพิวเตอร์ เราเตอร์ (Router) คืออุปกรณ์ที่ทำหน้าที่เชื่อมต่อและกำหนดการไหลผ่านของข้อมูลบนเครือข่ายคอมพิวเตอร์ และสวิตช์ (Switch) คืออุปกรณ์ที่ทำหน้าที่เชื่อมต่อระบบเครือข่ายคอมพิวเตอร์ที่มีส่วนต่อประสานกับเครื่องลูกข่าย



รูปที่ 2.15 ตัวอย่างเครือข่ายบริเวณกว้าง [14]

## 2.7.2 ซอฟต์แวร์

ซอฟต์แวร์คือโปรแกรมหรือชุดคำสั่งที่ถูกเขียนขึ้นมาจากภาษาคอมพิวเตอร์และถูกแปลงให้เป็นภาษาเครื่องเพื่อให้คอมพิวเตอร์ทำงานอย่างใดอย่างหนึ่งตามที่เราต้องการ ซอฟต์แวร์จะแบ่งออกเป็นประเภทใหญ่ ๆ คือ ซอฟต์แวร์ระบบ (System Software) และซอฟต์แวร์ประยุกต์ (Application Software) [15] ซึ่งมีรายละเอียดแสดงดังรูปที่ 2.16 มีรายละเอียดดังนี้



รูปที่ 2.16 ชั้นของซอฟต์แวร์ประเภทต่างๆ [16]

### 1). ซอฟต์แวร์ระบบ

ซอฟต์แวร์ระบบ คือ โปรแกรมที่มีหน้าที่ควบคุมการทำงานของฮาร์ดแวร์และอำนวยความสะดวกให้กับผู้ใช้เครื่องคอมพิวเตอร์ แบ่งออกเป็นโปรแกรมตามหน้าที่การทำงานดังนี้ โปรแกรมระบบปฏิบัติการ (OS: Operating System) โปรแกรมระบบที่ทำหน้าที่ในการอำนวยความสะดวก (Utility Software) โปรแกรมตัวแปลภาษา มิดเดิลแวร์ (Middleware) หรือ โปรแกรมที่ทำหน้าที่สนับสนุนให้ออปพลิเคชันต่างๆ สามารถทำงานร่วมกันหรือแลกเปลี่ยนข้อมูลระหว่างกันได้ เป็นต้น

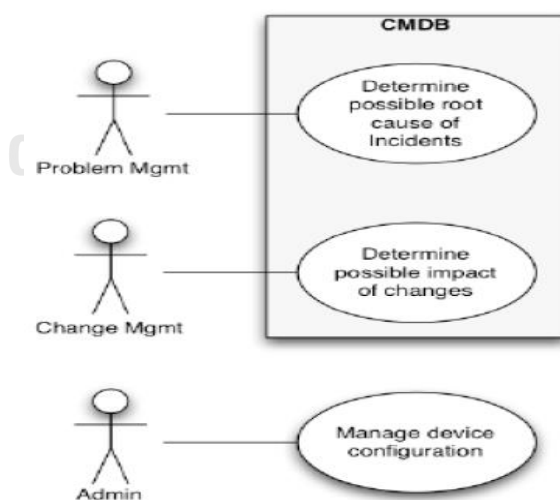
### 2). ซอฟต์แวร์ประยุกต์

ซอฟต์แวร์ประยุกต์ คือ โปรแกรมที่ตอบสนองกับฟังก์ชันทางธุรกิจ ทั้งนี้สามารถแยกได้เป็นโปรแกรมที่พัฒนาเองและโปรแกรมที่ซื้อสำเร็จรูป ยกตัวอย่างเช่น ซอฟต์แวร์ประมวลคำ (Word Processing Software) ซอฟต์แวร์ตารางทำงาน (Spread Sheet Software) ซอฟต์แวร์ระบบงานด้านบัญชี เป็นต้น

## 2.8 งานวิจัยที่เกี่ยวข้อง

2.8.1) งานวิจัย Designing CMDB data models with good utility and limited complexity โดย Brenner, M., Gillmeister, M. ปี ค.ศ. 2014 [17]

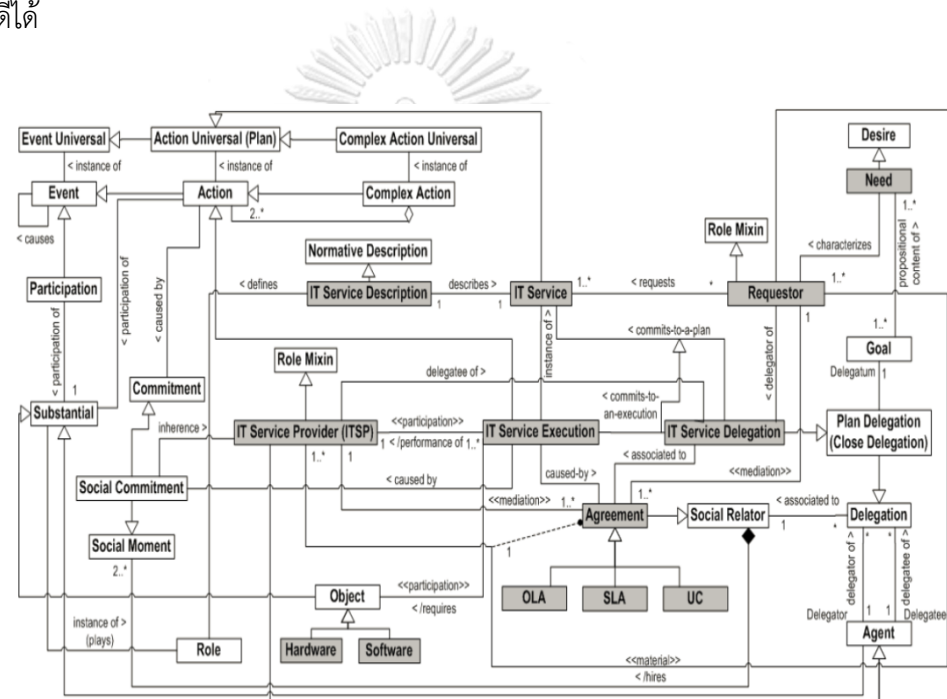
ไอทิล เวอร์ชัน 2 ได้นิยามฐานข้อมูลการจัดการโครงข่ายว่าออกแบบไว้เก็บข้อมูลคุณสมบัติของโครงข่ายและความสัมพันธ์ของโครงข่ายตลอดวงจรชีวิตของระบบ แต่ในไอทิล เวอร์ชัน 3 ได้นำเสนอระบบการจัดการโครงข่าย (Configuration Management System) ที่ครอบคลุมมากกว่าเดิมทำให้ฐานข้อมูลการจัดการโครงข่ายนิยามใหม่ว่าเป็นเครื่องมือสำหรับ รวบรวม จัดเก็บ จัดการ และทำการปรับปรุงวิเคราะห์และนำเสนอข้อมูล จึงจำเป็นต้องมีการออกแบบฐานข้อมูลการจัดการโครงข่ายใหม่เพื่อให้รองรับ ซึ่งโดยส่วนใหญ่แล้วระบบงานเกี่ยวกับฐานข้อมูลการจัดการโครงข่ายที่ล้มเหลว สาเหตุส่วนมากมาจากความซับซ้อน (Complexity) ของระบบ จึงต้องมีการออกแบบฐานข้อมูลการจัดการโครงข่ายเพื่อให้สามารถนำไปใช้ประโยชน์ได้หลากหลายและจำกัดความซับซ้อนโดยขั้นแรกคือ แยกแยะความต้องการและลำดับความสำคัญในการทำแผนภาพยูสเคสดังรูปที่ 3.1 หลังจากนั้น เป็นการออกแบบภายในโดยใช้ รูปแบบ (Pattern) คือ "Pattern Collective CI", "Pattern Rich CI Relation", "Pattern Multi-value Distribute" แนวทางในอนาคตจะมีการเพิ่มรูปแบบต่างๆ เข้ามาอีกได้ ผู้วิจัยได้สนใจงานวิจัยนี้โดยเฉพาะการนำไปใช้ประโยชน์ได้หลากหลายและจำกัดความและแยกแยะความต้องการและลำดับความสำคัญโดยการทำแผนภาพยูสเคสดังตัวอย่างรูปที่ 2.17



รูปที่ 2.17 แผนภาพยูสเคสเพื่อการออกแบบฐานข้อมูลการจัดการโครงข่าย [17]

2.8.2) งานวิจัย IT service management and governance modeling an ITSM Configuration process- A foundational ontology approach โดย Baioco, G., Costa, A.C.M., Calvi, C.Z., Garcia, A.S. ปี ค.ศ. 2009 [18]

งานวิจัยนี้ได้นำเสนอโมเดลสำหรับการนำออนโทโลยีมาใช้กับการบริหารจัดการบริการเทคโนโลยีสารสนเทศ โดยต้องการออกแบบฐานข้อมูลการจัดการโครงแบบ ให้สอดคล้องกับการทำงาน โดยวิเคราะห์ กระบวนการทางธุรกิจ (Business Process) และยังเสนอแนวคิดแม่แบบ (Concept Model) ที่ใช้ออนโทโลยีเข้ามาช่วยออกแบบฐานข้อมูล เพื่อช่วยในการจัดการและเห็นภาพรวม ความสัมพันธ์ต่างๆ ของ ธุรกิจ บุคคล กระบวนการ อุปกรณ์ และ เทคโนโลยีต่างๆ เพื่อการจัดการที่ดีได้



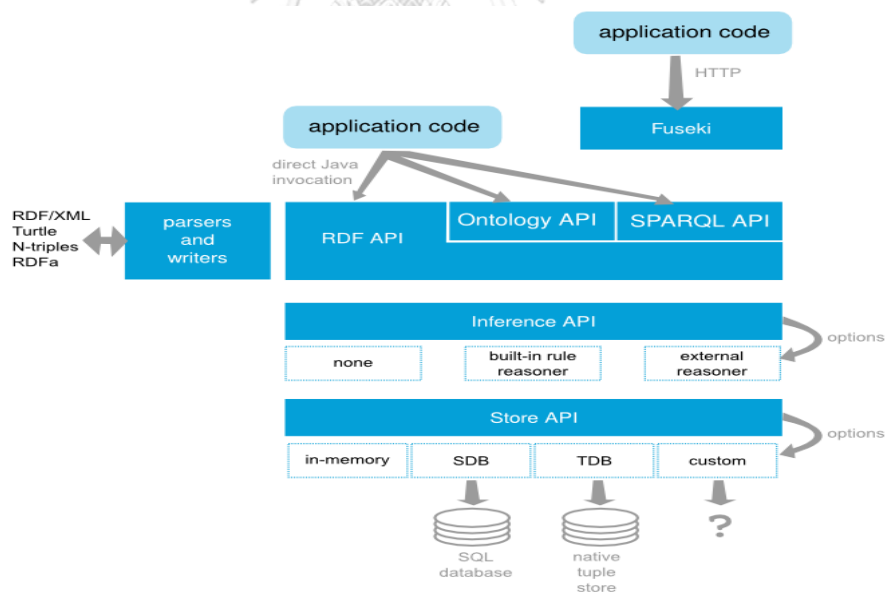
รูปที่ 2.18 โมเดลของการบริหารจัดการบริการเทคโนโลยีสารสนเทศ [18]

รูปที่ 2.18 แสดงผลลัพธ์การใช้ออนโทโลยีมาออกแบบโมเดลของการบริหารจัดการบริการเทคโนโลยีสารสนเทศ ซึ่งจะครอบคลุมกระบวนการและชิ้นงานต่างๆ ในกระบวนการ การบริการสารสนเทศ มีการโยงความสัมพันธ์ไปยังเอกสารข้อตกลง (Agreement) รวมทั้ง ฮาร์ดแวร์ ซอฟต์แวร์ ในส่วนของกระบวนการมีการกำหนดโรลของผู้ใช้งาน ซึ่งทำให้สามารถอธิบายโครงร่างระบบรวมทั้งกิจกรรมในการบริหารจัดการบริการเทคโนโลยีสารสนเทศได้ละเอียด

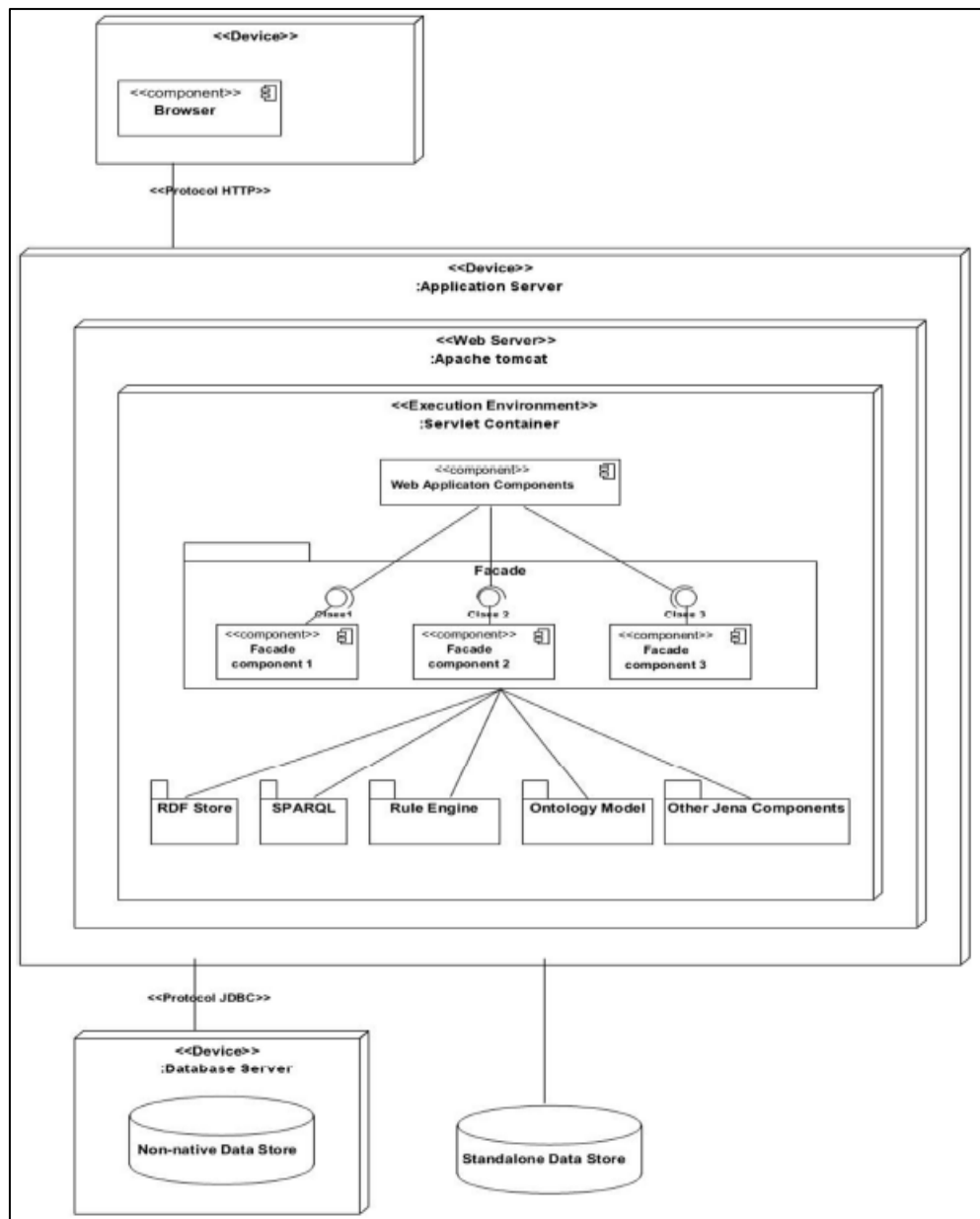
2.8.3) งานวิจัย ชั้นฟาสตสำหรับอาปาเซจีน่า โดย เรวัตร์ เลอเลิศวิชัย ปี ค.ศ. 2012 [19]

อาปาเซจีน่า (Apache Jena) เป็น กรอบงานของจาวาที่ใช้สร้างเว็บเชิงความหมาย อาปาเซจีน่า จะช่วยสรรหาสิ่งแวดล้อมให้โปรแกรมสำหรับ ภาษาอาร์ดีเอฟ ภาษาอาร์ดีเอฟเอส (RDFS) ภาษาอวาล์ ภาษาสปกวอล ภาษาจีอาร์ดีดีแอล (GRDDL) รวมถึงรูลเบสเอ็นจิน อาปาเซจีน่า ประกอบโดยโครงสร้างสถาปัตยกรรม ดังรูปที่ 2.19 ซึ่งจะประกอบไปด้วยชั้นของส่วนต่อประสานโปรแกรมประยุกต์ส่วนการเก็บข้อมูล (Store API) ส่วนต่อประสานโปรแกรมประยุกต์สำหรับการอนุมาน (Inference API) และ ส่วนของ แอปพลิเคชันโค้ด (Application Code)

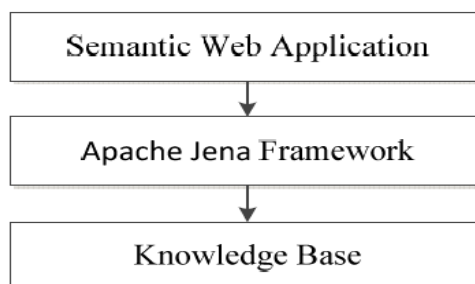
ส่วนการพัฒนาได้เสนอแนวคิดการพัฒนาเว็บเชิงความหมาย โดยใช้อาปาเซจีน่าเก็บความรู้ไว้และใช้แอปพลิเคชันเว็บเชิงความหมายดึงข้อมูล ซึ่งระหว่างอาปาเซจีน่าและเว็บเชิงความหมายนั้น เรวัตร์ได้เสนอแนวคิดที่พัฒนาชั้นฟาสตเป็นตัวกลางในการดึงข้อมูลออกจากอาปาเซจีน่าซึ่งทำให้ได้ประโยชน์ช่วยในการเชื่อมต่อและ ลดระดับความเกี่ยวข้อง หรือ ฟิงพากันของโมดูล (Coupling) ต่างๆ ในระบบ รูปที่ 2.20 จะแสดงแผนภาพดีพลอยเมนต์ ออกแบบมุมมองของระบบ และใช้เว็บเชิงความหมายติดต่อกับอาปาเซจีน่าซึ่งเก็บคลังความรู้ไว้ และรูปที่ 2.21 แสดงแนวคิดและวิธีวิจัย ซึ่งผู้วิจัยได้สนใจงานวิจัยนี้ในส่วนของการเก็บข้อมูลออนไลน์ไว้บนกรอบงานของอาปาเซจีน่า



รูปที่ 2.19 โครงสร้างสถาปัตยกรรมของอาปาเซจีน่า [19]



รูปที่ 2.20 แผนภาพดีพลอยเมนต์ชั้นฟาสาดสำหรับอปาเซจินา [19]



รูปที่ 2.21 แนวคิดการพัฒนาเว็บเชิงความหมายด้วยกรอบงานอปาเซจินา [19]

2.8.4) งานวิจัย Integrated Change and Configuration Management โดย Ward, C., Aggarwal, V., Bucu, M., Olsson, E., Weinberger S. ปี ค.ศ. 2007 [2]

งานวิจัยนี้ได้วิจัยเพื่อสรุปแนวทางที่ดีของ กระบวนการจัดการการเปลี่ยนแปลงและ การจัดการโครงสร้างซึ่งเป็นแกนสำคัญของไอทีล และ ได้เสนอแนวคิดว่าทำอย่างไรที่เสริมการให้บริการต่างๆ ในโดเมน ด้วย ฐานข้อมูลการจัดการโครงสร้างของไอบีเอ็มคือ "IBM Tivoli Change and Configuration Management Database" (CCMDB) ที่เป็นหัวใจสำคัญของยุทธการในบริการเทคโนโลยีสารสนเทศ ดังแผนภาพกระบวนการในการสนับสนุนการบริการแสดงดังรูปที่ 2.2 ได้มีการจัดแบ่งกระบวนการในการสนับสนุนการบริการและชิ้นงาน(Artifact) และมองว่า การจัดการการเปลี่ยนแปลงและการจัดการโครงสร้าง เป็นสิ่งที่ต้องทำควบคู่กัน

ผู้วิจัยได้แนวคิดในการมองฐานข้อมูลการจัดการโครงสร้างและการจัดการการเปลี่ยนแปลง เป็นหัวใจของการออกแบบระบบ รวมถึงภาพรวมกระบวนการที่สำคัญในการออกแบบซีไอ

2.8.5) งานวิจัย Automatic Generation of OWL Ontologies from UML Class Diagrams Based on Meta-Modelling and Graph Grammars โดย Aissam Belghiat และ Mustapha Bourahla [20]

งานวิจัยนี้มองว่าการสร้างข้อมูลออนโทโลยีตั้งแต่เริ่มต้นเลยเป็นงานที่ยาก และพบว่ายูเอ็มแอลกับภาษาอวาล์นั้นมีส่วนที่คล้ายกันเช่นคลาสและความสัมพันธ์ ในงานวิจัยนี้จึงใช้ประโยชน์โดยการแปลงยูเอ็มแอลเป็นภาษาอวาล์โดยการทำโมเดลเบื้องต้นและใช้ไวยากรณ์ของกราฟ นำมาสร้างเป็นสถาปัตยกรรมโมเดลที่สร้างข้อมูลภาษาอวาล์ได้อัตโนมัติ โดยมีการสร้างกฎ ดังนี้

- 1) แปลงยูเอ็มแอลคลาสเป็นคลาสในภาษาอวาล์
- 2) คลาสที่มีการสืบทอดจะสร้างเป็นซับคลาส
- 3) แปลงคุณสมบัติของคลาสเป็นคุณสมบัติเชิงข้อมูลในภาษาอวาล์
- 4) แปลงความสัมพันธ์ของคลาสเป็นคุณสมบัติเชิงวัตถุในภาษาอวาล์
- 5) การแปลงหน้าที่ของคลาสจะขึ้นอยู่กับคุณสมบัติคลาส

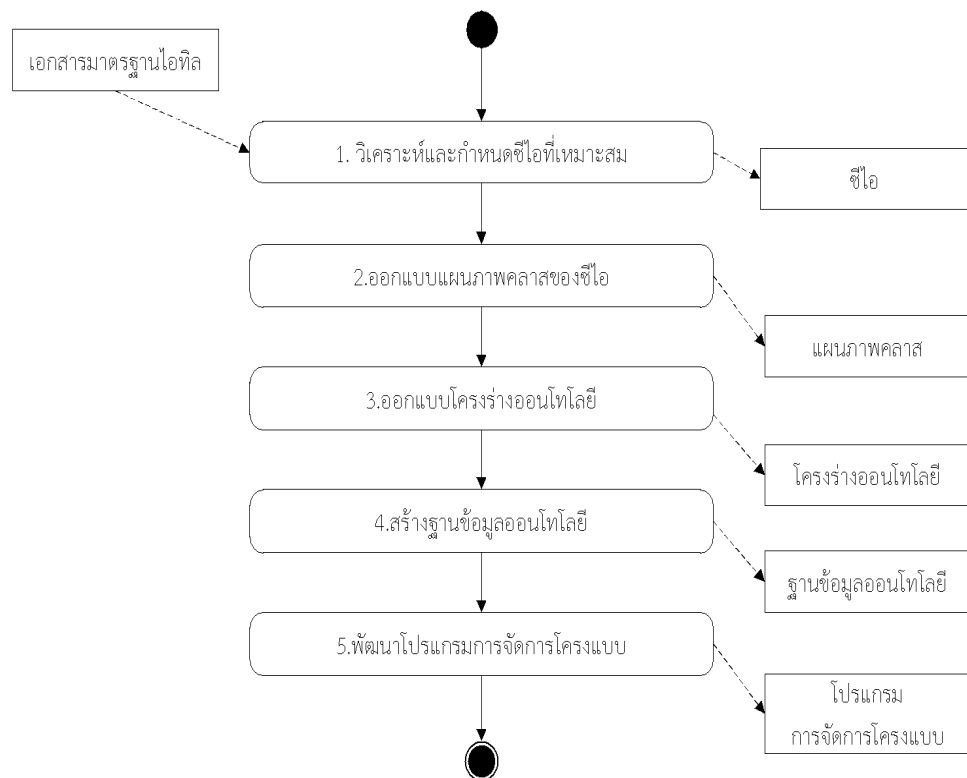
ผู้วิจัยได้แนวคิดในการแปลงแผนภาพคลาสมาเป็นข้อมูลออนโทโลยีจากงานวิจัยนี้

### บทที่ 3

#### การวิเคราะห์และออกแบบโครงสร้างออนไลน์ของซีไอ

ในบทนี้จะครอบคลุมเนื้อหาเกี่ยวกับการวิเคราะห์และออกแบบโครงสร้างออนไลน์ของซีไอ โดยจะนำเกณฑ์และแนวทางจากมาตรฐานไอทิลมาช่วยทำการวิเคราะห์และกำหนดซีไอ และเมื่อระบุซีไอได้แล้ว นำซีไอที่ได้ไปทำการการออกแบบแผนภาพคลาสของซีไอ โดยทำการออกแบบคลาส คุณสมบัติของคลาส และ ความสัมพันธ์ระหว่างคลาส จากนั้นใช้แผนภาพคลาสของซีไอที่ได้มาเป็นต้นแบบในการออกแบบโครงสร้างออนไลน์โดยใช้โปรแกรมโปรเทจ ซึ่งจะได้ผลลัพธ์คือโครงสร้างออนไลน์ของซีไอในรูปแบบของภาษาอาวล์

รูปที่ 3.1 แสดงภาพรวมการพัฒนากระบวนการจัดการโครงแบบด้วยออนไลน์ ซึ่งหลังจากทำการวิเคราะห์และออกแบบโครงสร้างออนไลน์ของซีไอแล้ว จะทำการสร้างฐานข้อมูลออนไลน์ของซีไอ และทำการพัฒนาโปรแกรมการจัดการโครงแบบต่อไป



รูปที่ 3.1 ภาพรวมการพัฒนากระบวนการจัดการโครงแบบด้วยออนไลน์



### 3.1 การวิเคราะห์และกำหนดซีไอ

การวิเคราะห์และกำหนดซีไอจะใช้แนวทางจากเอกสารไอทิลเวอร์ชัน 3 [4] และเอกสารไอทิลส่วนสนับสนุนการให้บริการ [5] เนื่องจากไอทิลเป็นมาตรฐานที่ได้รับการยอมรับและมีการอ้างอิงกันอย่างแพร่หลายในการบริการเทคโนโลยีสารสนเทศ

ไอทิลได้แนะนำวิธีการกำหนดซีไอไว้โดยให้สร้างโครงสร้างแบบแยกย่อยจำแนกย่อยจากรายการระดับบนสุดลงมาโดยขึ้นอยู่กับความต้องการทางธุรกิจและการบริการ ผู้วิจัยสร้างโครงสร้างแบบแยกย่อยของโครงสร้างพื้นฐานเทคโนโลยีสารสนเทศโดยกำหนดโครงสร้างแบบแยกย่อยเป็น 2 ชั้น ภายในโครงสร้างแบบแยกย่อยชั้นแรกกำหนดว่าเป็นชั้นของประเภทซีไอ ส่วนในชั้นที่สองจะแยกย่อยลงไปในแต่ละประเภทซีไอเพื่อกำหนดซีไอที่ควรมีอย่างน้อยในระบบจัดการโครงแบบ

โครงสร้างแบบแยกย่อยแสดงในรูปที่ 3.2 โดยชั้นแรกจะครอบคลุมถึง อุปกรณ์ฮาร์ดแวร์ ซอฟต์แวร์ เครือข่าย และเอกสาร ตามการแนะนำของไอทิล ดังสรุปในตารางที่ 3.1 โครงสร้างแบบแยกย่อยของโครงสร้างพื้นฐานเทคโนโลยีสารสนเทศ ชั้นแรกจะประกอบด้วยประเภทซีไอชื่อ Hardware, Software, Network และ Document

จากโครงสร้างแบบแยกย่อยในชั้นแรกที่เป็นประเภทซีไอจะทำการแยกย่อยลงมาในชั้นที่สองเพื่อหาซีไอที่ควรมีอย่างน้อยในระบบเทคโนโลยีสารสนเทศ โดยทำการแจกแจงซีไอสำหรับแต่ละประเภทซีไอ ดังนี้

1) สำหรับประเภทซีไอ Hardware เมื่อวิเคราะห์ถึงซีไอพื้นฐานที่ควรจัดอยู่ในกลุ่มนี้จะอิงตามอุปกรณ์และการเชื่อมต่อเครือข่ายเบื้องต้นในรูปที่ 2.13 คือ เซิร์ฟเวอร์ ไฟร์วอลล์ สวิตช์ และเราเตอร์ ดังนั้นซีไอที่ควรมีอย่างน้อยคือ ซีไอชื่อ Server, Firewall, Switch และ Router

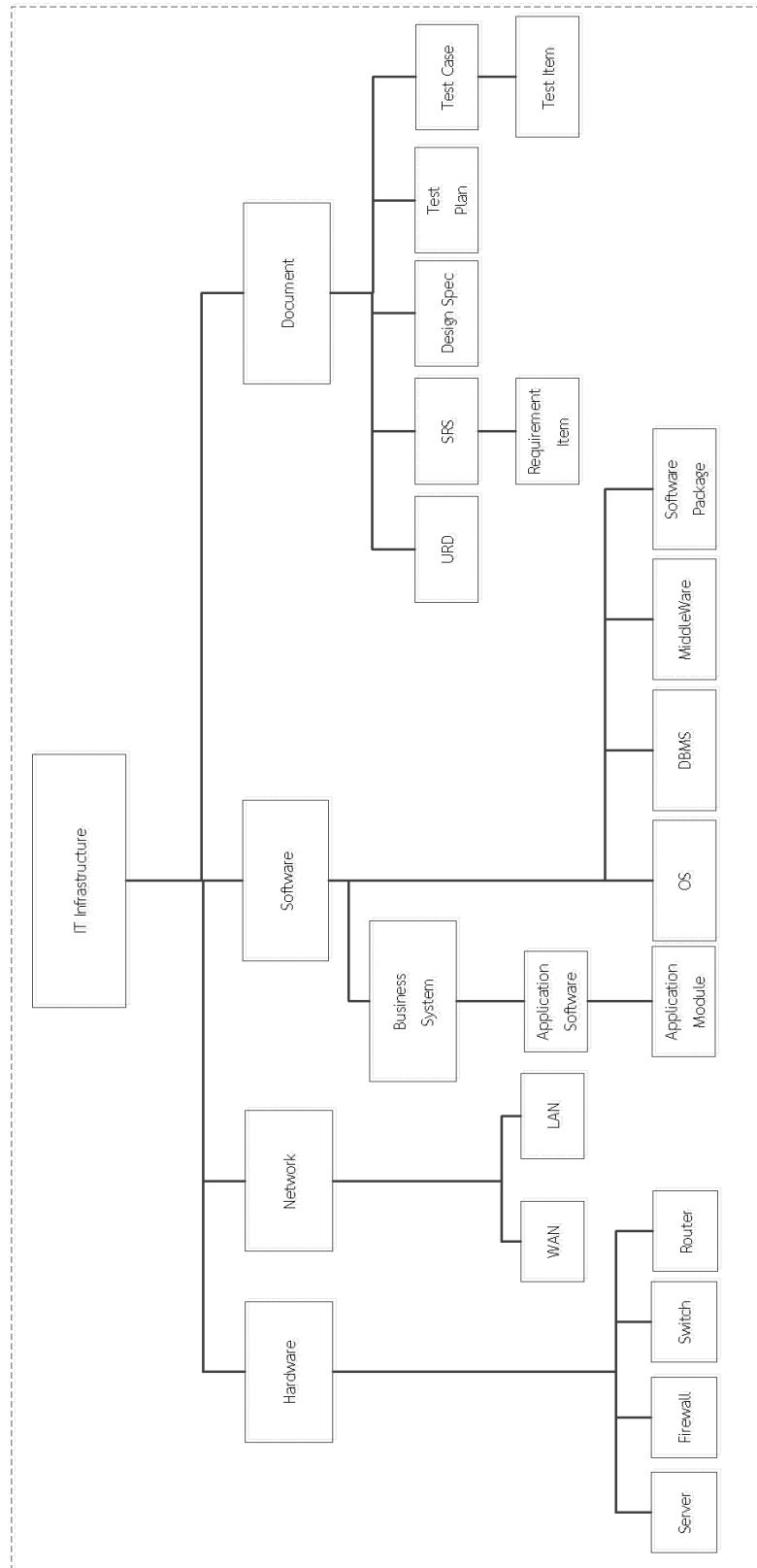
2) สำหรับประเภทซีไอ Software อ้างอิงจากรูปที่ 2.4 ไอทิลมีกลุ่มของ Business System คือระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจจึงระบุเป็นซีไอชื่อ Business System และเมื่อวิเคราะห์ซอฟต์แวร์ตามรูปที่ 2.16 ซอฟต์แวร์จะแบ่งเป็นซอฟต์แวร์ประยุกต์ และซอฟต์แวร์ระบบ ซึ่งซอฟต์แวร์ประยุกต์ คือซอฟต์แวร์ที่ตอบสนองกับฟังก์ชันทางธุรกิจ จึงกำหนดให้เป็นส่วนแยกย่อยภายใต้ซีไอ Business System และตั้งชื่อซีไอว่า Application Software นอกจากนี้พบว่าลักษณะของซีไอชื่อ Application Software ควรอยู่ในรูปแบบที่เป็นแอกกรีเกตของหลายๆ ซีไอรวมกัน เนื่องจากด้วยเพราะซอฟต์แวร์ที่ให้บริการทางธุรกิจอาจจะประกอบด้วยซอฟต์แวร์ประยุกต์หลายๆ ตัวทำงานรวมกันเป็นระบบซอฟต์แวร์ และในซีไอชื่อ Application Software จะประกอบไปด้วยแอกกรีเกตของหลายๆ ซอฟต์แวร์โมดูลรวมกันด้วยจึงกำหนดซอฟต์แวร์โมดูลให้เป็นซีไอชื่อ Application Module จากนั้นเมื่อดูจากความจำเป็นต่อการใช้งานของระบบ ซอฟต์แวร์ที่มีความสำคัญได้แก่ โปรแกรมระบบปฏิบัติการ และ มิดเดิลแวร์ เพราะเป็นซอฟต์แวร์ที่ช่วยในการทำงานของซอฟต์แวร์

อื่นๆ นอกจากนั้นมีความสำคัญของการเก็บข้อมูลจึงควรกำหนดให้มีซีไอของซอฟต์แวร์ระบบการจัดการฐานข้อมูล (DBMS: Database Management System) ด้วย ดังนั้นควรมีซีไอชื่อ OS คือระบบปฏิบัติการ ซีไอชื่อ Middleware คือมิดเดิลแวร์ และซีไอชื่อ DBMS คือระบบการจัดการฐานข้อมูล นอกจากซอฟต์แวร์ข้างต้นที่ได้กล่าวมาแล้วยังมีซอฟต์แวร์สำเร็จรูปอื่นๆที่ซื้อมาใช้ งาน จึงกำหนดชื่อซีไอชื่อ Software Package เป็นซอฟต์แวร์สำเร็จรูปทั่วไป

3) สำหรับประเภทซีไอ Network เมื่อวิเคราะห์เครือข่ายโดยทั่วไปสามารถจำแนกเป็นเครือข่ายบริเวณกว้าง และ เครือข่ายภายใน โดยที่ เครือข่ายภายใน คือเครือข่ายที่ใช้งานภายในองค์กร และ เครือข่ายบริเวณกว้าง คือ เครือข่ายที่ระบบภายในองค์กรติดต่อไปยังเครือข่ายข้างนอก ดังนั้นซีไอที่ควรมีอย่างน้อยคือ ซีไอชื่อ WAN คือ เครือข่ายบริเวณกว้าง และ ซีไอชื่อ LAN คือ เครือข่ายภายใน

4) สำหรับประเภทซีไอ Document เมื่อวิเคราะห์การให้บริการเทคโนโลยีสารสนเทศเพื่อสนองความต้องการทางธุรกิจ เอกสารจะมีความจำเป็นสำหรับการพัฒนาและบำรุงรักษาผลิตภัณฑ์ ในวงจรการพัฒนาระบบซอฟต์แวร์จะมีส่วนขั้นตอนของการเก็บความต้องการ ซึ่งเริ่มจากการเก็บเอกสารความต้องการจากผู้ใช้งาน (User Requirement Document: URD) จากนั้นนำมาพัฒนาเป็น เอกสารข้อกำหนดความต้องการ (Software Requirement Specification: SRS) ซึ่งขั้นตอนถัดไปก่อนจะนำไปพัฒนาจะมีการทำ เอกสารรายละเอียดการออกแบบระบบ (Software Design Specification: SDS) ถัดไปในขั้นตอนของการทดสอบ จะต้องมีการทำเอกสารแผนการทดสอบ เพื่อวางแผนและระบุว่าการทดสอบจะเกิดขึ้นในลักษณะไหนอย่างไร จากนั้นก็มีจัดทำเอกสารกรณีทดสอบซึ่งจะทำเพื่อระบุกรณีที่ใช้ทดสอบว่ามีอะไรบ้างที่จะทดสอบ และจากการอ้างอิง บทที่ 2.6 เรื่องเอกสารการทดสอบ [12] โครงสร้างของเอกสารกรณีทดสอบ จะประกอบด้วย รายละเอียดของรายการทดสอบ (Test Items) จึงกำหนดให้ มีซีไอของรายการทดสอบเป็นส่วนหนึ่งของซีไอเอกสารกรณีทดสอบ โดยซีไอรายการทดสอบนี้ จะระบุถึงชิ้นงานที่จะถูกทดสอบ รวมทั้งซอฟต์แวร์และฮาร์ดแวร์ที่เกี่ยวข้องได้ ดังนั้นซีไอที่ควรมีอย่างน้อยคือ ซีไอของ เอกสารความต้องการจากผู้ใช้งาน เอกสารข้อกำหนดความต้องการ เอกสารรายละเอียดการออกแบบระบบ โดยให้เป็นซีไอชื่อ URD, SRS และ Design Spec ส่วนของการทดสอบจะมีซีไอของ เอกสารการทดสอบ เอกสารกรณีทดสอบ รายการที่ใช้ในการทดสอบ โดยกำหนดให้เป็นซีไอชื่อ Test Plan, TestCase และ Test Item

จากซีไอที่ได้กำหนดมาข้างต้นแล้วนั้น เพื่อบันทึกประวัติการเปลี่ยนแปลงจึงเพิ่มซีไอของการเปลี่ยนแปลง คือ ซีไอ ChangeRequest เข้ามาด้วย ดังนั้นจะสามารถสรุปข้อมูลซีไอได้ตามตารางที่ 3.2 และเขียนโครงสร้างแบบแยกย่อยของระบบงานเทคโนโลยีสารสนเทศได้ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างแบบแยกย่อยของโครงสร้างพื้นฐานระบบงานเทคโนโลยีสารสนเทศ

ตารางที่ 3.1 ประเภทซีไอที่กำหนด

| ประเภทซีไอ | แหล่งที่มา | ความหมาย   |
|------------|------------|--|
| Hardware   | ไอทีล      | ฮาร์ดแวร์หรืออุปกรณ์ในระบบเทคโนโลยีสารสนเทศ  |
| Software   | ไอทีล      | โปรแกรมที่ทำหน้าที่ให้บริการกับแอปพลิเคชันอื่นๆ หรือเพื่อตอบสนองในการใช้งานทางธุรกิจ |
| Network    | ไอทีล      | เครือข่ายคอมพิวเตอร์ที่ใช้ในระบบเทคโนโลยีสารสนเทศ                                    |
| Document   | ไอทีล      | เอกสารในกระบวนการให้บริการระบบเทคโนโลยีสารสนเทศ                                      |

ตารางที่ 3.2 ซีไอที่กำหนด

| ซีไอ                 | ความหมาย  | ประเภทซีไอ |
|----------------------|---|------------|
| Server               | เซิร์ฟเวอร์หรือเครื่องคอมพิวเตอร์ที่ให้บริการหรือเครื่องคอมพิวเตอร์แม่ข่ายสำหรับสนับสนุนการให้บริการเทคโนโลยีสารสนเทศ | Hardware   |
| Firewall             | ไฟร์วอลล์หรืออุปกรณ์ที่ทำหน้าที่ตรวจสอบและกำหนดเงื่อนไขในการไหลผ่านของข้อมูลบนเครือข่ายคอมพิวเตอร์                    | Hardware   |
| Router               | อุปกรณ์ที่ทำหน้าที่เชื่อมต่อระบบเครือข่ายคอมพิวเตอร์แบบเราเตอร์   | Hardware   |
| Switch               | อุปกรณ์ที่ทำหน้าที่เชื่อมต่อระบบเครือข่ายคอมพิวเตอร์แบบสวิตช์   | Hardware   |
| Wan                  | ระบบเครือข่ายแบบกว้างใช้เพื่อเชื่อมต่อเครือข่ายกับเครือข่าย   | Network    |
| Lan                  | ระบบเครือข่ายภายในใช้ในการเชื่อมต่อเครื่องคอมพิวเตอร์ต่างๆ  | Network    |
| BusinessSystem       | ระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจ  | Software   |
| Application Software | โปรแกรมหรือซอฟต์แวร์ที่ตอบสนองกับฟังก์ชันทางธุรกิจ  | Software   |
| Application Module   | โมดูลหรือส่วนประกอบในซอฟต์แวร์  | Software   |
| OS                   | ซอฟต์แวร์ระบบปฏิบัติการ   | Software   |
| DBMS                 | ซอฟต์แวร์ระบบการจัดการฐานข้อมูล   | Software   |
| Middleware           | มิดเดิลแวร์ หรือซอฟต์แวร์ตัวกลาง  | Software   |
| Software Package     | ซอฟต์แวร์สำเร็จรูป  | Software   |
| Test Plan            | เอกสารแผนการทดสอบคือมุมมองแบบกว้างที่ระบุว่าการทดสอบจะเกิดขึ้นในลักษณะไหน อะไรจะถูกนำมาทดสอบบ้าง                      | Document   |
| TestCase             | เอกสารทดสอบที่มีการระบุกรณีที่จะใช้ในการทดสอบ   | Document   |
| Test Item            | รายการที่ใช้ในการทดสอบ  | Document   |
| SRS                  | เอกสารที่ระบุความต้องการในการพัฒนาซอฟต์แวร์   | Document   |
| URD                  | เอกสารความต้องการจากผู้ใช้งาน   | Document   |
| Design Spec          | เอกสารรายละเอียดการออกแบบระบบ   | Document   |
| ChangeRequest        | คำร้องขอเปลี่ยนแปลง   |            |

### 3.2 การออกแบบแผนภาพคลาส

หลังจากมีการกำหนดประเภทซีไอและซีไอแล้ว ขั้นตอนต่อไปคือการออกแบบคุณสมบัติและกำหนดความสัมพันธ์ของประเภทซีไอและซีไอ ผู้วิจัยพบว่าเราสามารถเปรียบเทียบซีไอเสมือนคลาสในแผนภาพคลาส และเปรียบเทียบความสัมพันธ์ระหว่างประเภทซีไอและซีไอเสมือนความสัมพันธ์ระหว่างคลาสเช่นเดียวกัน ดังนั้นเพื่อให้อ่านเข้าใจง่ายและสามารถแปลงให้เป็นโครงร่างออนโทโลยีได้ง่ายขึ้น จึงทำการออกแบบแผนภาพคลาสของประเภทซีไอและซีไอ โดยขั้นตอนหลักในการออกแบบแผนภาพคลาสมือ 3 ขั้นตอนคือ

- 1) ทำการสร้างแผนภาพคลาสที่ระบุการสืบทอดจากคลาสแม่สู่คลาสลูก
- 2) ทำการระบุความสัมพันธ์ระหว่างคลาส
- 3) ทำการระบุคุณสมบัติของแต่ละคลาส

#### 3.2.1 การสร้างแผนภาพคลาสที่ระบุการสืบทอดจากคลาสแม่สู่คลาสลูก

ในหัวข้อนี้จะทำการแปลงข้อมูลประเภทซีไอและซีไอจากรูปที่ 3.2 โดยกำหนดให้มีโครงสร้างแบบคลาสแม่และคลาสลูก ซึ่งจะสามารถเขียนแผนภาพคลาสเบื้องต้นของระบบได้ตามรูปที่ 3.3 ประกอบไปด้วย คลาส ChangeRequest และคลาสหลักคือ IT Infrastructure

ลำดับถัดมา คลาส IT Infrastructure จะมีคลาสลูกสืบทอดเป็น คลาสชื่อ Hardware, Software, Network และ Document

กลุ่มของคลาสที่มีการสืบทอดในขั้นสุดท้าย คือ คลาสชื่อ Server, Firewall, Router, Switch, Wan, Lan, BusinessSystem, Application Software, Application Module, OS, DBMS, Middleware, Software Package, Test Plan, TestCase, Test Item , SRS, URD และ Design Spec

#### 3.2.2 การระบุความสัมพันธ์ระหว่างคลาส

ในหัวข้อนี้จะทำการวิเคราะห์ความสัมพันธ์ของแต่ละคลาสโดยพิจารณาจากบริบทของแต่ละซีไอที่ควรมีและพิจารณาจากความต้องการของระบบจัดการโครงข่ายของระบบเทคโนโลยีสารสนเทศซึ่งมีความต้องการเบื้องต้นของระบบ เช่น สามารถมองภาพรวมของแต่ละแอปพลิเคชันใช้ทรัพยากรอะไรบ้าง หรือ สามารถช่วยในการวิเคราะห์ในการบริหารจัดการค่าใช้จ่ายของโครงการ เป็นต้น เริ่มวิเคราะห์จากซีไอบนสุดที่เป็นคลาสแม่คือ คลาส Infrastructure ซึ่งซีไอจะต้องรองรับคำขอการเปลี่ยนแปลงได้จึงกำหนดให้มีความสัมพันธ์ชื่อ hasChange ซึ่งจะโยงไปคำขอการเปลี่ยนแปลงหรือ

คลาส ChangeRequest ต่อมาประเภทซีไอฮาร์ดแวร์ อุปกรณ์ฮาร์ดแวร์จะมีการเชื่อมต่อกันจึงกำหนดให้มีความสัมพันธ์ชื่อ connects อุปกรณ์ฮาร์ดแวร์มีความใช้งานที่เข้ากันได้ต่อกันกำหนดให้มีความสัมพันธ์ชื่อ isCompatibleWith และอุปกรณ์ฮาร์ดแวร์จะอยู่ภายใต้เครือข่ายภายในจึงกำหนดให้มีความสัมพันธ์ชื่อ resides ทั้งนี้เครือข่ายภายในก็มีการเชื่อมต่อไปยังเครือข่ายขนาดกว้างจึงใช้ความสัมพันธ์ชื่อ connects ด้วย ในส่วนของซอฟต์แวร์ จะมีการติดตั้งซอฟต์แวร์ลงบนเซิร์ฟเวอร์จึงกำหนดให้มีความสัมพันธ์ชื่อ install ซอฟต์แวร์ที่ต้องการใช้ฮาร์ดแวร์เฉพาะกำหนดให้มีความสัมพันธ์ชื่อ required ซอฟต์แวร์ที่ต้องการใช้ซอฟต์แวร์อื่นร่วมกันกำหนดให้มีความสัมพันธ์ชื่อ requiredSoftwares นอกจากนั้นซอฟต์แวร์จะมีความสัมพันธ์ที่อ้างอิงการมีเอกสารจึงกำหนดให้มีความสัมพันธ์ชื่อ HasDocument และส่วนของมิตเดิลแวร์ที่เป็นตัวกลางนั้น ซอฟต์แวร์บางซอฟต์แวร์จะมีปฏิสัมพันธ์กับมิตเดิลแวร์จึงกำหนดให้มีความสัมพันธ์ชื่อ interact ต่อมาในส่วนของเอกสารแต่ละเอกสารจะมีการอ้างอิงถึงกันเช่น เอกสารแผนการทดสอบจะอ้างอิงถึงเอกสารข้อกำหนดความต้องการและเอกสารความต้องการจากผู้ใช้งาน เอกสารข้อกำหนดความต้องการก็มีอ้างอิงไปยังเอกสารความต้องการจากผู้ใช้งาน เอกสารรายละเอียดการออกแบบระบบก็อ้างอิงไปยังเอกสารข้อกำหนดความต้องการ เอกสารกรณีทดสอบมีการอ้างอิงไปยังเอกสารแผนการทดสอบและไปยังรายการความต้องการ ซึ่งจะกำหนดการอ้างอิงเอกสารเป็นชื่อความสัมพันธ์ ReferTo จะสรุปอธิบายความสัมพันธ์ตามที่กำหนดมาได้ตามตารางที่ 3.3 และแสดงแผนภาพคลาสในรูปที่ 3.3

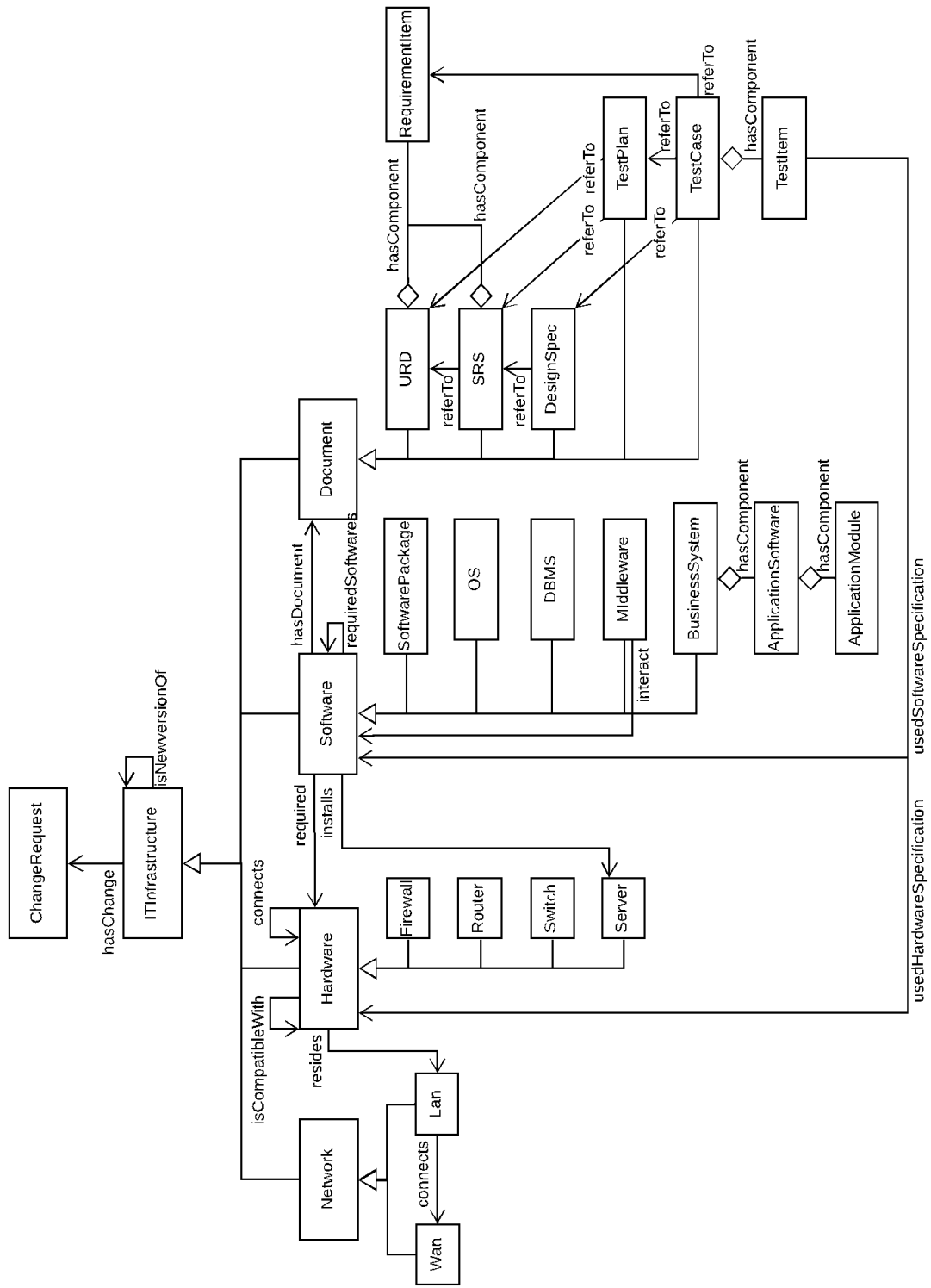
ตารางที่ 3.3 ความสัมพันธ์ระหว่างคลาส

| คลาส     | ชื่อความสัมพันธ์  | คลาสที่สัมพันธ์ด้วย | รูปแบบ | คำอธิบาย  |
|----------|-------------------|---------------------|--------|---|
| Hardware | connects          | Hardware            | 1..*   | อุปกรณ์ฮาร์ดแวร์จะมีการเชื่อมต่อกัน                     |
| Hardware | isCompatibleWith  | Hardware            | 1      | อุปกรณ์ฮาร์ดแวร์มีความใช้งานที่เข้ากันได้ต่อกัน         |
| Hardware | resides           | Lan                 | 1..*   | แสดงความสัมพันธ์ของฮาร์ดแวร์ที่อยู่ภายใต้เครือข่ายภายใน |
| Lan      | connects          | WAN                 | 1      | แสดงเครือข่ายภายในที่มีการเชื่อมต่อกับเครือข่ายภายนอก   |
| Software | installs          | Server              | 0..*   | ซอฟต์แวร์ติดตั้งลงบนเซิร์ฟเวอร์                         |
| Software | required          | Hardware            | 0..*   | ซอฟต์แวร์ต้องการใช้ฮาร์ดแวร์เฉพาะ                       |
| Software | requiredSoftwares | Software            | 0..*   | ซอฟต์แวร์ที่ต้องการใช้                                  |

|                      |              |                       |      |  |
|----------------------|--------------|-----------------------|------|--|
|                      |              |                       |      | ซอฟต์แวร์ร่วมกัน   |
| Software             | hasDocument  | Document              | 0..* | แสดงความสัมพันธ์ของซอฟต์แวร์ซึ่งมีเอกสารต่างๆที่เกี่ยวข้อง   |
| Software             | interact     | Middleware            | 0..* | ซอฟต์แวร์มีปฏิสัมพันธ์กับมิดเดิลแวร์   |
| BusinessSystem       | hasComponent | ApplicationSoftware[] | 0..* | ระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจสามารถประกอบไปด้วยโปรแกรมต่างๆได้มากกว่าหนึ่ง  |
| Application Software | hasComponent | ApplicationModule[]   | 0..* | โปรแกรมต่างๆสามารถประกอบไปด้วย โปรแกรมโมดูลหรือส่วนย่อยของโปรแกรมด้วย  |
| URD                  | hasComponent | RequirementItem[]     | 0..* | เอกสารความต้องการของผู้ใช้งาน ประกอบด้วยรายการความต้องการหลายๆรายการประกอบกัน  |
| SRS                  | hasComponent | RequirementItem[]     | 0..* | เอกสารข้อกำหนดความต้องการประกอบด้วยรายการความต้องการหลายๆรายการประกอบกัน   |
| SRS                  | referTo      | URD                   | 1    | เอกสารข้อกำหนดความต้องการจะต้องมีการอ้างอิงถึงเอกสารความต้องการของผู้ใช้งาน เพื่อให้ระบุได้ว่า ข้อมูลเรื่องความต้องการในเชิงของผู้ใช้มีความสัมพันธ์อย่างไร |
| Design Specification | referTo      | SRS                   | 1    | เอกสารการออกแบบระบบต้องสามารถอ้างอิงถึงเอกสารข้อกำหนดความต้องการได้  |
| TestPlan             | referTo      | URD                   | 1    | เอกสารแผนการทดสอบสามารถอ้างอิงถึง เอกสารความต้องการของผู้ใช้งานที่ใช้เป็นข้อมูลตั้งต้นในการออกแบบ  |

|                  |                             |                 |      |  |
|------------------|-----------------------------|-----------------|------|--|
|                  |                             |                 |      | เรื่องการทดสอบได้  |
| TestPlan         | referTo                     | SRS             | 1    | เอกสารแผนการทดสอบสามารถอ้างอิงเอกสารข้อกำหนดความต้องการที่ใช้เป็นข้อมูลตั้งต้นในการออกแบบเรื่องการทดสอบได้   |
| TestCase         | has                         | TestPlan        | 1    | เอกสารกรณีทดสอบจะอ้างอิงกับแผนการทดสอบเสมอ   |
| TestCase         | hasItem                     | TestItem[]      | 0..* | เอกสารกรณีทดสอบจะประกอบไปด้วยรายการทดสอบต่างๆ  |
| TestCase         | referTo                     | RequirementItem | 1..* | เอกสารกรณีทดสอบสามารถอ้างอิงรายการความต้องการได้   |
| TestItem         | usesHardware Specifications | Hardware[]      | 1..* | รายการทดสอบสามารถระบุได้ว่าใช้ฮาร์ดแวร์ที่มีข้อมูลจำเพาะใดๆได้ ยกตัวอย่างเช่น การทดสอบ Software A จำเป็นต้องใช้ฮาร์ดแวร์ที่มีระบบประมวลผลความเร็วไม่ต่ำกว่าเท่าใด หน่วยความจำไม่ต่ำกว่าเท่าใด และพื้นที่ในการจัดเก็บข้อมูลเท่าใด เป็นต้น |
| TestItem         | usesSoftware Specification  | Software[]      | 0..* | รายการทดสอบสามารถระบุได้ว่าใช้ซอฟต์แวร์ที่มีข้อมูลใดๆได้บ้างยกตัวอย่างเช่น ทดสอบบนระบบปฏิบัติการที่เป็น Windows 2008 เป็นต้น   |
| ITInfrastructure | hasChange                   | ChangeRequest   | 1    | ซีไอจะมีการอ้างอิง ChangeRequest เมื่อต้องการจะแก้ไข   |
| ITInfrastructure | IsNewVersionOf              | ChangeRequest   | 1    | ความสัมพันธ์นี้จะแสดงหลังจากการแก้ไขซีไอเพื่อแทนว่าเป็นเวอร์ชันใหม่  |





รูปที่ 3.3 แผนภาพคลาสเบื้องต้นของระบบ

### 3.2.3 การกำหนดคุณสมบัติของคลาส

การกำหนดคุณสมบัติของคลาสจะเลือกใช้และอ้างอิงคุณสมบัติทั่วไปของซีไอตามที่ไอทิลแนะนำ [4] และตามกระบวนการของวิศวกรรมซอฟต์แวร์ [6] ซึ่งมีคุณสมบัติเบื้องต้น คือ ข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน ชื่อ ประเภทของซีไอรายละเอียด เวอร์ชัน และ สถานะ เป็นคุณสมบัติให้กับคลาส Infrastructure ซึ่งเป็นคลาสแม่บนสุด ดังตารางที่ 3.4 จากนั้นจะวิเคราะห์คุณสมบัติเพิ่มเติมนอกจากคุณสมบัติที่สืบทอดมาแล้วให้กับคลาสอื่นๆ ในหัวข้อถัดไป

ตารางที่ 3.4 คุณสมบัติของคลาส ITInfrastructure

| คลาส             | คุณสมบัติ   | คำอธิบาย   |
|------------------|-------------|--|
| ITInfrastructure | ID          | ข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน   |
| ITInfrastructure | Name        | ชื่อ ยกตัวอย่างเช่น Windows2008 ซึ่งเป็นชื่อของซีไอ Software   |
| ITInfrastructure | CIType      | ประเภทของซีไอ คือการจัดหมวดหมู่ประเภทเพื่อให้ง่ายในการใช้งาน เช่น ประเภทซีไอ ฮาร์ดแวร์ ซอฟต์แวร์ เป็นต้น                           |
| ITInfrastructure | Description | รายละเอียด ที่จะอธิบายเพิ่มเติม  |
| ITInfrastructure | Version     | เวอร์ชัน ใช้เพื่อแยกแยะซีไอได้เมื่อมีการแก้ไข  |
| ITInfrastructure | Status      | สถานะของซีไอ โดยสถานะจะมีรูปแบบแตกต่างกันตามประเภทของซีไออื่นๆ ยกตัวอย่างซีไอ Hardware สถานะอาจจะเป็น Active หรือ Inactive เป็นต้น |

#### 3.2.3.1 คุณสมบัติของคลาส ChangeRequest

สำหรับคำร้องขอเปลี่ยนแปลงจะมีข้อมูลคุณสมบัติ คือ ข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน และควรเก็บข้อมูลความเร่งด่วนของคำขอเพื่อสะดวกในการจัดการ เก็บค่าผลกระทบของคำขอเพื่อไว้ประเมินผลกระทบต่อระบบโดยรวม นอกจากนั้นควรเก็บ ชื่อผู้ร้องขอ วันที่แจ้งคำร้อง วันเวลาที่เริ่มดำเนินการ วันเวลาที่ดำเนินการแล้วเสร็จ เพื่อทำบันทึกประวัติของคำร้องขอ ซึ่งจะได้สรุปคุณสมบัติเพิ่มเติมของคลาส ChangeRequest ดังตารางที่ 3.5

ตารางที่ 3.5 คุณสมบัติของคลาส ChangeRequest

| คลาส          | คุณสมบัติ        | คำอธิบาย   |
|---------------|------------------|--|
| ChangeRequest | changed          | ข้อมูลเฉพาะที่ไม่ซ้ำเพื่อระบุตัวตน   |
| ChangeRequest | serverityLevel   | ความเร่งด่วนของคำขอนั้นๆ สามารถระบุเป็นระดับ high, medium, low ได้   |
| ChangeRequest | impactLevel      | การส่งผลกระทบต่อคำขอนั้นๆ สามารถระบุเป็นระดับ high, medium, low ได้ การที่จะมีผลกระทบระดับสูงนั้นแปลว่า คำขอนี้ อาจจะมีผลกระทบอื่นๆ ในวงกว้างซึ่งต้องใช้เวลาในการตรวจสอบผลกระทบต่างๆ นานขึ้น |
| ChangeRequest | requestedBy      | การระบุผู้ที่ขอเปลี่ยนแปลง   |
| ChangeRequest | requestDate      | วันที่แจ้งคำร้อง   |
| ChangeRequest | processStartDate | วันเวลาที่เริ่มดำเนินการ   |
| ChangeRequest | processEndDate   | วันเวลาที่ดำเนินการแล้วเสร็จ   |

### 3.2.3.2 คุณสมบัติเพิ่มเติมของคลาส Hardware และซัพพลายของ Hardware

เนื่องด้วยการบริหารจัดการงานเทคโนโลยีสารสนเทศจำเป็นต้องทราบค่าบำรุงรักษาและงบในการลงทุนของอุปกรณ์ฮาร์ดแวร์ เพื่อคำนวณบริหารค่าใช้จ่าย จึงกำหนดให้มีคุณสมบัติ ค่าบำรุงรักษาครั้งแรก ค่าบำรุงรักษาต่อปี วันที่เริ่มให้บริการ และควรมีคุณสมบัติของ แบรนด ชื่อผลิตภัณฑ์ เพื่อสะดวกในการซ่อมบำรุงและจัดซื้อ นอกจากนั้นในปัจจุบันฮาร์ดแวร์ส่วนมากจะเชื่อมกับเครือข่ายจึงให้มีคุณสมบัติเลขไอพีและค่าตัวเลขเฉพาะในแต่ละอุปกรณ์ในเครือข่ายอีกด้วย เมื่อวิเคราะห์ซัพพลายของฮาร์ดแวร์ ไฟร์วอลล์ซึ่งทำหน้าที่บล็อกและคัดกรองข้อมูลควรมีคุณสมบัติเงื่อนไขต่างๆ ที่ไฟร์วอลล์อนุญาตให้ผ่าน และลิสต์ของเว็บไซต์ที่อนุญาตด้วย ในส่วนของเราเตอร์บางชนิดจะมีคุณลักษณะแบบไร้สายจึงให้มีคุณสมบัติที่เช็คว่ามีคุณลักษณะแบบไร้สายหรือไม่ สำหรับอุปกรณ์สวิตช์ซึ่งใช้เชื่อมต่อกับคอมพิวเตอร์จะกำหนดให้เก็บคุณสมบัติที่บอกจำนวนพอร์ตเชื่อมต่อทั้งหมดเพื่อเช็คว่ามีเพียงกับคอมพิวเตอร์ที่ต้องใช้งาน ในส่วนของเซิร์ฟเวอร์จะเก็บคุณสมบัติความเร็วซีพียู พื้นที่จัดเก็บ การรองรับระบบปฏิบัติการ และขนาดหน่วยความจำเพื่อสะดวกในการดูแลและอัปเกรดเครื่อง ดังนั้นจะสามารถสรุปคุณสมบัติเพิ่มเติมของคลาส Hardware และซัพพลายของ Hardware ได้ดังตารางที่ 3.6

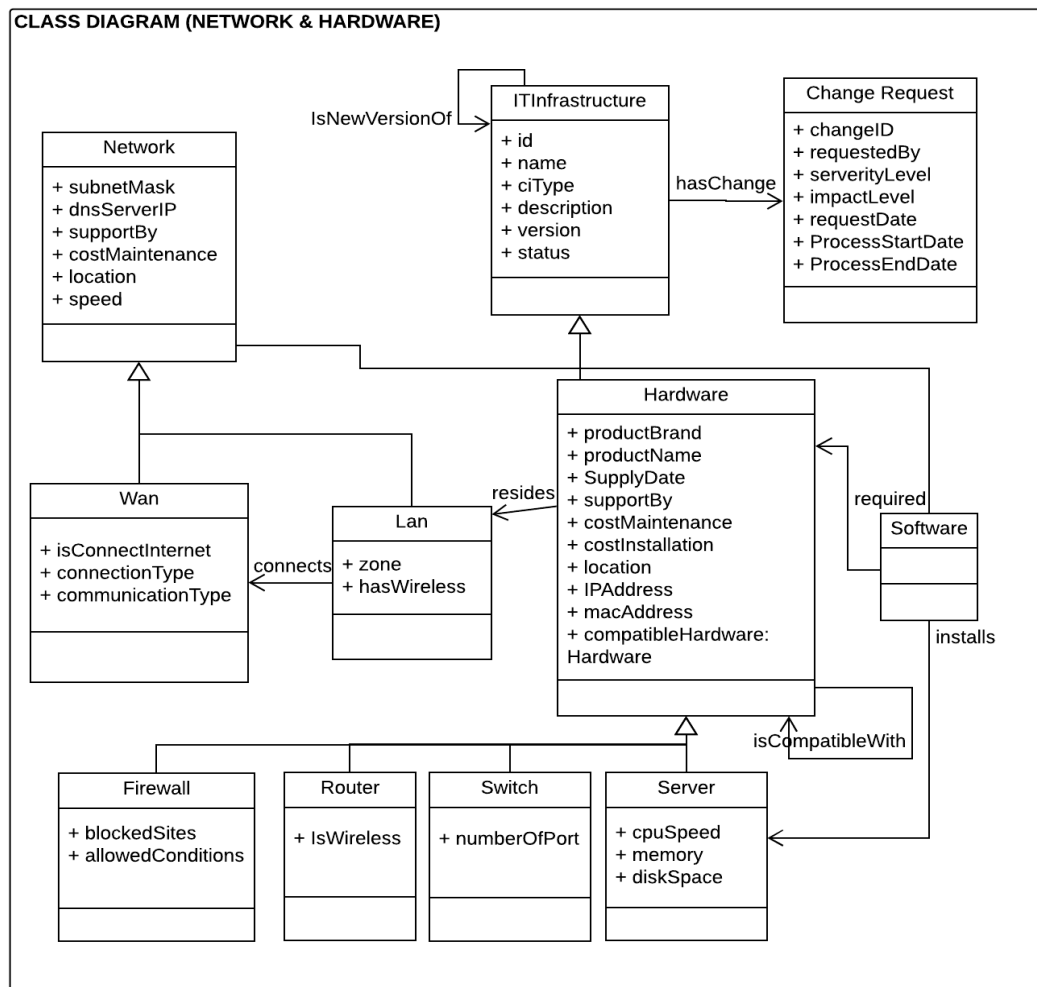
ตารางที่ 3.6 ผลสรุปคุณสมบัติเพิ่มเติมของคลาส Hardware และซัพพลายของ Hardware

| คลาส     | คุณสมบัติ         | คำอธิบาย  |
|----------|-------------------|---|
| Hardware | costMaintenance   | ค่าบำรุงรักษาต่อปี  |
| Hardware | costInstallation  | ค่าดำเนินการติดตั้งครั้งแรก                               |
| Hardware | Location          | สถานที่เซิร์ฟเวอร์นั้นติดตั้งอยู่                         |
| Hardware | SupportBy         | ชื่อผู้ดูแลระบบ   |
| Hardware | SupplyDate        | วันที่เริ่มให้บริการ                                      |
| Hardware | ProductBrand      | แบรนด์  |
| Hardware | ProductName       | ชื่อผลิตภัณฑ์   |
| Hardware | IPAddress         | เลขไอพี   |
| Hardware | macAddress        | ค่าเลขเฉพาะในแต่ละอุปกรณ์ในเครือข่าย                      |
| Firewall | allowedConditions | เงื่อนไขต่างๆ ที่ไฟร์วอลล์อนุญาตให้ผ่าน                   |
| Firewall | blockedSites      | ชื่อเว็บไซต์ต่างๆ หรือไอพีต่างๆ ที่ไม่อนุญาตให้เข้าถึงได้ |
| Router   | isWireless        | ข้อมูลจะเป็น true ถ้ามีคุณลักษณะแบบไร้สาย                 |
| Server   | cpuSpeed          | ความเร็วของซีพียู   |
| Server   | diskSpace         | ขนาดพื้นที่ในการจัดเก็บข้อมูล                             |
| Server   | Memory            | พื้นที่หน่วยความจำ ตัวอย่างเช่น 16GB                      |
| Server   | compatibleOS      | รองรับระบบปฏิบัติการอะไรบ้าง                              |
| Switch   | numberOfPort      | จำนวนพอร์ตเชื่อมต่อทั้งหมดของอุปกรณ์สวิตช์                |

### 3.2.3.3 คุณสมบัติเพิ่มเติมของคลาส Network และซัพพลายของ Network

ข้อมูลเครือข่ายคุณสมบัติที่ควรมีของคลาสเครือข่ายคือ ส่วนแบ่งย่อยของเครือข่าย ไอพีของเซิร์ฟเวอร์ที่ให้บริการดีเอ็นเอส ซึ่งเป็นค่าที่ต้องใช้ในการติดตั้งบนคอมพิวเตอร์ที่เชื่อมต่อ นอกจากนี้ควรมีคุณสมบัติ ค่าบำรุงรักษาต่อปี สถานที่ที่ระบบเครือข่ายนั้นติดตั้งอยู่ ชื่อผู้ดูแลระบบ ความเร็วในการเชื่อมต่อ เพื่อใช้ในการจัดการและบำรุงรักษาระบบ ในส่วนของซัพพลายของ Network เครือข่ายภายในจะเพิ่มคุณสมบัติของโซน เช่นโซนหลังไฟร์วอลล์ เพื่อแยกแยะระดับความปลอดภัยของเครือข่าย และมีคุณสมบัติที่เก็บว่ามีการใช้งานแบบไร้สายหรือไม่ สำหรับเครือข่ายภายนอกจะเพิ่มคุณสมบัติว่าเชื่อมต่ออินเทอร์เน็ตหรือไม่ เพื่อดูแลความปลอดภัยของข้อมูลในเครือข่าย ประเภทของ

เทคโนโลยีการสื่อสาร และประเภทของการเชื่อมต่อของเครือข่ายภายนอก จะสรุปคุณสมบัติเพิ่มเติมของคลาส Network ได้ดังตารางที่ 3.7 และ ในรูปที่ 3.4 แสดงแผนภาพคลาสแบบละเอียดของคลาส ITInfrastructure, ChangeRequest, Hardware และ Network



รูปที่ 3.4 แผนภาพคลาสแบบละเอียดของ คลาส ITInfrastructure, คลาส ChangeRequest, คลาส Hardware และ คลาส Network

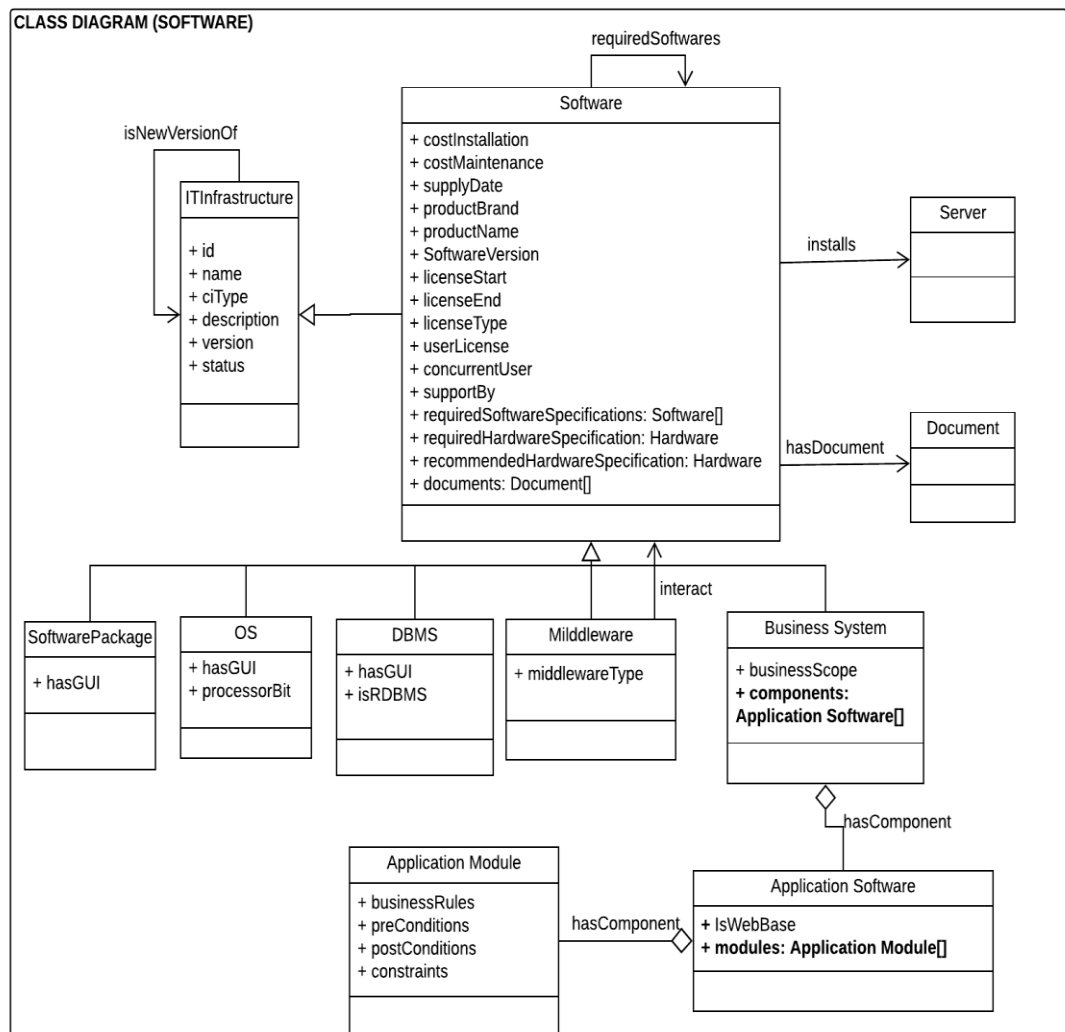
ตารางที่ 3.7 ผลสรุปคุณสมบัติเพิ่มเติมของคลาส Network และซัพคลาสของ Network

| คลาส    | คุณสมบัติ   | คำอธิบาย                                |
|---------|-------------|---|
| Network | subnetMask  | ส่วนแบ่งย่อยของเครือข่าย                |
| Network | dnsServerIP | ไอพีของเซิร์ฟเวอร์ที่ให้บริการดีเอ็นเอส |

|         |                   |  |
|---------|-------------------|--|
| Network | costMaintenance   | ค่าบำรุงรักษาต่อปี   |
| Network | location          | สถานที่ที่ระบบเครือข่ายนั้นติดตั้งอยู่                             |
| Network | supportBy         | ชื่อผู้ดูแลระบบ  |
| Network | Speed             | ความเร็วในการเชื่อมต่อ เช่น 10Mbps                                 |
| Lan     | zone              | โซนของที่เครือข่ายตั้งอยู่ เช่น อยู่ในโซนที่อยู่หลังไฟร์วอลล์      |
| Lan     | hasWireless       | เป็นค่าบอกว่าเครือข่ายมีการใช้งานแบบไร้สายหรือไม่                  |
| Wan     | isConnectInternet | ถ้ามีค่าเป็น true คือมีการเชื่อมต่อออกอินเทอร์เน็ต                 |
| Wan     | connectionType    | ประเภทของการเชื่อมต่อ เช่น PPPoE, PPPoA เป็นต้น                    |
| Wan     | communicationType | ประเภทของเทคโนโลยีการสื่อสาร เช่น เคเบิล เอดีเอสแอล (ADSL) เป็นต้น |

### 3.2.3.4 คุณสมบัติเพิ่มเติมของคลาส Software และซัพคลาสของ Software

เนื่องด้วยการบริหารจัดการงานเทคโนโลยีสารสนเทศจำเป็นต้องทราบค่าบำรุงรักษาและงบในการลงทุนของซอฟต์แวร์ เพื่อคำนวณบริหารค่าใช้จ่าย จึงกำหนดให้มีคุณสมบัติ ค่าบำรุงรักษาครั้งแรก และค่าบำรุงรักษาต่อปี และ วันที่เริ่มให้บริการ และควรมีคุณสมบัติของ แบนด์ ซึ่งผลิตภัณฑ์ และ เวอร์ชันของผลิตภัณฑ์ เพื่อสะดวกในการค้นหาข้อมูลและออกรายงาน เก็บค่าจำนวนผู้ใช้งานพร้อมกันเพื่อเป็นข้อมูลรองรับการใช้งาน นอกจากนี้ควรมี วันที่สิทธิ์ใช้งานเริ่มต้น วันที่สิทธิ์ใช้งานสิ้นสุด และจำนวนผู้ใช้งานต่อลิขสิทธิ์ เพื่อเป็นข้อมูลของลิขสิทธิ์ สำหรับซัพคลาสของ Software มิติเดิลแวร์สามารถมีได้หลายประเภทจึงกำหนดให้มีคุณสมบัติที่เก็บประเภทของมิติเดิลแวร์ ระบบจัดการฐานข้อมูลและระบบปฏิบัติการควรมีข้อมูลให้รู้วาระรองรับผู้ใช้งานได้เท่าไร ระบบปฏิบัติการมีโปรแกรมสำเร็จรูป และระบบจัดการฐานข้อมูลจะมีคุณสมบัติให้บ่งบอกได้ว่า มีกราฟฟิคหน้าจอการใช้งานหรือไม่ ระบบจัดการฐานข้อมูลเก็บคุณสมบัติว่าเป็นฐานข้อมูลแบบระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) หรือไม่ ระบบปฏิบัติการควรมีเก็บข้อมูลของบิตในการประมวลผลเช่น 32 บิต 64 บิต และกำหนดให้ระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจมีคุณสมบัติที่บอกถึงจุดมุ่งหมายของระบบและอธิบายภาพรวมระบบ สำหรับโปรแกรมประยุกต์จะให้เก็บคุณสมบัติว่าเป็นโปรแกรมที่ทำงานแบบเว็บเบสเพื่อช่วยในการจัดสรรทรัพยากรแก่คอมพิวเตอร์ของผู้ใช้งาน สำหรับโปรแกรมโมดูลซึ่งเป็นส่วนย่อยที่สุดของโปรแกรมเพื่อให้เห็นจุดประสงค์การใช้งานจึงกำหนดให้เก็บคุณสมบัติของกฎเกณฑ์ต่างๆในมุมมองเชิงธุรกิจ ข้อจำกัดโปรแกรม เงื่อนไขก่อนการใช้งาน และเงื่อนไขหลังใช้งาน จะสรุปคุณสมบัติเพิ่มเติมของคลาส Software และซัพคลาสของ Software ได้ดังตารางที่ 3.8 และแสดงแผนภาพคลาสในรูปที่ 3.5



รูปที่ 3.5 แผนภาพคลาสแบบละเอียดของคลาส Software และซัพคลาสของคลาส Software

ตารางที่ 3.8 คุณสมบัติเพิ่มเติมของคลาส Software และซัพคลาสของ Software

| คลาส     | คุณสมบัติ       | คำอธิบาย              |
|----------|-----------------|-----------------------|
| Software | CostInstall     | ค่าบำรุงรักษาครั้งแรก |
| Software | CostMaintenance | ค่าบำรุงรักษาต่อปี    |
| Software | SupplyDate      | วันที่เริ่มให้บริการ  |
| Software | ProductBrand    | แบรนด์                |

|                            |                 |  |
|----------------------------|-----------------|--|
| Software                   | ProductName     | ชื่อผลิตภัณฑ์  |
| Software                   | Softwareversion | รุ่นของซอฟต์แวร์นั้นๆ เช่น รุ่นปี 2018 เป็นต้น   |
| Software                   | licenseStart    | วันที่สิทธิ์การใช้งานเริ่มต้น  |
| Software                   | licenseEnd      | วันที่สิทธิ์การใช้งานสิ้นสุด   |
| Software                   | userLicense     | จำนวนผู้ใช้งานต่อลิขสิทธิ์   |
| Software                   | supportBy       | ชื่อผู้ดูแลระบบ  |
| DBMS                       | isRDBMS         | ข้อมูลนี้จะบ่งบอกว่าฐานข้อมูลเป็นแบบ RDBMS หรือไม่   |
| MiddleWare                 | middlewareType  | มิดเดิลแวร์สามารถมีได้หลายประเภทเช่น มิดเดิลแวร์ของฐานข้อมูล มิดเดิลแวร์ของเว็บไซต์ เป็นต้น  |
| OS                         | processorBit    | เก็บค่าบิตในการประมวลผล เช่น 32 บิต หรือ 64 บิต เป็นต้น  |
| DBMS, Software Package, Os | hasGUI          | ข้อมูลนี้จะบ่งบอกถึงว่าระบบปฏิบัติการนี้มีหน้าจอให้ผู้ใช้ใช้งานในกราฟฟิกหรือไม่ โดยปกติแล้วระบบปฏิบัติการจะมีอยู่สองหน้าต่างตาในการใช้งานคือแบบเทอร์มินอล (Terminal) คือมีหน้าจอสีดำและรอรับผลจากคีย์บอร์ดส่งเข้าเป็นคำสั่งหรือ เป็น หน้าจอสวยงามมีไอคอนรูปภาพและใช้เมาส์ ได้ หาก ข้อมูลนี้เป็นจริง (true )แปลว่ามีหน้าจอสวยงามให้ผู้ใช้ทั่วไปทำงานได้ |
| BusinessSystem             | bussinessScope  | จุดมุ่งหมายของระบบและอธิบายภาพรวมระบบ  |
| ApplicationSoftware        | IsWebBase       | ข้อมูลนี้จะบ่งบอกว่าซอฟต์แวร์เป็นแบบเว็บเบสหรือไม่   |
| ApplicationModule          | businessRules   | กฎเกณฑ์ต่างๆในมุมมองเชิงธุรกิจ   |
| ApplicationModule          | Constraints     | ข้อจำกัดต่างๆ ของส่วนย่อยของโปรแกรม  |
| ApplicationModule          | postConditions  | เงื่อนไขต่างๆ หลังจากการใช้งานในส่วนนี้ ตัวอย่างเช่นระบบล็อคอินหลังจากการทำล็อคอินแล้วต้องมีการส่งเมลล์ไปให้ผู้ใช้งานทราบอีกทีหนึ่ง  |



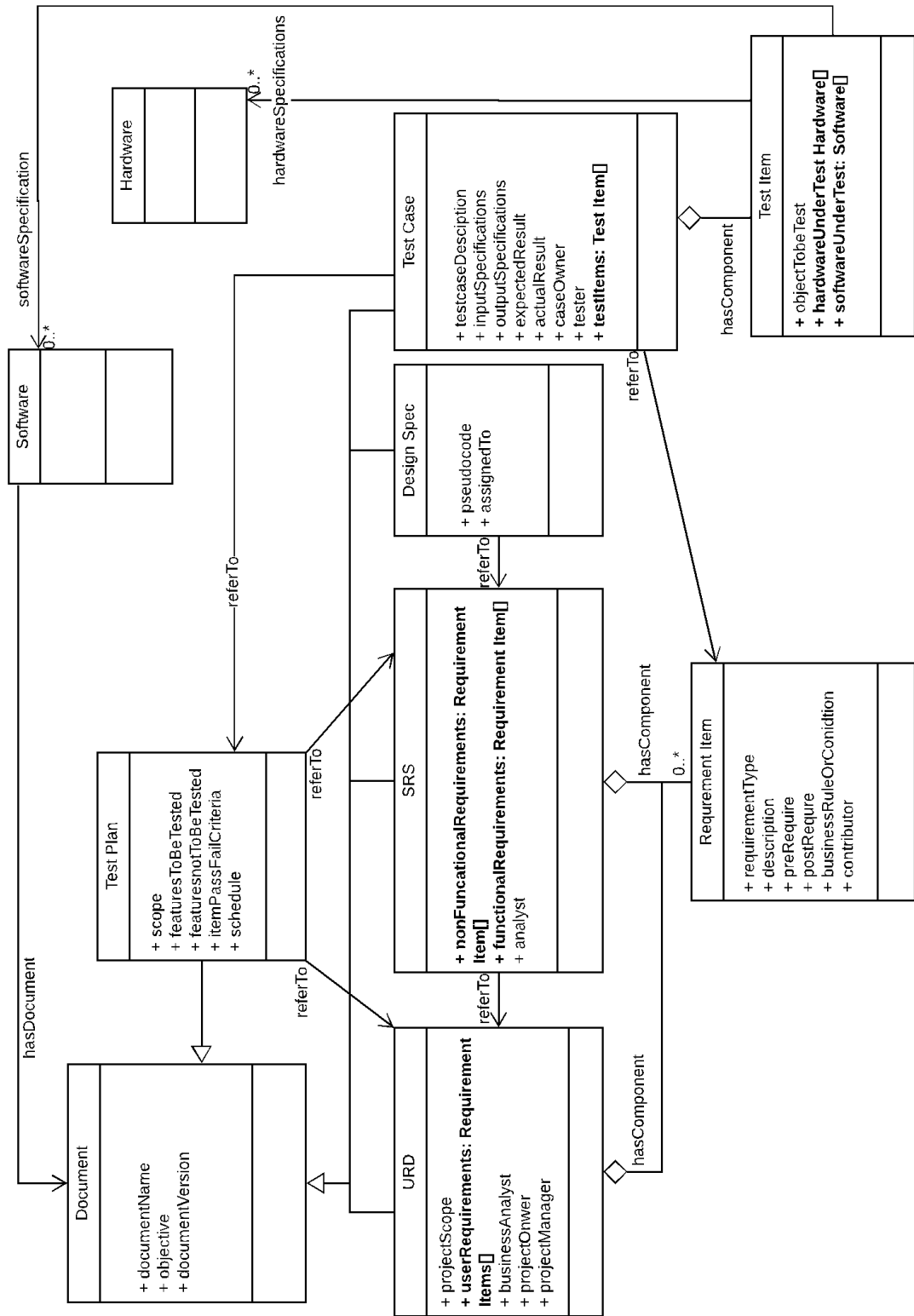
|                   |               |  |
|-------------------|---------------|--|
| ApplicationModule | preConditions | เงื่อนไขต่างๆก่อนที่จะใช้งานในส่วนนี้ ตัวอย่างเช่น ผู้ใช้ต้องมีสิทธิ์การเข้าถึง หรือ มีการสร้างข้อมูลก่อนแล้วถึงจะสามารถใช้งานในส่วนนี้ได้ |
|-------------------|---------------|--|

### 3.2.3.5 คุณสมบัติเพิ่มเติมของคลาส Document และซัพคลาสของคลาส Document

เพื่อใช้เป็นข้อมูลอธิบายเอกสารเพิ่มเติมจึงควรมีคุณสมบัติคือ ชื่อของเอกสาร เวอร์ชันของเอกสาร วัตถุประสงค์ของเอกสาร จากนั้นทำการวิเคราะห์ซัพคลาสของคลาส Document คือ เอกสารความต้องการจากผู้ใช้งานซึ่งควรจะมีรายละเอียดของนักวิเคราะห์และเก็บข้อมูล ผู้จัดการ วิศวกร เจ้าของโครงการ และ ขอบเขตของโครงการ ในส่วนของเอกสารข้อกำหนดความต้องการ จะเก็บชื่อนักวิเคราะห์ระบบ ความต้องการในเชิงหน้าที่การทำงานของซอฟต์แวร์ ความต้องการที่ไม่ใช่ในเชิงหน้าที่การทำงานของซอฟต์แวร์ เพื่อเก็บรายละเอียดแยกประเภทข้อมูลของความต้องการ ในส่วนของรายการความต้องการ จะเก็บกฎเกณฑ์หรือเงื่อนไขต่างๆ ของความต้องการ ผู้ให้ข้อมูลของความต้องการ คำอธิบายของความต้องการ สิ่งที่จะต้องทำก่อนหน้าเพื่อใช้ประกอบในความต้องการนั้น สิ่งที่จะต้องทำหลังการเสร็จสิ้นความต้องการนั้น และ ประเภทของความต้องการ ส่วนเอกสารข้อกำหนดการออกแบบนั้น กำหนดให้มีคุณสมบัติ ผู้พัฒนาระบบที่ถูกมอบหมายงานให้ และ วิธีการทำงานซึ่งเป็นการเขียนโปรแกรม

สำหรับเอกสารการทดสอบจะวิเคราะห์โดยอ้างอิงจากหัวข้อ 2.6 ดังนี้ เอกสารแผนการทดสอบ จะกำหนดให้เก็บคุณสมบัติ ขอบเขตของการทดสอบ คุณลักษณะที่จะนำมาทดสอบ คุณลักษณะที่จะไม่นำมาทดสอบ เงื่อนไขของการผ่านหรือไม่ผ่านการทดสอบ กำหนดเวลาในการทดสอบ ต่อมาเอกสารกรณีทดสอบ จะกำหนดให้เก็บคุณสมบัติ คือ รายละเอียดของการทดสอบ ข้อกำหนดข้อมูลนำเข้า ข้อกำหนดข้อมูลส่งออก ผลลัพธ์ที่คาดการณ์เอาไว้ ผลลัพธ์ที่ได้ตามจริง ผู้ออกแบบการทดสอบ ผู้ทำการทดสอบ และสุดท้ายรายการทดสอบจะ เก็บคุณสมบัติที่ระบุว่าส่วนของซอฟต์แวร์ที่ระบุไว้ว่าจะใช้ทดสอบ

จะสรุปคุณสมบัติเพิ่มเติมของคลาส Document และซัพคลาสของคลาส Document ได้ดังตารางที่ 3.9 และแสดงแผนภาพคลาสในรูปแบบที่ 3.6



รูปที่ 3.6 แผนภาพคลาสแบบละเอียดของคลาส Document และซัพคลาสของคลาส Document

ตารางที่ 3.9 คุณสมบัติเพิ่มเติมของคลาส Document และซัพคลาสของคลาส Document

| คลาส            | คุณสมบัติ                  | คำอธิบาย   |
|-----------------|----------------------------|--|
| Document        | documentName               | ชื่อเอกสาร   |
| Document        | documentVersion            | รุ่นของเอกสารนั้นๆซึ่งโดยเอกสารจะเริ่มต้นด้วย 1.0.0 เสมอ หากมีการเปลี่ยนแปลงเล็กน้อย (minor change) ให้เปลี่ยนเลขหลักท้ายสุด หากมีการแก้ไขโครงสร้างหรือแก้ไขหลักๆ (major change) ให้เปลี่ยนเลขหลักกลาง หากมีการอนุมัติ ยืนยันแล้วให้เปลี่ยนเลขหน้าสุด โดยหากมีการเปลี่ยนแปลงเลขตัวหน้า เลขตัวหลังทั้งหมดจะต้องมีการปรับให้โดยเริ่มต้นเป็น 0 ยกตัวอย่างเช่น 1.0.0 -> 1.0.1 -> 1.0.2 -> 1.1.0 -> 1.1.1 -> 1.2.0 -> 2.0.0 เป็นต้น |
| Document        | objective                  | วัตถุประสงค์ของเอกสารนั้นๆ   |
| URD             | businessAnalyst            | ระบุชื่อนักวิเคราะห์ในเชิงธุรกิจของโครงการนั้นๆ  |
| URD             | projectManager             | ระบุผู้จัดการโครงการ   |
| URD             | projectOwner               | ระบุเจ้าของโครงการ   |
| URD             | projectScope               | ขอบเขตของโครงการ   |
| URD             | userRequirements           | ความต้องการต่างๆ ในมุมมองของผู้ใช้ทั่วไปพื้นฐานสำหรับการพัฒนาและออกแบบ โครงการ   |
| SRS             | analyst                    | ชื่อนักวิเคราะห์ระบบ   |
| SRS             | functionalRequirements     | ความต้องการในเชิงหน้าที่การทำงานของซอฟต์แวร์ ตัวอย่างเช่น ซอฟต์แวร์สามารถคำนวณดอกเบี้ยได้ เป็นต้น  |
| SRS             | nonFuncationalRequirements | ความต้องการที่ไม่ใช่ในเชิงหน้าที่การทำงานของซอฟต์แวร์ ตัวอย่างเช่น ซอฟต์แวร์จะต้องมีสีเขียว  |
| RequirementItem | businessRuleOrCondition    | กฎเกณฑ์หรือเงื่อนไขต่างๆของความต้องการข้อนี้ๆ  |
| RequirementItem | contributor                | ผู้ให้ข้อมูลของความต้องการนี้  |
| RequirementItem | description                | คำอธิบายของความต้องการข้อนี้ๆ  |
| RequirementItem | postRequire                | สิ่งที่จำเป็นต้องทำก่อนหน้าเพื่อใช้ประกอบในความ ต้องการนั้นๆ เช่นต้องมีการตรวจสอบสิทธิ์ของผู้ใช้ ก่อนเข้าถึงการคำนวณต่างๆ  |

|                 |                       |  |
|-----------------|-----------------------|--|
| RequirementItem | preRequire            | สิ่งที่จำเป็นต้องทำหลังการเสร็จสิ้นความต้องการนั้นๆ<br>เช่น ต้องบันทึกข้อมูลการใช้งานลงในล็อก  |
| RequirementItem | requirementType       | ประเภทของความต้องการ จะแบ่งเป็นความต้องการเชิงฟังก์ชัน (Functional Requirement) ความต้องการไม่เป็นฟังก์ชัน (Non Functional Requirement ) |
| DesignSpec      | assignedTo            | ผู้พัฒนาระบบที่ถูกมอบหมายงานให้  |
| DesignSpec      | pseudocode            | วิธีการทำงานซึ่งเป็นการเขียนโปรแกรมแบบคร่าวๆ   |
| TestPlan        | scope                 | ขอบเขตของการทดสอบ  |
| TestPlan        | featuresToBeTested    | ฟังก์ชันหรืออะไรที่จะถูกทดสอบในแผน   |
| TestPlan        | featuresnotToBeTested | ฟังก์ชันหรืออะไรที่จะไม่ถูกทดสอบในแผน  |
| TestPlan        | itemPassFailCriteria  | เงื่อนไขในการผ่านและไม่ผ่านการทดสอบ  |
| TestPlan        | schedule              | กำหนดเวลาของแผนการทดสอบ  |
| TestCase        | testcaseDescription   | รายละเอียดของกรณีทดสอบโดยละเอียด   |
| TestCase        | inputSpecifications   | ข้อกำหนดของข้อมูลในส่วนที่นำเข้าสู่ระบบ  |
| TestCase        | outputSpecifications  | ข้อกำหนดของข้อมูลในส่วนที่แสดงผลหรือนำออกจากระบบ   |
| TestCase        | expectedResult        | ผลลัพธ์ที่คาดการณ์เอาไว้   |
| TestCase        | actualResult          | ผลลัพธ์ที่ได้ตามจริง   |
| TestCase        | caseOnwer             | เจ้าของการทดสอบกรณีนั้นๆ   |
| TestCase        | tester                | ผู้ทำการทดสอบสำหรับกรณีนั้นๆ   |
| TestItem        | objectTobeTest        | ส่วนของซอฟต์แวร์ที่ระบุไว้ว่าจะใช้ทดสอบ  |

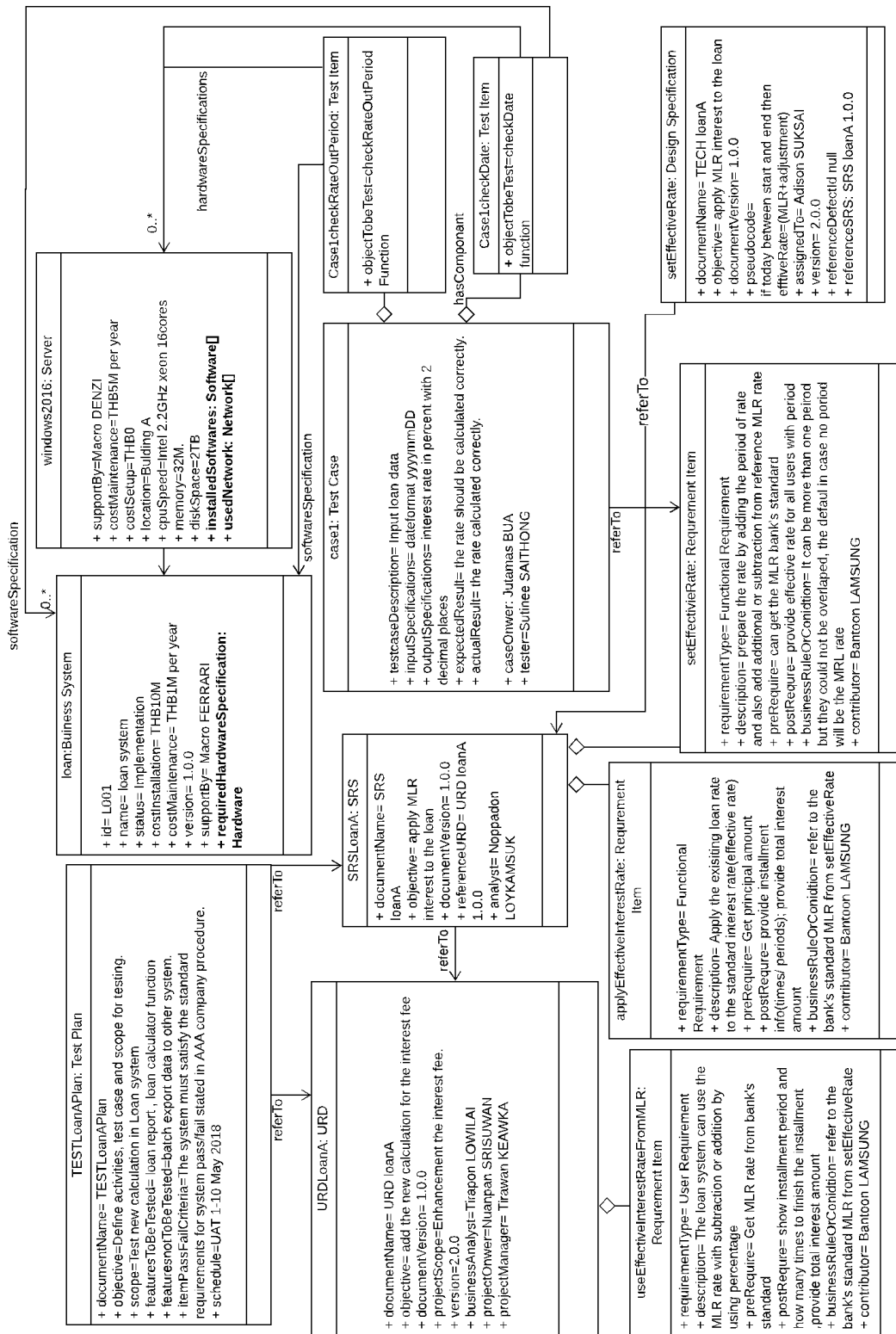
### 3.2.4 การสร้างแผนภาพวัตถุ (Object Diagram)

จากที่ได้ออกแบบแผนภาพคลาสมาในหัวข้อก่อนหน้านี้แล้วนั้น เพื่อช่วยในการวิเคราะห์ให้เห็นว่าคลาสที่ออกแบบสามารถใช้งานได้ถูกต้องจึงได้จัดทำแผนภาพวัตถุ ดังแสดงในภาพที่ 3.7 เป็นการจำลองสร้างซีไอของโครงการเรื่องกู้ยืม (Loan) ที่ได้มีการพัฒนาเสร็จสิ้นไปแล้วแต่มีการขอเพิ่มการทำงานบางส่วนเข้ามา ทั้งนี้แล้วแผนภาพแสดงให้เห็นถึง จุดเริ่มต้นนั่นคือ เอกสารความต้องการของ

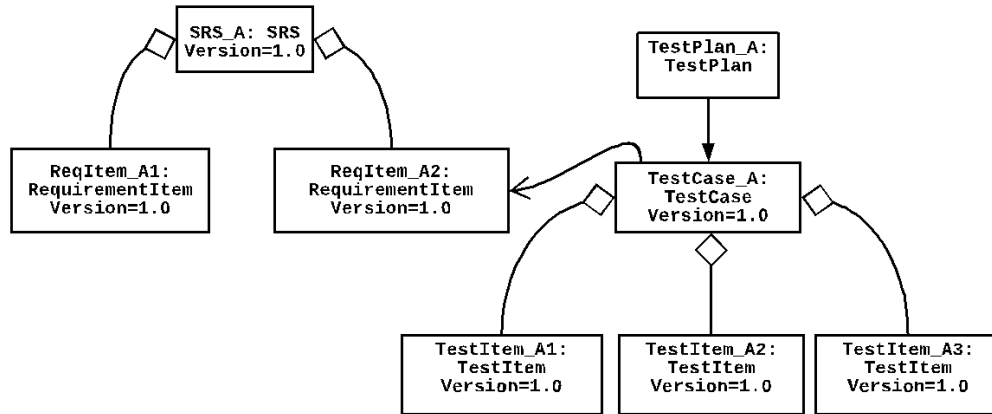
ผู้ใช้งานโดยไปยังจุดสุดท้ายนั้นคือการทดสอบที่เราพยายามแสดงให้เห็นข้อมูลในแต่ละส่วนตามลำดับ ดังนี้

- 1) นักวิเคราะห์ความต้องการทางธุรกิจ (Business Analyst) จัดทำเอกสารความต้องการของผู้ใช้งาน เพื่อแสดงถึงความต้องการใหม่ที่ต้องการเพิ่มการคิดดอกเบี้ยโดยให้ตั้งจากมาตรฐานกลางของธนาคาร ในตัวอย่างภาพจะเป็นเอกสารชื่อ URDLoanA
- 2) นักวิเคราะห์ความต้องการทางธุรกิจจัดทำเอกสารข้อกำหนดความต้องการ ชื่อ SRSLoanA โดยจำเป็นต้องอ้างอิงเอกสารความต้องการของผู้ใช้งาน
- 3) นักวิเคราะห์ระบบ (Systems Analyst) ทำการออกแบบ พร้อมทั้งทำเอกสารการออกแบบชื่อ setEffectiveRate เพื่อให้ โปรแกรมเมอร์สามารถทำการพัฒนาได้
- 4) เทสเตอร์จัดทำ แผนการทดสอบชื่อ TESTLoanA ซึ่งจำเป็นต้องอ้างอิงเอกสารข้อกำหนดความต้องการ SRSLoanA และเอกสารการออกแบบ setEffectiveRate
- 5) เทสเตอร์จัดทำ เอกสารกรณีทดสอบชื่อ case1 โดยอ้างอิงตาม รายการความต้องการแต่ละข้อที่ได้ จากเอกสารข้อกำหนดความต้องการ เป็นต้น

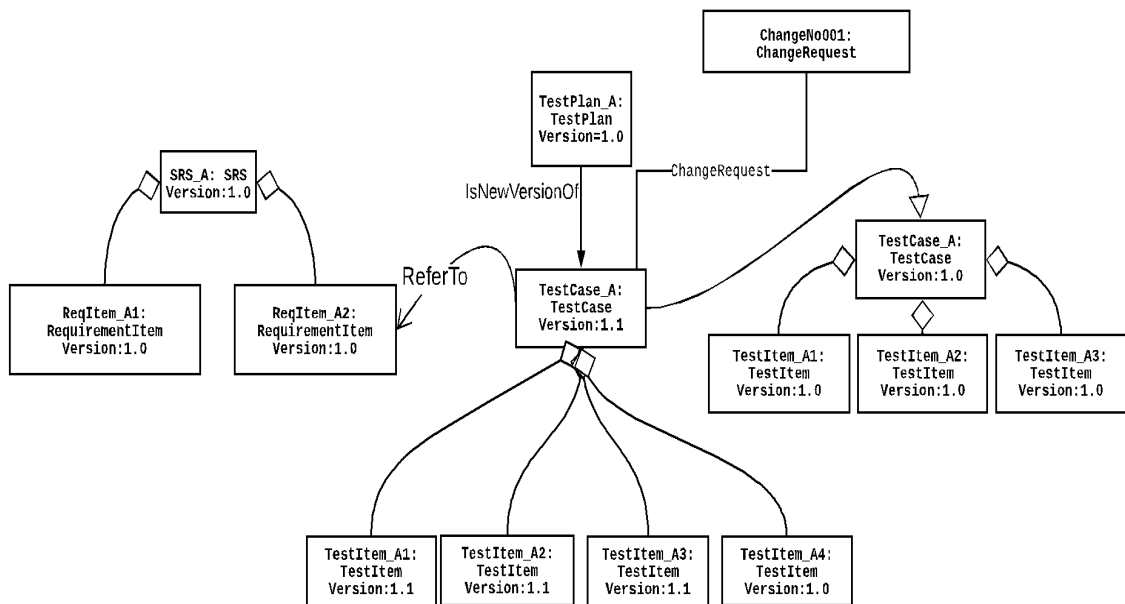
นอกจากนั้นยังมีกรณีที่มีการขอเปลี่ยนแปลงชื่อไอ ดังนั้นเพื่อแสดงให้เห็นว่ามีการจัดเก็บ ข้อมูลชื่อไออย่างไรจึงได้ทำการจำลองแผนภาพ ดังแสดงในภาพที่ 3.8 ซึ่งประกอบไปด้วยชื่อไอชื่อ TestPlan\_A คือเอกสารแผนการทดสอบ, TestCase\_A คือเอกสารข้อกำหนดการทดสอบ และ SRS\_A คือเอกสารข้อกำหนดความต้องการ และใน TestCase\_A จะมีชื่อไอแอกกรีเกต อย่าง TestItem\_A1, TestItem\_A2 และ TestItem\_A3 ด้วย ซึ่งทั้งหมดจะเป็นเวอร์ชัน 1.0 ซึ่งเมื่อมีการขอเปลี่ยนแปลงชื่อไอ TestCase\_A จะแสดงให้เห็นการเปลี่ยนแปลงในภาพที่ 3.9 โดยมีการสร้าง ชื่อไอ ChangeNo001 เพื่อทำการเปลี่ยนแปลง ชื่อไอ TestCase\_A ซึ่งได้เพิ่ม รายการทดสอบ TestItem\_A4 เข้ามา ดังนั้นชื่อไอ TestCase\_A จึงปรับเวอร์ชันเป็น 1.1 รวมทั้งชื่อไอแอกกรีเกต อย่าง TestItem\_A1, TestItem\_A2, TestItem\_A3 ก็ปรับเวอร์ชันเป็น 1:1 ด้วย TestCase\_A เวอร์ชัน 1.1 จะมีความสัมพันธ์คือเป็น IsNewVersion ของ TestCase\_A เวอร์ชัน 1.0 ส่วนความสัมพันธ์ จาก TestPlan\_A และ SRS\_A จะชี้มาที่ TestCase\_A เวอร์ชัน 1.1 แทน เป็นต้น สำหรับกรณีชื่อไอ ประเภทซอฟต์แวร์ ฮาร์ดแวร์ และ เครือข่าย ก็สามารถทำเวอร์ชันคอนโทรลได้ เช่น ชื่อไอเครือข่าย ภายใน มีการเปลี่ยนค่าดีเอ็นเอสเซิร์ฟเวอร์ หรือซบเน็ต ก็จะทำให้เวอร์ชันของชื่อไอเครือข่ายภายใน เปลี่ยนไป เป็นต้น



รูปที่ 3.7 แผนภาพวัตถุส่วนของเอกสาร



รูปที่ 3.8 แผนภาพวัตถุแสดงตัวอย่างเอกสารก่อนการเปลี่ยนเวอร์ชัน



รูปที่ 3.9 แผนภาพวัตถุแสดงตัวอย่างเอกสารหลังมีการขอเปลี่ยนแปลง

เพื่อแสดงให้เห็นว่ามีการจัดเก็บข้อมูลซีไอซอฟต์แวร์ที่มีการเปลี่ยนแปลงได้ทำการจำลองแผนภาพ ดังแสดงในรูปที่ 3.10 ตามตัวอย่างจะแสดงว่ามีซอฟต์แวร์ชื่อ Github ที่ได้ชื่อลิขสิทธิ์มาโดย

มีเงื่อนไขว่า ต้องลงโปรแกรม u-center และ jenkins ด้วยถึงใช้งานได้ ตัวโปรแกรมต้องลงบนเครื่องที่มี ram 32Gb และมี storage คือ 4 Tb

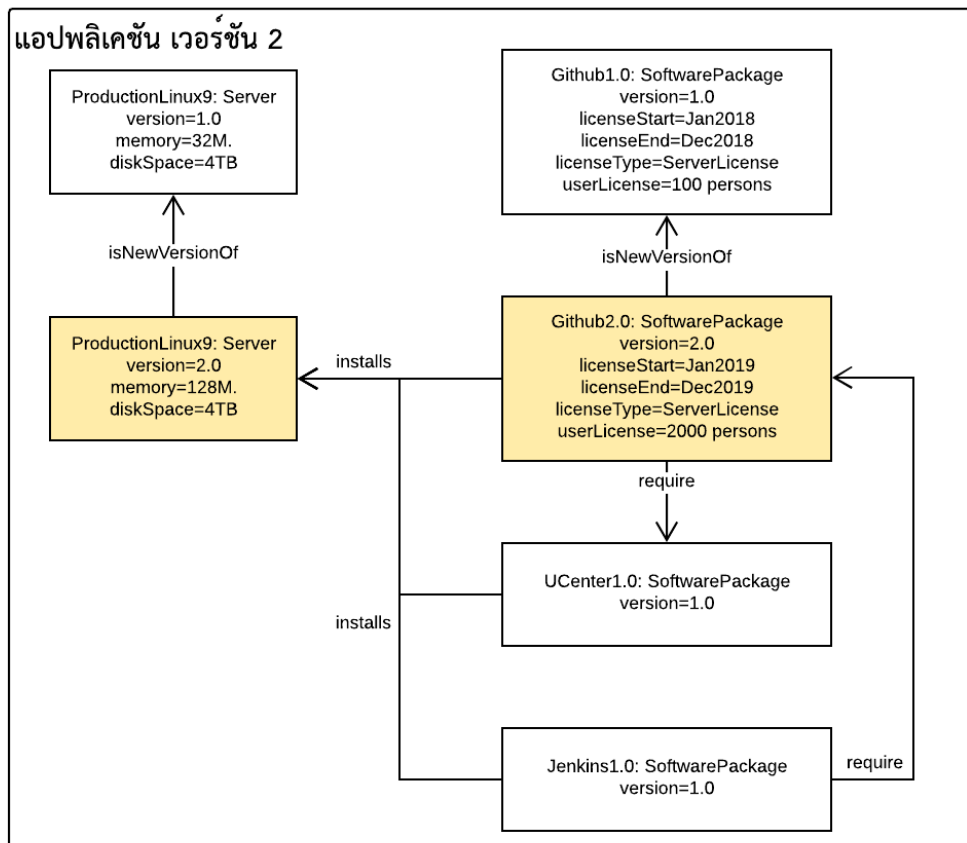
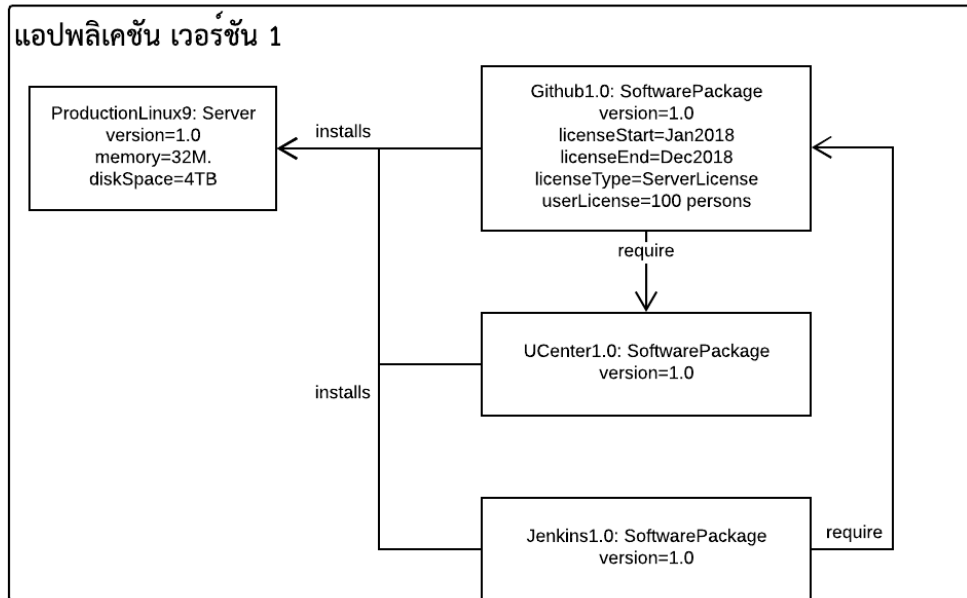
ภายหลังได้มีการอัปเดตเวอร์ชันของซอฟต์แวร์ Github จึงได้ทำการติดตั้งใหม่พร้อมทั้งอัปเดตเซิร์ฟเวอร์ให้รองรับโดย ซอฟต์แวร์ Github เปลี่ยนเป็นเวอร์ชัน 2.0 และ เซิร์ฟเวอร์ได้ทำการเพิ่มหน่วยความจำเป็น 128Gb ในรูปที่ 3.10 จะแสดงให้เห็นการเชื่อมโยงเมื่อมีการอัปเดตเซิร์ฟเวอร์และซอฟต์แวร์ซึ่งจะเป็นความสัมพันธ์ชื่อ IsNewVersionOf แสดงว่าเป็นตัวเวอร์ชันใหม่ของซีไอเดิม

และเพื่อแสดงให้เห็นว่ามีการจัดเก็บข้อมูลซีไอฮาร์ดแวร์และเครือข่ายที่มีการเปลี่ยนแปลง ได้ทำการจำลองแผนภาพ ดังแสดงในรูปที่ 3.11 จะแสดงให้เห็นว่ามีเครือข่ายภายในชื่อ lanProduction และ lanDR เชื่อมต่อเครือข่ายภายนอกชื่อ wan โดย lanProduction แสดงถึงเครือข่ายที่ใช้งานจริงมีอุปกรณ์เชื่อมต่อคือไฟร์วอลล์ชื่อ firewallProduction ซึ่งจะมีความสัมพันธ์ชื่อ resides กับ lanProduction หมายถึงอยู่ภายในเครือข่าย lanProduction มีเราเตอร์ชื่อ routerProduction สวิตช์ชื่อ swiftchA และเซิร์ฟเวอร์ชื่อ ProductionLinux1 ซึ่งอุปกรณ์ทั้งหมดจะมีความสัมพันธ์ชื่อ connects เชื่อมต่อกัน และ lanDR คือเครือข่ายสำรอง มีอุปกรณ์เชื่อมต่อคือไฟร์วอลล์ชื่อ firewallDR ซึ่งจะมีความสัมพันธ์ชื่อ resides กับ lanDR หมายถึงอยู่ภายในเครือข่าย lanDR เราเตอร์ชื่อ routerDR สวิตช์ชื่อ swiftchX และเซิร์ฟเวอร์ชื่อ ProductionLinux2 ซึ่งอุปกรณ์ทั้งหมดจะมีความสัมพันธ์ชื่อ connects เชื่อมต่อกันเช่นกัน

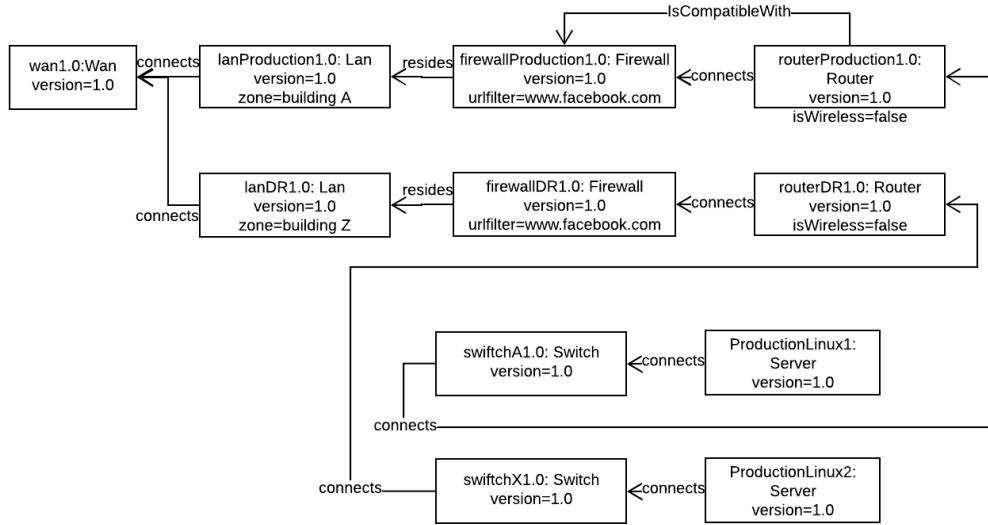
ในรูปที่ 3.12 จะแสดงให้เห็นกรณีที่มีการเพิ่มสวิตช์และเซิร์ฟเวอร์ใหม่ในแต่ละเครือข่าย คือมีการเพิ่มเซิร์ฟเวอร์ชื่อ ProductionLinux3, ProductionLinux4 และ สวิตช์ ชื่อ SwitchB, SwitchZ เข้ามา ซึ่งผลลัพธ์ที่ได้หลังการเพิ่มจะแสดงให้เห็นว่าเวอร์ชันของอุปกรณ์เดิมจะไม่มีเปลี่ยนแปลงเพียงแต่มีการเพิ่มเส้นความสัมพันธ์ใหม่เท่านั้น

ในรูปที่ 3.13 จะแสดงให้เห็นว่าถ้ามีการเปลี่ยนแปลงค่าของเครือข่ายภายใน เช่น เปลี่ยนโชนของเครือข่าย lanProduction ก็จะทำให้การเปลี่ยนเวอร์ชันจาก 1.0 เป็นเวอร์ชัน 1.1 มีการเปลี่ยนค่าของไฟร์วอลล์ชื่อ firewallProduction โดยเพิ่มค่า urlfilters ไฟร์วอลล์ก็จะทำการเปลี่ยนเวอร์ชันจาก 1.0 เป็นเวอร์ชัน 1.1 และเราเตอร์ชื่อ routerProduction ทำการเปิดฟังก์ชันไร้สายให้ใช้งานก็จะเปลี่ยนจากเวอร์ชัน 1.0 เป็นเวอร์ชัน 1.1 เช่นกัน ซึ่งซีไอใหม่นั้นจะมีความสัมพันธ์ชื่อ IsNewVersionOf กับ ซีไอเดิม

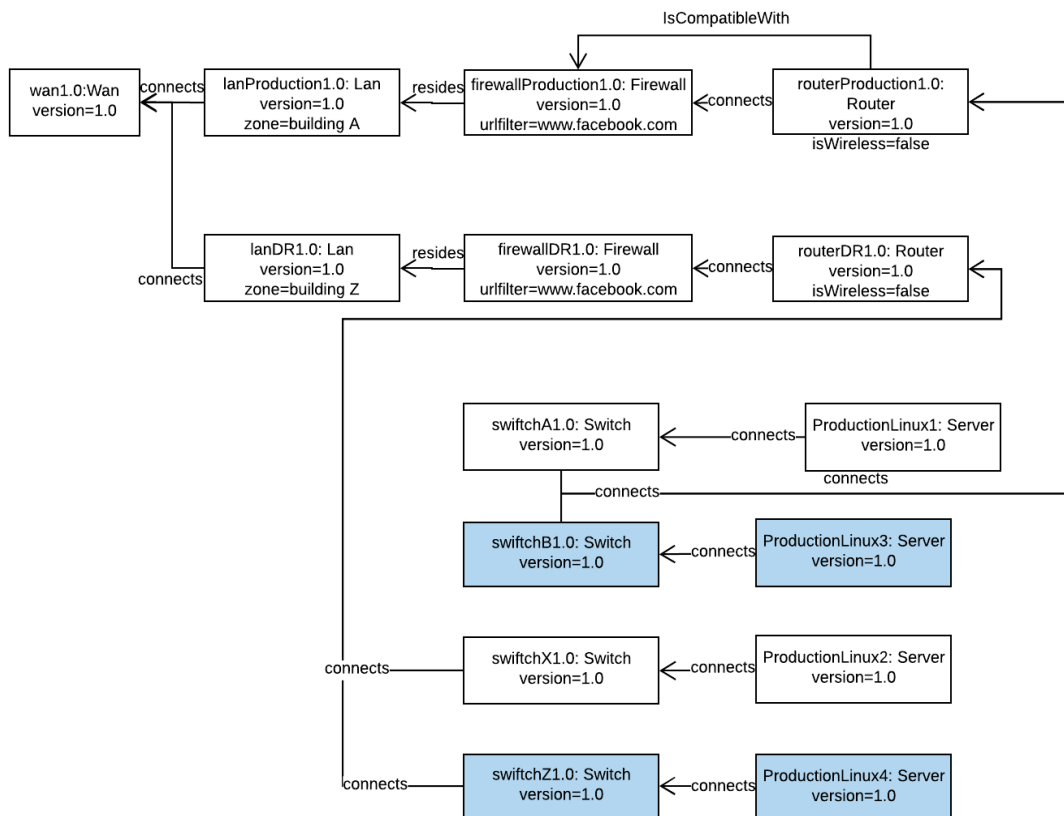




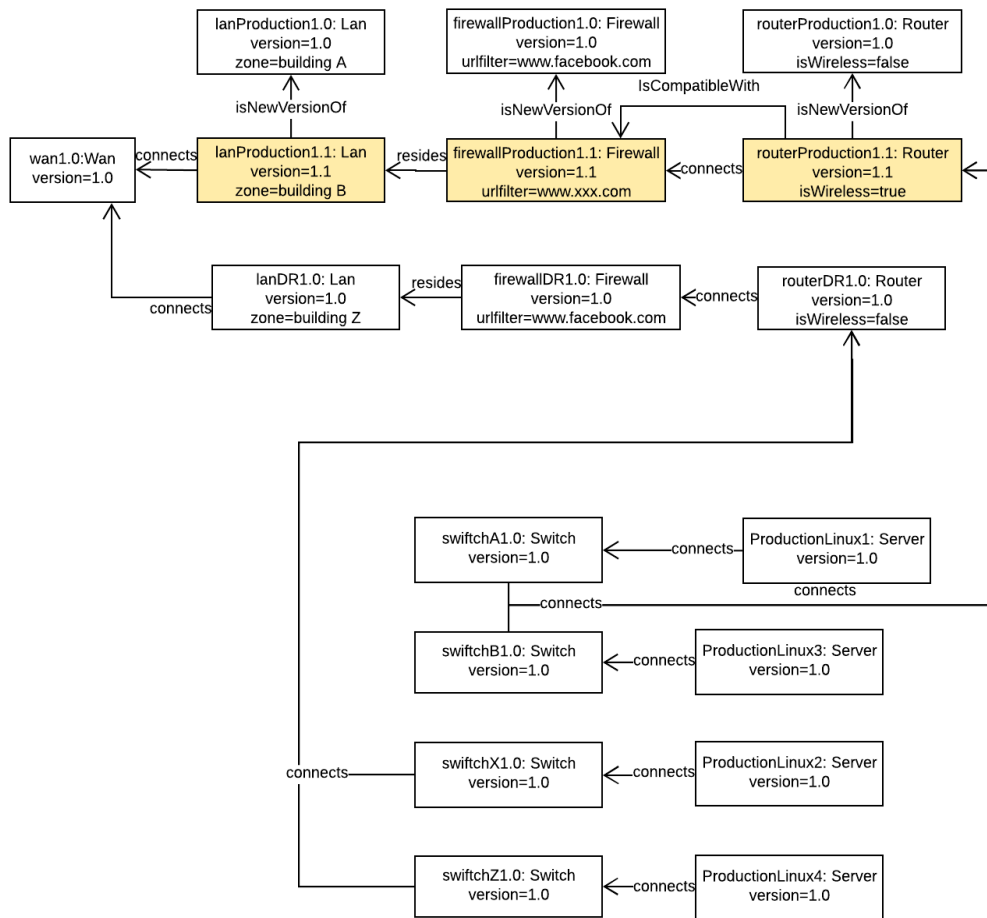
รูปที่ 3.10 แผนภาพวัตถุแสดงตัวอย่างซอฟต์แวร์เมื่อมีการอัปเดต



รูปที่ 3.11 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเบื้องต้น



รูปที่ 3.12 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเมื่อมีการเพิ่มฮาร์ดแวร์



รูปที่ 3.13 แผนภาพวัตถุแสดงตัวอย่างฮาร์ดแวร์และเครือข่ายเมื่อมีการเปลี่ยนแปลงค่า

จุฬาลงกรณ์มหาวิทยาลัย

### 3.3 การออกแบบโครงสร้างพื้นฐานออนไลน์

จากการออกแบบแผนภาพคลาส จะสามารถนำคุณสมบัติและความสัมพันธ์ของคลาสไปออกแบบโครงสร้างพื้นฐานออนไลน์ได้ ซึ่งประกอบด้วยแบบ คุณสมบัติแบบชนิดข้อมูล และคุณสมบัติแบบวัตถุ ดังที่ได้อธิบายมาก่อนหน้านี้แล้วในบทที่ 2 ผู้วิจัยเลือกใช้แนวทางในการแปลงข้อมูลแผนภาพคลาสไปเป็นโครงสร้างพื้นฐานออนไลน์และนำไปออกแบบโปรแกรมโปรเจกต์ ดังนี้

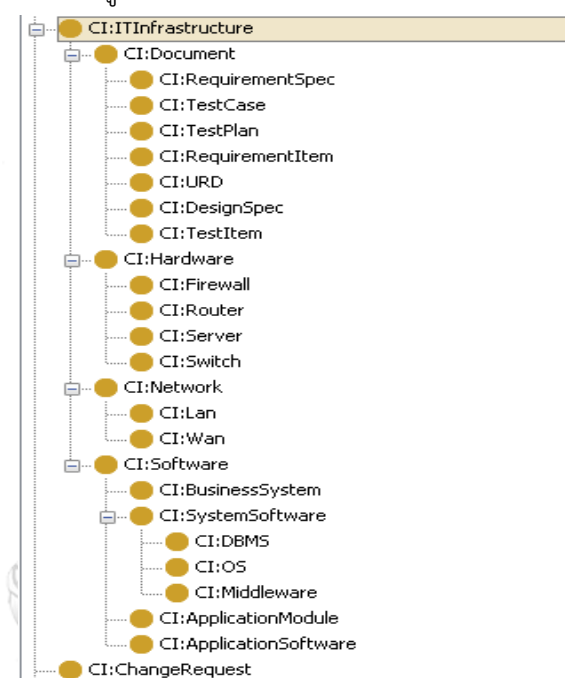
- 1) การแปลงคลาสที่ออกแบบได้ให้เป็นออนไลน์คลาสบนโปรแกรมโปรเจกต์
- 2) การแปลงคุณสมบัติของคลาสให้เป็นคุณสมบัติแบบชนิดข้อมูลบนโปรแกรมโปรเจกต์
- 3) การแปลงความสัมพันธ์ของคลาสให้เป็นคุณสมบัติแบบวัตถุบนโปรแกรมโปรเจกต์

ผู้วิจัยขออธิบายรายละเอียดของการแปลงในหัวข้อต่อไปนี้

### 3.4.1 การแปลงคลาสที่ออกแบบได้ให้เป็นออนโทโลยีคลาสบนโปรแกรมโปรเทจ

เริ่มต้นการออกแบบออนโทโลยีคลาสบนโปรแกรมโปรเทจ โดยพิจารณาจากแผนภาพคลาสตามรูปที่ 3.3 เริ่มต้นสร้างออนโทโลยีคลาสชื่อ ITInfrastructure

ในออนโทโลยีคลาสขั้นถัดไป มีออนโทโลยีคลาสชื่อ Hardware ,Network ,Software, และ Document เป็นออนโทโลยีคลาสย่อยของออนโทโลยีคลาส ITInfrastructure จากนั้นทำการสร้างเป็นออนโทโลยีคลาสย่อยลงมาจนครบตามที่ออกแบบแผนภาพคลาสไว้ ดังแสดงรูปภาพจอบนโปรแกรมโปรเทจที่ออกแบบคลาสในรูปที่ 3.14

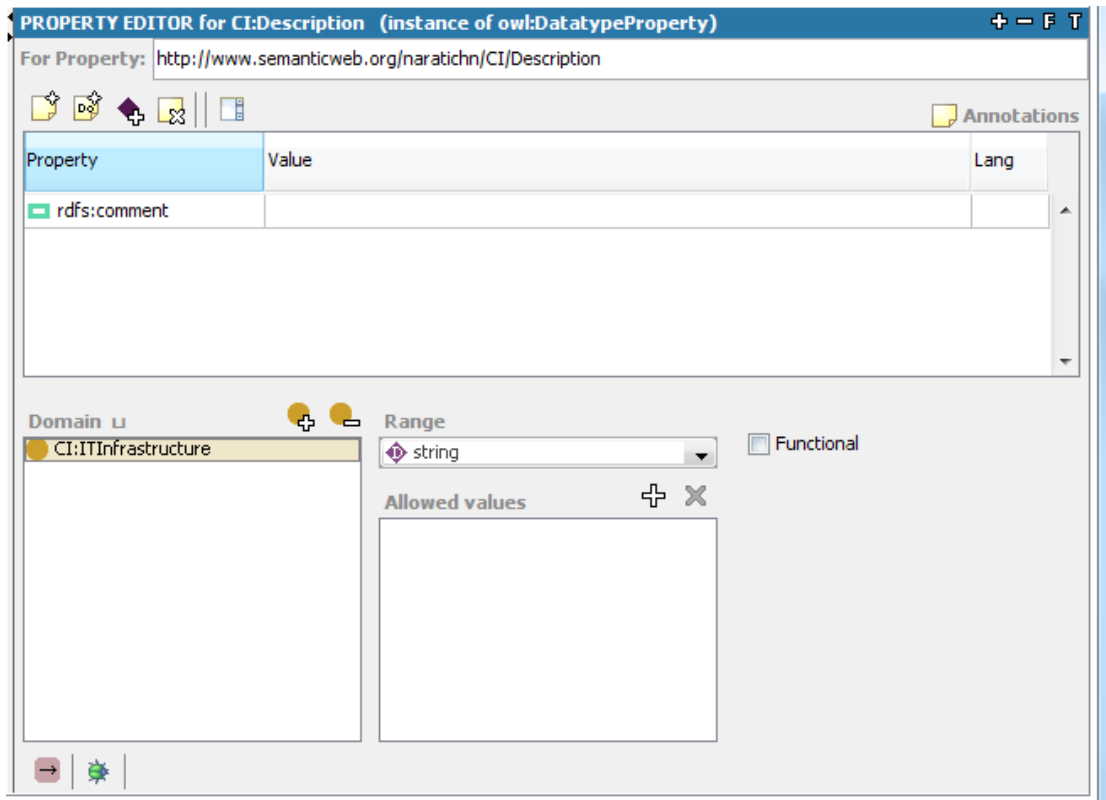


รูปที่ 3.14 การออกแบบออนโทโลยีคลาสบนโปรแกรมโปรเทจ

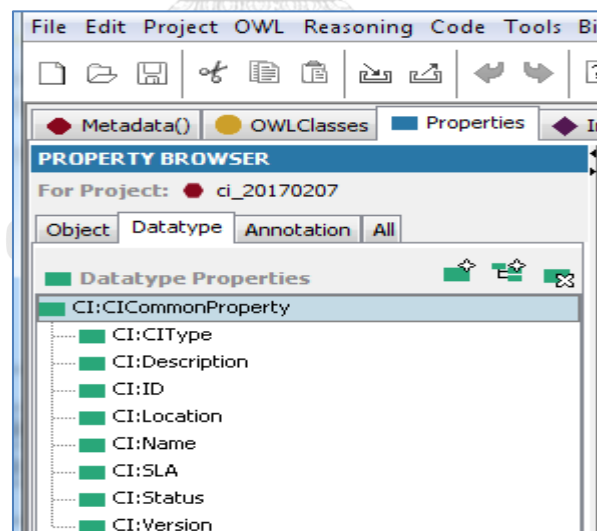
CHULALONGKORN UNIVERSITY

### 3.4.2 การแปลงคุณสมบัติของคลาสให้เป็นคุณสมบัติแบบชนิดข้อมูลบนโปรแกรมโปรเทจ

การแปลงคุณสมบัติของคลาสให้เป็นคุณสมบัติแบบชนิดข้อมูล จะนำคุณสมบัติของคลาสต่างๆ คลาสในแผนภาพคลาสที่ได้มาออกแบบเป็นคุณสมบัติแบบชนิดข้อมูลบนโปรแกรมโปรเทจ ยกตัวอย่างการสร้างคุณสมบัติชื่อ Description ของคลาส IT Infrastructure บนโปรแกรมโปรเทจ ปรากฏในรูปที่ 3.15 โดยจากรูปกำหนดให้มีค่า domain เป็น IT Infrastructure มีความหมายคือ เป็นคุณสมบัติของคลาส IT Infrastructure และมีค่า Range เป็น string มีความหมายคือ กำหนดชนิดข้อมูลเป็นตัวอักษร เป็นต้น ยกตัวอย่างเมื่อนำคุณสมบัติทั้งหมดของคลาส IT Infrastructure จากตารางที่ 3-4 มาออกแบบจะได้ผลตามรูปที่ 3.16 มีคุณสมบัติแบบชนิดข้อมูลคือ ID, CType, Name, Description, Location, Version, SupportBy และ Status



รูปที่ 3.15 ตัวอย่างการออกแบบคุณสมบัติแบบชนิดข้อมูล

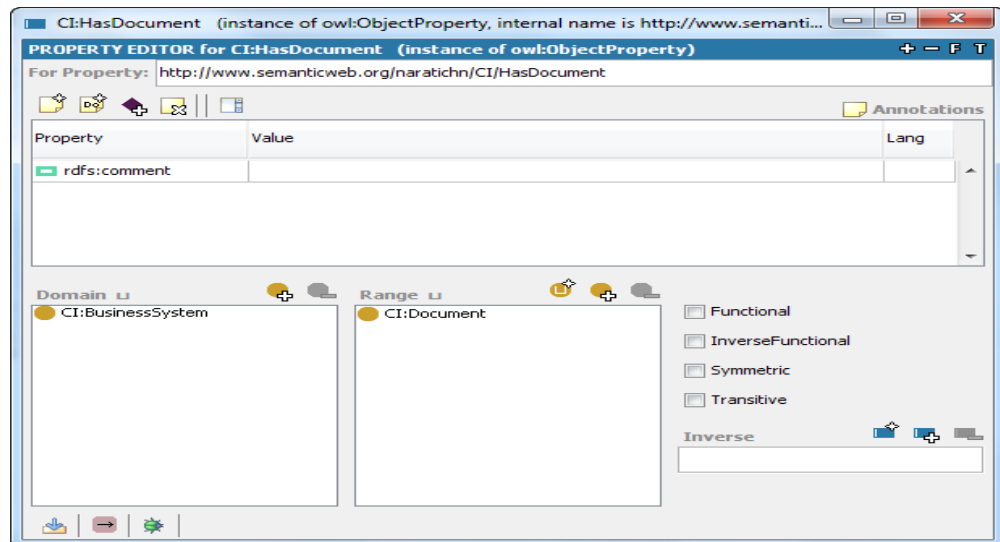


รูปที่ 3.16 ผลการออกแบบคุณสมบัติแบบชนิดข้อมูลของคลาส IT Infrastructure

### 3.4.3 การแปลงความสัมพันธ์ของคลาสให้เป็นคุณสมบัติแบบวัตถุบนโปรแกรมโปรเทจ

การแปลงความสัมพันธ์ของคลาสให้เป็นคุณสมบัติแบบวัตถุจะนำทุกๆ ความสัมพันธ์จากแผนภาพคลาสที่ออกแบบมาได้ไปออกแบบบนโปรแกรมโปรเทจ ยกตัวอย่างเช่น ความสัมพันธ์ชื่อ

HasDocument ที่ปรากฏในแผนภาพคลาส ถูกออกแบบบนโปรแกรมโปรเตจได้ตามรูปที่ 3.17 โดยจากรูปกำหนดให้ Domain คือคลาส BusinessSystem มีค่า Range เป็น Document มีความหมายคือ เป็นคุณสมบัติที่บอกว่า มีเอกสารที่เกี่ยวข้อง โดยที่มีความสัมพันธ์เริ่มจากคลาส BusinessSystem มาสู่คลาสชื่อ Document



รูปที่ 3.17 ตัวอย่างการสร้างคุณสมบัติแบบวัตถุชื่อ HasDocument

หลังจากที่ได้ออกแบบโครงสร้างข้อมูลออนโทโลยีด้วยโปรแกรมโปรเตจเสร็จแล้วจะนำข้อมูลออนโทโลยีไปใช้งานได้ โดยทำการบันทึกให้เป็นแฟ้มข้อมูลแบบอวาล์และจัดการนำเข้าบนเครื่องเซิร์ฟเวอร์ที่ติดตั้งจินาฟูซิกิ หลังจากนั้นจะออกแบบโปรแกรม และพัฒนาโปรแกรมด้วยภาษาจาวา ซึ่งจะอธิบายในบทถัดไป

## บทที่ 4

### การพัฒนาโปรแกรมสำหรับการจัดการโครงแบบ

บทนี้จะกล่าวถึงขั้นตอนการพัฒนาโปรแกรมสำหรับการจัดการโครงแบบ ซึ่งเริ่มจากการออกแบบแผนภาพยูสเคสและแผนภาพคลาส ออกแบบสถาปัตยกรรมของระบบ จากนั้นออกแบบแผงผังเว็บแอปพลิเคชัน และดีไซน์หน้าเว็บแอปพลิเคชัน และ สร้างฐานข้อมูลออนไลน์ด้วย ด้งรายละเอียดต่อไปนี้

#### 4.1 การออกแบบแผนภาพยูสเคส

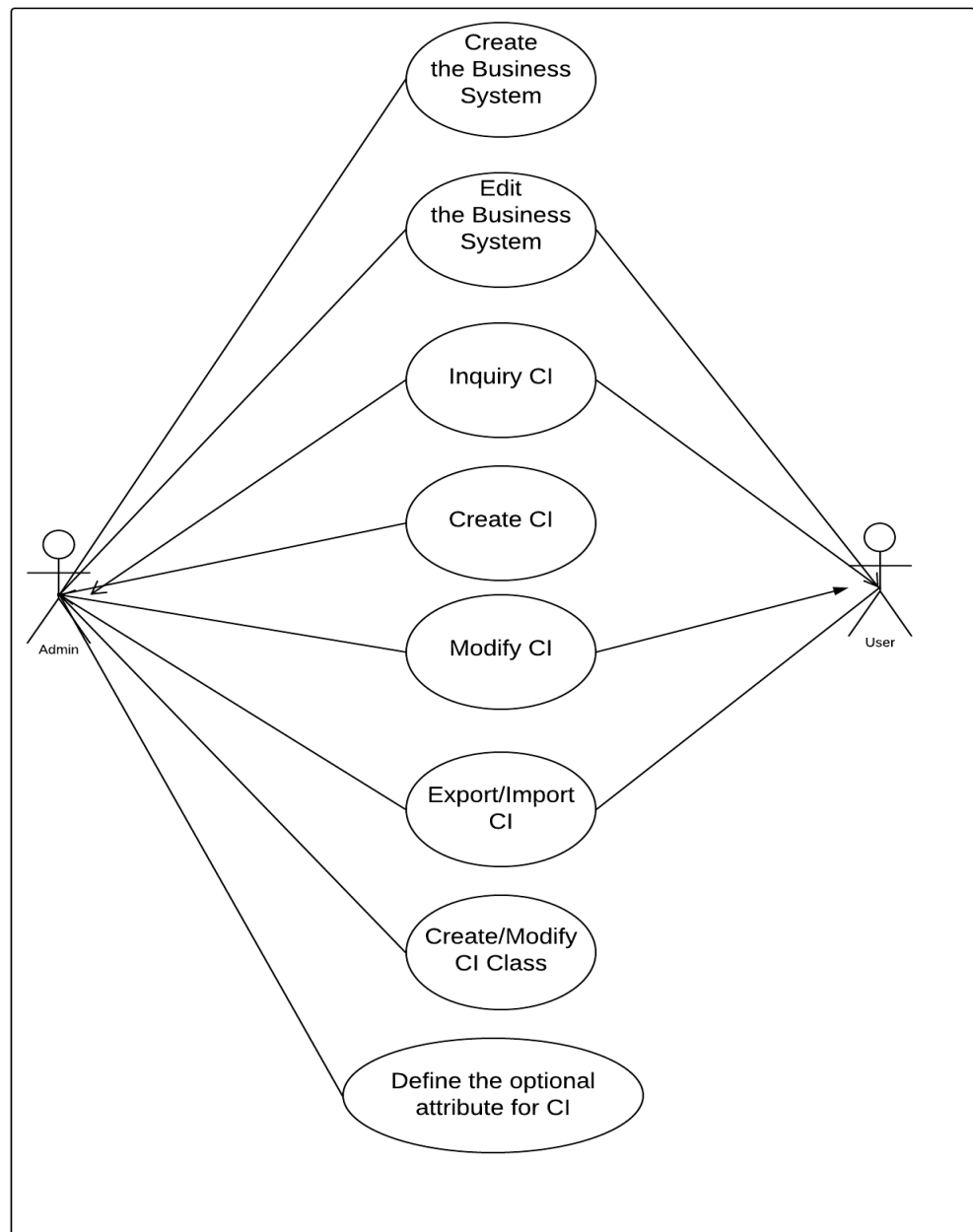
เริ่มต้นการออกแบบจะทำการแยกฟังก์ชันที่จะใช้งานแล้วออกแบบลงในแผนภาพยูสเคสดังแสดงในแผนภาพยูสเคสรูปที่ 4.1 ซึ่งจะประกอบไปด้วย กิจกรรมดังนี้

- 1) Create the Business System เป็นส่วนของการสร้าง Business System ใหม่
- 2) Edit the Business System เป็นส่วนที่เข้าไปเพื่อเลือกการจัดการ Business System เดิม
- 3) Inquiry CI เป็นส่วนที่ใช้ค้นหาและดูข้อมูลซีไอ
- 4) Create CI เป็นส่วนที่สร้างข้อมูลซีไอ
- 5) Modify CI เป็นส่วนที่แก้ไขข้อมูลในซีไอ
- 6) Create/Modify CI Class เป็นส่วนที่ทำการสร้างหรือแก้ไขคลาสของซีไอ
- 7) Define the optional attribute for CI เป็นส่วนที่เพิ่มคุณสมบัติของซีไอ

และรูปที่ 4.2 แสดงแผนภาพกิจกรรมแสดงการสร้างและแก้ไขซีไอ ซึ่งแสดงให้เห็นว่า Admin ขณะทำหน้าที่สร้างคลาสซีไอและข้อมูลซีไอก่อนส่วน User สามารถแก้ไขหรือกรอกข้อมูลซีไอได้เท่านั้น หากต้องการจะเพิ่มคุณสมบัติจะต้องขอให้ทาง Admin ทำการเพิ่ม

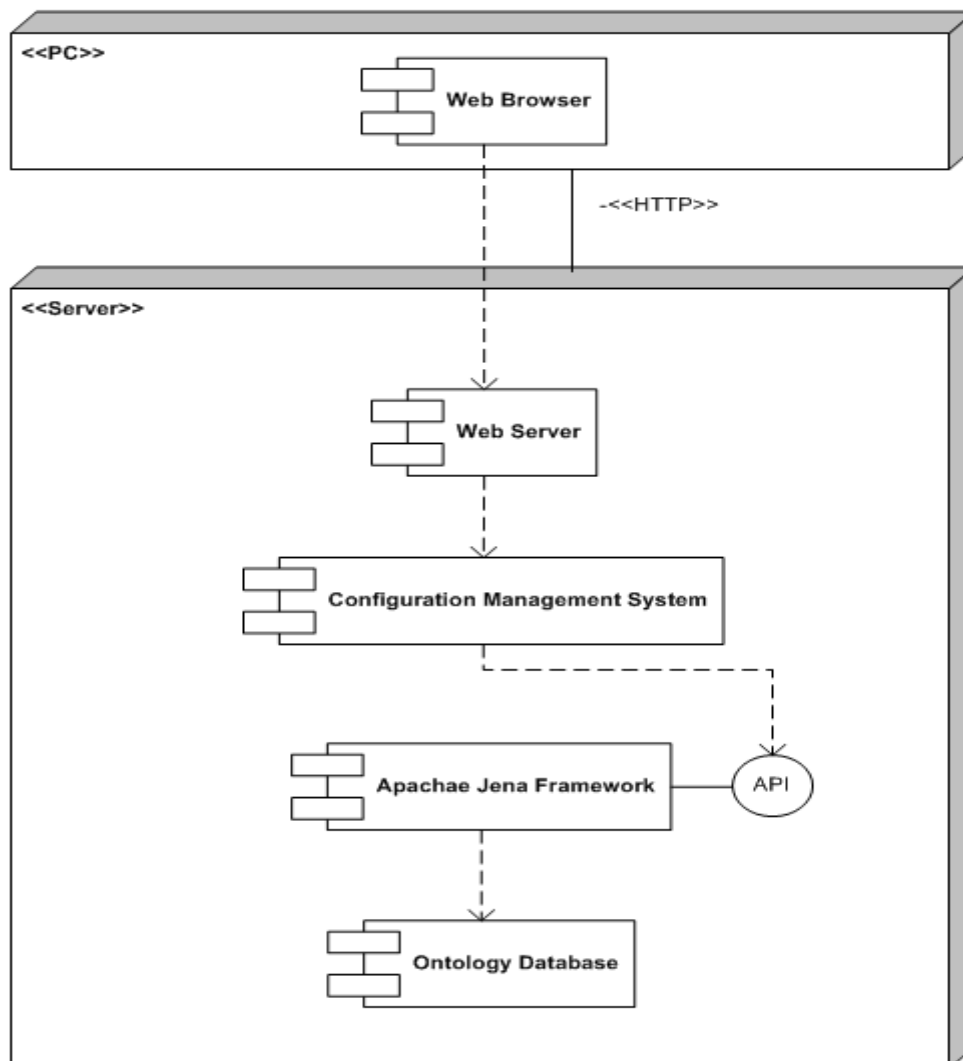
#### 4.2 การออกแบบสถาปัตยกรรมของระบบการจัดการโครงแบบ

การออกแบบสถาปัตยกรรมของระบบแสดงดังรูปที่ 4.2 ส่วนของการพัฒนา Ontology Database ได้อธิบายไปแล้วในบทที่ผ่านมาซึ่งอยู่บนกรอบงานของอาปาเซจินา อาปาเซจินาได้มีบริการส่วนต่อประสานโปรแกรมประยุกต์ (API) และปลั๊กอิน (Plugin) เป็นคลาสในภาษาจาวา (Java) การพัฒนาโปรแกรมการจัดการโครงแบบทางผู้พัฒนาจึงได้เลือกพัฒนาเป็นแบบเว็บแอปพลิเคชันด้วยภาษาจาวาโดยใช้เครื่องมือ Eclipse และเว็บเซิร์ฟเวอร์คือ Tomcat



รูปที่ 4.1 แผนภาพยูสเคสระบบจัดการโครงแบบ



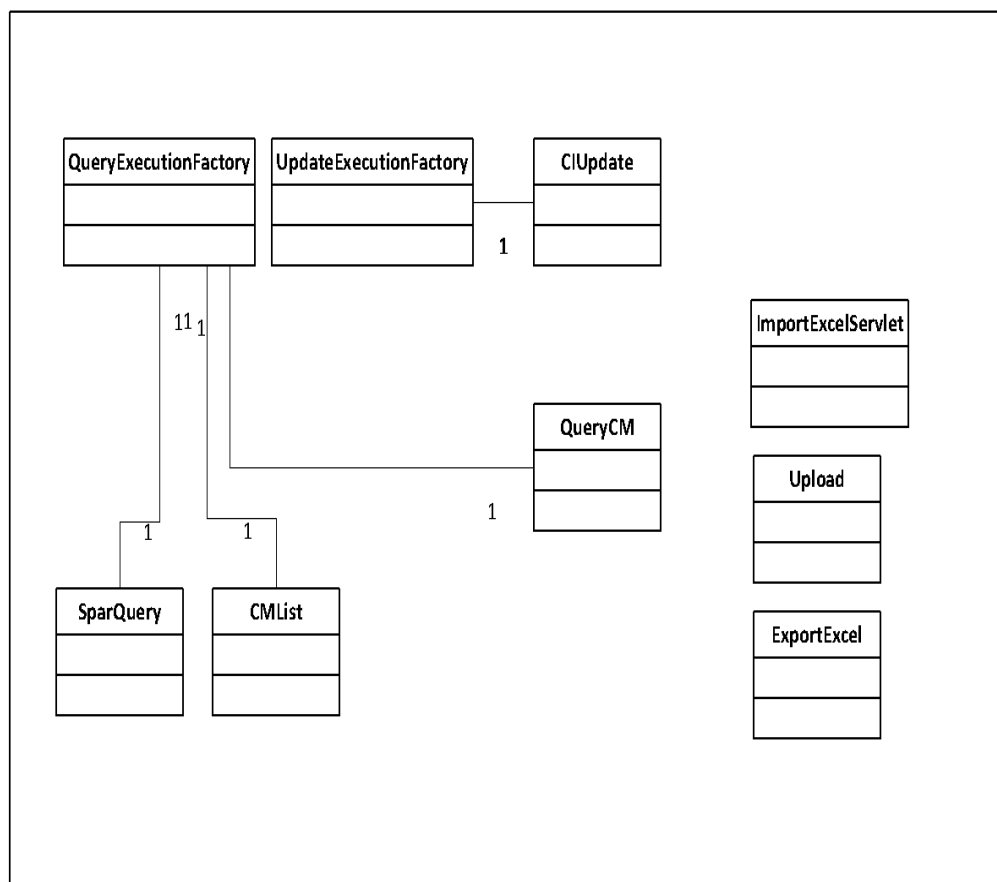


รูปที่ 4.2 แผนภาพสถาปัตยกรรมของระบบ  
CHULALONGKORN UNIVERSITY

### 4.3 การออกแบบแผนภาพคลาสและแผนภาพลำดับ

จากสถาปัตยกรรมของระบบดังรูปที่ 4.2 โปรแกรมจะออกแบบให้มีคลาสที่ช่วยในการติดต่อกับส่วนต่อประสานโปรแกรมประยุกต์ของฟุเซกิ ซึ่งส่วนต่อประสานโปรแกรมประยุกต์ของฟุเซกิจะแยกยูอาร์แอลสำหรับการสืบค้นและการอัปเดตดังนั้นโปรแกรมจึงออกแบบให้มีคลาสในการติดต่อ 2 คลาสคือ QueryCM และ CIUpdate โดยที่ QueryCM จะทำหน้าที่ในการสืบค้นและ CIUpdate ทำหน้าที่ในการอัปเดต โดยทำการเรียกไปยังคลาสที่อาปาเซจินาให้มาคือ org.apache.jena.query.QueryExecution และ org.apache.jena.update.UpdateExecutionFactory

ในส่วนของโปรแกรมส่วนอื่นได้ออกแบบคลาสชื่อ CMList ไว้ให้รองรับการแสดงผลบนหน้าจอ หน้าเว็บยังมีฟังก์ชันให้แสดงเส้นกราฟเชื่อมโยงแต่ละซีไอซึ่งได้สร้างคลาสชื่อ SparQuery เพื่อสร้างกราฟ สำหรับฟังก์ชันการนำเข้าจะมีสองคลาสที่เกี่ยวข้องคือคลาส Upload จะทำหน้าที่รองรับการอัปโหลดไฟล์จากหน้าเว็บจากนั้นจะเรียกใช้งานคลาส ImportExcelServlet เพื่อนำเข้าข้อมูลจากไฟล์เอ็กเซล ส่วนการนำข้อมูลออกจะเรียกใช้งานคลาส ExportExcel ซึ่งทำหน้าที่แปลงข้อมูลซีไอออกมาเป็นแฟ้มข้อมูลเอ็กเซล ซึ่งสามารถแสดงแผนภาพคลาสได้ดังรูปที่ 4.3 และ แผนภาพคลาสโดยละเอียดของคลาสที่ติดต่อกับส่วนต่อประสานโปรแกรมประยุกต์ของฟูเซกิในรูปที่ 4.4



รูปที่ 4.3 แผนภาพคลาสหลักของโปรแกรมจัดการโครงแบบ

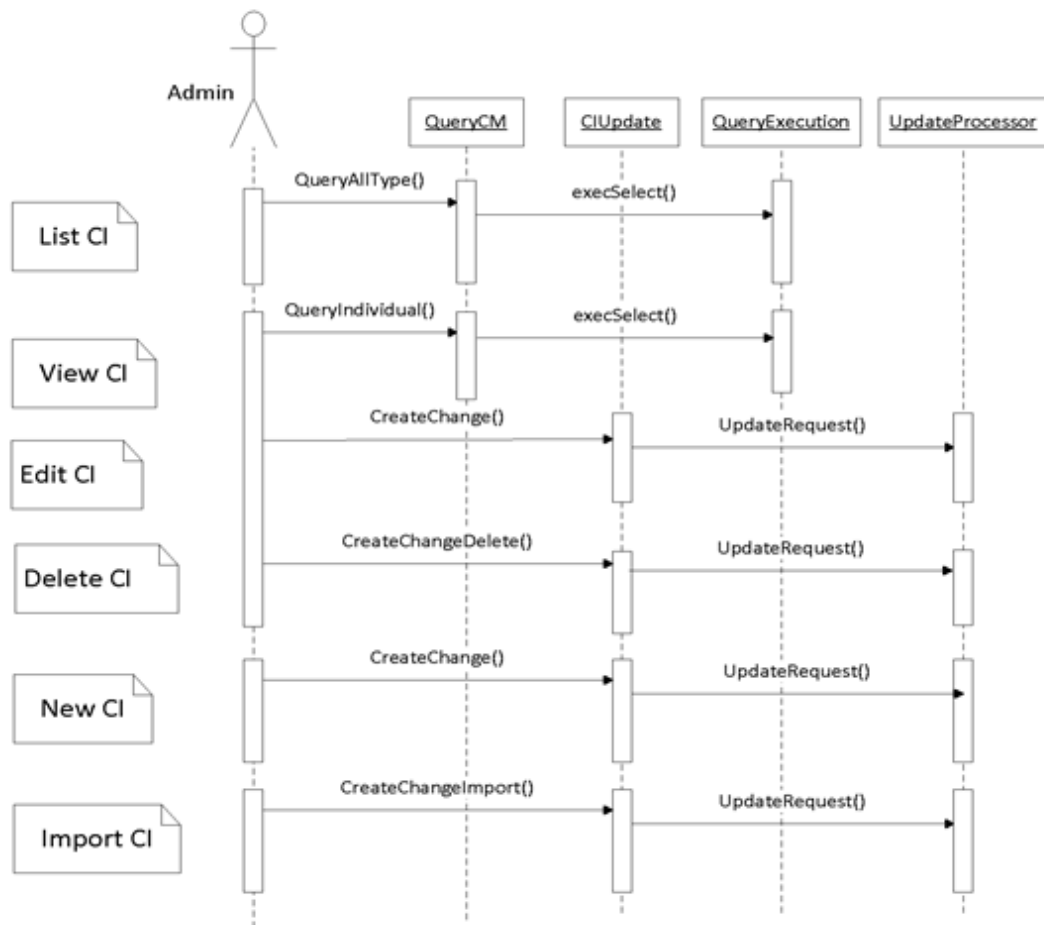


รูปที่ 4.4 แผนภาพคลาสของโปรแกรมจัดการโครงแบบ

สำหรับแผนภาพลำดับแสดง ในรูปที่ 4.5 จะเป็นการแสดงลำดับการเรียกใช้งานแต่ละฟังก์ชันคือ List CI, View CI, Edit CI, New CI, Import CI คือ

1) ฟังก์ชัน List CI, View CI จะเรียกใช้งานคลาส QueryCM ก่อนจากนั้นคลาส QueryCM จะไปเรียกฟังก์ชัน execSelect() ซึ่งเป็นเอพีไอไปสืบค้นข้อมูลจากจินาฟูเซกิ

2) ฟังก์ชัน Edit CI, New CI และ Import CI จะเรียกใช้งานคลาส CIUpdate ก่อนจากนั้นคลาส CIUpdate จะไปเรียกฟังก์ชัน UpdateRequest() ซึ่งเป็นเอพีไอไปอัปเดตข้อมูลบนจินาฟูเซกิ



รูปที่ 4.5 แผนภาพลำดับการใช้งานแต่ละฟังก์ชัน

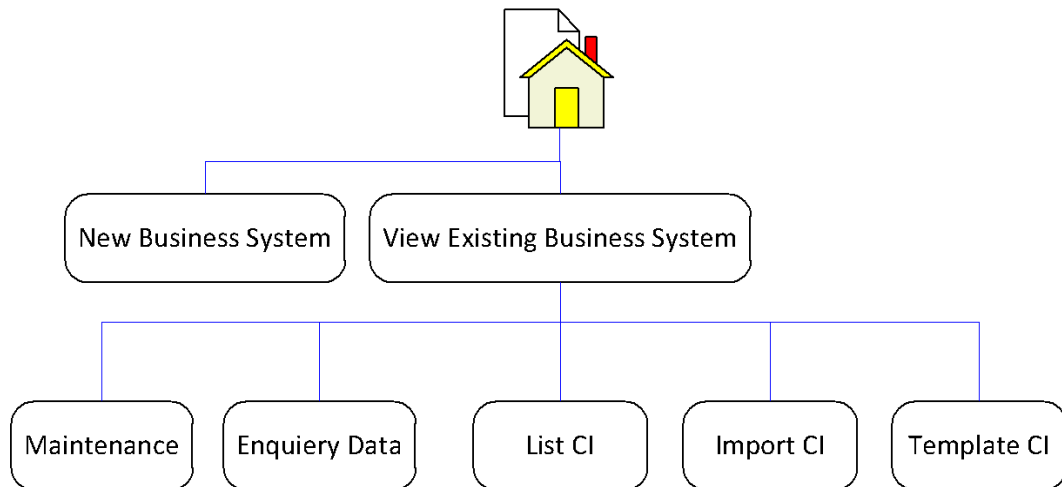
จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

#### 4.4 การออกแบบแผนผังเว็บแอปพลิเคชันการจัดการโครงแบบ

การออกแบบแผนผังยึดฟังก์ชันที่ได้ออกแบบไว้ในแผนภาพยูสเคสมาเป็นแต่ละเว็บเพจ ซึ่งแผนผังเว็บแอปพลิเคชันการจัดการโครงแบบแสดงในรูปที่ 4.6 มีรายละเอียดแต่ละหน้าคือ

ส่วนแรกเมื่อเข้าไปจะให้เลือกหน้า New Business System และ View Existing Business และเมื่อเลือก View Existing Business จะมีเมนูให้เข้าหน้า Maintenance, Inquiry Data, List CI, Import CI และ Template CI

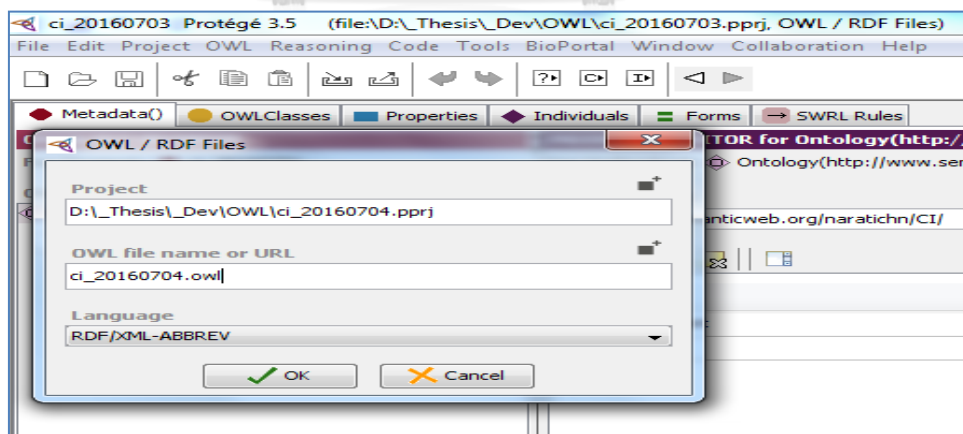


รูปที่ 4.6 แผนผังเว็บไซต์การจัดการโครงแบบ

#### 4.5 การสร้างฐานข้อมูลออนไลน์

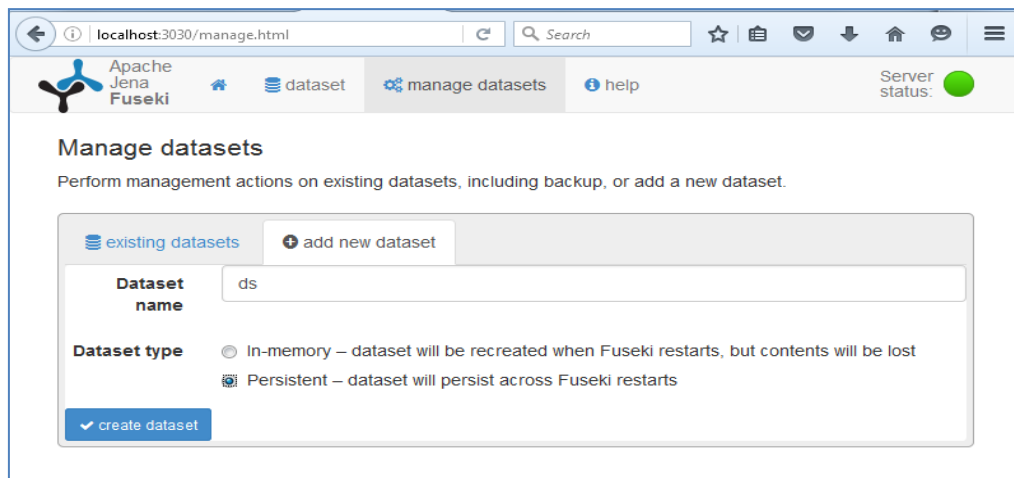
ในส่วนของการสร้างฐานข้อมูลออนไลน์นั้นหลังจากที่ได้ออกแบบโครงร่างข้อมูลออนไลน์เรียบร้อยแล้วการจะนำข้อมูลออนไลน์ไปใช้งานจะมีขั้นตอนในการนำไปติดตั้งบนจินาฟูซิกิ ซึ่งจะอธิบายตามขั้นตอนต่อไปนี้

- 1) บันทึกข้อมูลภาษาอวาล์จากโปรแกรมโพรเทเจเป็นไฟล์เป็นข้อมูลชนิด RDF/XML-ABBREV ดังรูปที่ 4.7 ซึ่งเราจะได้ข้อมูลเป็นแฟ้ม .owl มา



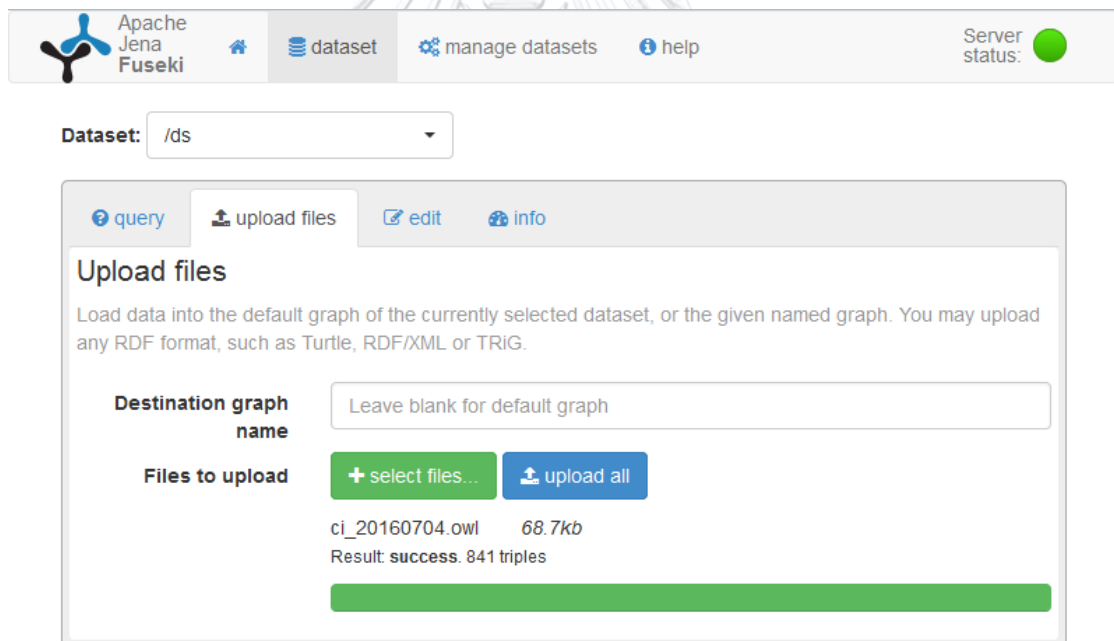
รูปที่ 4.7 ภาพการบันทึกข้อมูลจากโปรแกรมโพรเทเจ

- 2) สร้างชุดข้อมูล (DataSet) บนอาปาเซจินาฟูซิกิ ชื่อชุดข้อมูลคือ “ds” ตามรูปที่ 4.8



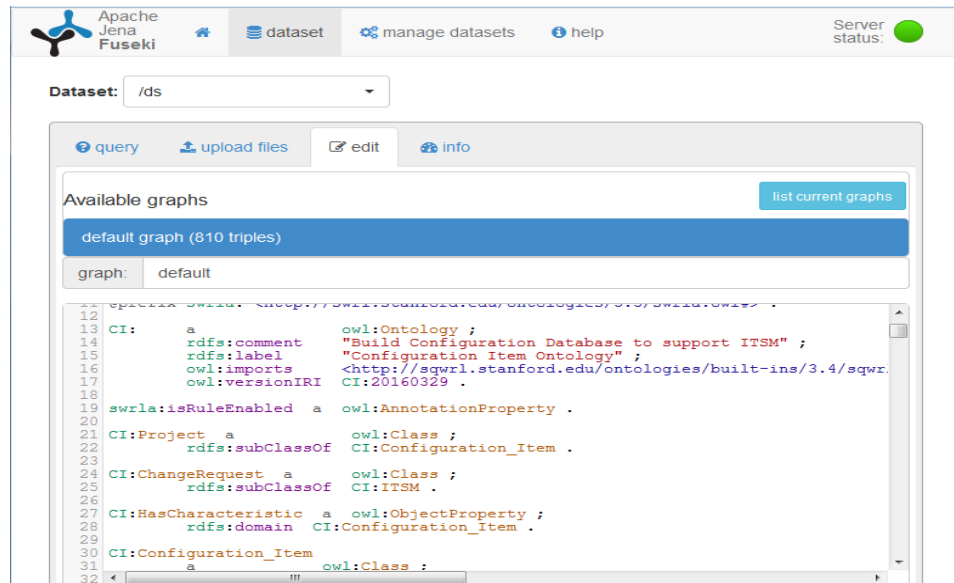
รูปที่ 4.8 หน้าจอเมนูส่วนการจัดการชุดข้อมูลบนอปาเซจินาฟูเซกิ

3) อัปโหลดข้อมูลภาษาอวาล์ ตามรูปที่ 4.9



รูปที่ 4.9 หน้าจอเมนูส่วนการจัดการชุดข้อมูลบนอปาเซจินาฟูเซกิ

4) ตรวจสอบข้อมูลที่ได้อัปโหลดไปแล้วจากเมนู “edit” ดังรูปที่ 4.10 ซึ่งจากรูปแสดงให้เห็นว่าเราสร้างชุดข้อมูลออนโทโลยีตามที่ได้ออกแบบมาเสร็จเรียบร้อยแล้ว



Apache Jena Fuseki

Server status: ●

Dataset: /ds

query upload files edit info

Available graphs list current graphs

default graph (810 triples)

graph: default

```
12 prefix swrl: <http://swrl.stanford.edu/ontologies/3.4/swrl.owl#> .
13 CI: a owl:Ontology ;
14   rdfs:comment "Build Configuration Database to support ITSM" ;
15   rdfs:label "Configuration Item Ontology" ;
16   owl:imports <http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#> .
17   owl:versionIRI CI:20160329 .
18
19 swrla:isRuleEnabled a owl:AnnotationProperty .
20
21 CI:Project a owl:Class ;
22   rdfs:subClassOf CI:Configuration_Item .
23
24 CI:ChangeRequest a owl:Class ;
25   rdfs:subClassOf CI:ITSM .
26
27 CI:HasCharacteristic a owl:ObjectProperty ;
28   rdfs:domain CI:Configuration_Item .
29
30 CI:Configuration_Item
31   a owl:Class ;
32
```

รูปที่ 4.10 หน้าจอแสดงผลข้อมูลภายในชุดข้อมูล “ds”



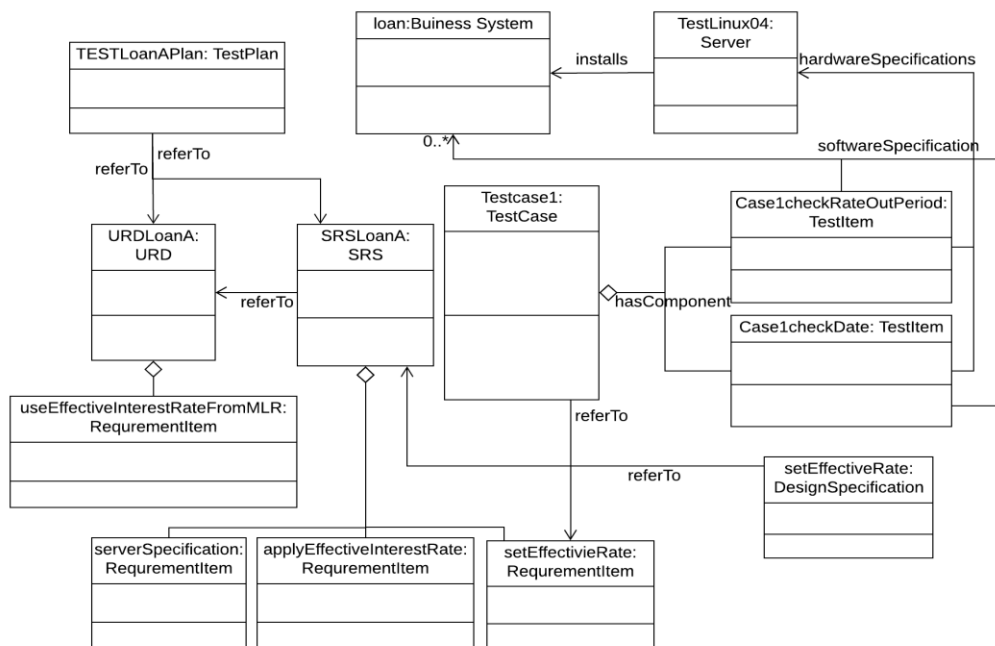
## บทที่ 5

### กรณีศึกษาสำหรับระบบการจัดการโครงแบบ

ในบทนี้จะแสดงการใช้งานระบบการจัดการโครงแบบโดยจำลองข้อมูลระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจของระบบเงินกู้ของธนาคาร โดยมีข้อมูลซีไอและซีไอแอกกรีเกตที่สามารถทดสอบได้ครบถ้วนตามที่ได้ระบบออกแบบไว้ โดยจะทำการทดสอบตามขั้นตอน ดังรายละเอียดต่อไปนี้

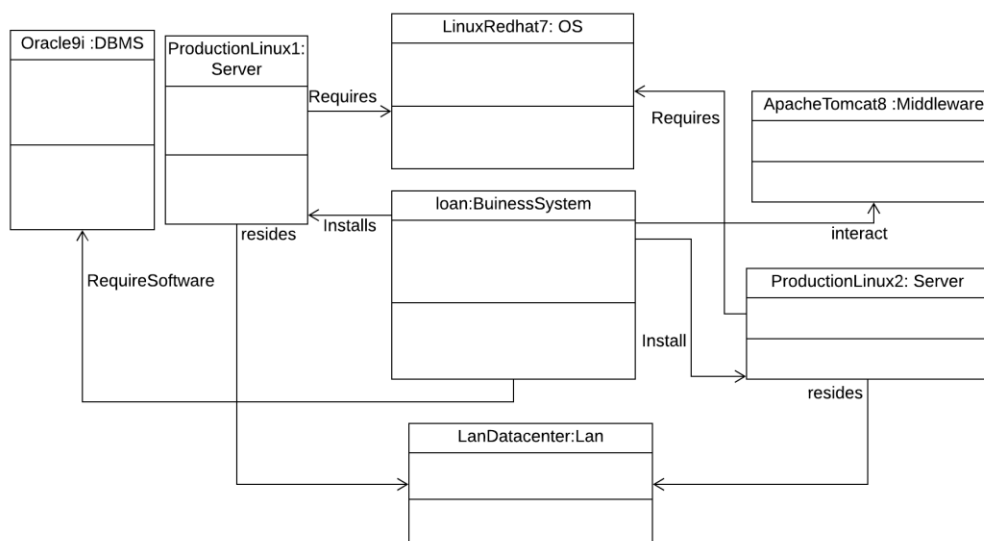
#### 5.1 ออกแบบกรณีศึกษาสร้างข้อมูลซอฟต์แวร์ที่ให้บริการทางธุรกิจของธนาคาร

ระบบธนาคารมีซอฟต์แวร์ที่ให้บริการทางธุรกิจคือระบบกู้ยืม ชื่อว่า Loan ประกอบไปด้วยเครือข่าย ฮาร์ดแวร์ ซอฟต์แวร์ และ เอกสาร ในส่วนของเอกสารจะจำลองสร้างเอกสารตามที่เคยออกแบบไว้ในรูปที่ 3.7 แผนภาพวัตถุของเอกสาร ซึ่งจะได้อุปแบบแผนภาพวัตถุแบบพื้นฐานดังรูปที่ 5.1 ซึ่งจะประกอบไปด้วย เอกสารข้อกำหนดความต้องการ ชื่อ SRSLoanA เอกสารการออกแบบชื่อ setEffectiveRate แผนการทดสอบชื่อ TESTLoanA เอกสารข้อกำหนดความต้องการ SRSLoanA เอกสารกรณีทดสอบชื่อ testcase1 และ testcase2 ในส่วนของเครือข่ายฮาร์ดแวร์และซอฟต์แวร์ได้ออกแบบตามรูปที่ 5.2 ประกอบไปด้วยเซิร์ฟเวอร์ ระบบปฏิบัติการ มิดเดิลแวร์ ฐานข้อมูลและเครือข่าย



รูปที่ 5.1 แผนภาพแบบวัตถุของระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจ Loan ส่วนของการทดสอบ





รูปที่ 5.2 แผนภาพแบบวัตถุของระบบซอฟต์แวร์ที่ให้บริการทางธุรกิจ Loan ส่วนของเครือข่าย ฮาร์ดแวร์และซอฟต์แวร์

## 5.2 ทดสอบใช้งานโปรแกรมการจัดการโครงแบบ

จากที่ได้ออกแบบกรณีทดสอบไปตามข้อ 5.1 การสร้างข้อมูลจะสร้างบนโปรแกรมการจัดการโครงแบบ โดยเปิดเบราว์เซอร์ไปยังเว็บโปรแกรมการจัดการโครงแบบจะแสดงหน้าจอดังรูปที่ 5.3 โดยจะมีเมนูดังนี้

- 1) Home - หน้าแรกเมื่อเข้าระบบมา
- 2) New Business System เป็นหน้าของการสร้าง Business System ใหม่
- 3) View Existing Business System เป็นหน้าที่เข้าไปเพื่อเลือกการจัดการ Business System เดิม
- 4) Maintenance เป็นหน้าแสดงข้อมูลและมีส่วนจัดการ Business System ที่ได้เลือกไว้
- 5) Inquiry Data เป็นหน้าที่ใช้ออกรายงานชิ้นงานที่อยู่ใน Business System ที่ได้เลือกไว้
- 6) List CI เป็นหน้าลิสต์ซีไอทั้งหมดใน Business System ที่ได้เลือกไว้
- 7) Import CI เป็นหน้าของการนำเข้าข้อมูลซีไอจากไฟล์เอกซ์เซล
- 8) Template CI เป็นหน้าของการนำเข้าข้อมูลซีไอจากไฟล์เอกซ์เซล

รายละเอียดการทำงานในแต่ละหน้า จะนำเสนอในหัวข้อย่อยต่อไปนี้

### 5.2.1 หน้าแรกเมื่อเข้าระบบ

หน้าจอจะให้เลือกเมนูซึ่งมีให้เลือก New Business System และ View Existing Business System แสดงในรูปที่ 5.3



รูปที่ 5.3 หน้าจอหน้าแรกของโปรแกรมการจัดการโครงแบบ

### 5.2.2 หน้า New Business System

หน้าจอจะให้กรอกรายละเอียดของ Business System เพื่อสร้าง Business System ใหม่ ดังแสดงในรูปที่ 5.4 ซึ่งก็จะทำการป้อนข้อมูล Business System ใหม่ลงไป

รูปที่ 5.4 หน้าจอ New Business System

เมื่อกรอกข้อมูล Business System แล้วหน้าจะไปหน้าการเพิ่มข้อมูลฮาร์ดแวร์ดังรูปที่ 5.5 ซึ่งสามารถเลือกประเภทฮาร์ดแวร์และกรอก ดังแสดงรูปที่ 5.6 การกรอกข้อมูลของเซิร์ฟเวอร์ซึ่งเมื่อกรอกเสร็จแล้วจะประกฏข้อมูลในลิสต์ดังรูปที่ 5.7

## Ontology-Based Configuration Management System

[Home](#)

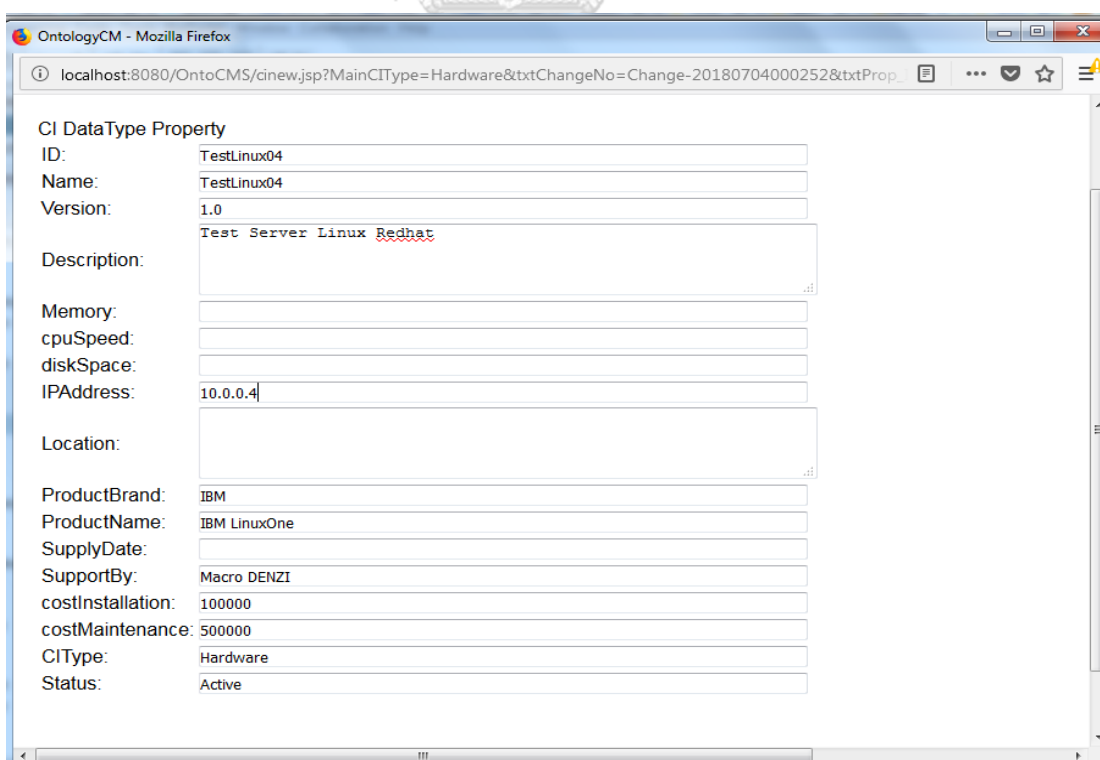
### New Project - Add Hardware

Add Hardware for Project\_Loan.

Select Hardware Type:

Server

รูปที่ 5.5 หน้าจอเพิ่มข้อมูลฮาร์ดแวร์



OntologyCM - Mozilla Firefox

localhost:8080/OntoCMS/cinew.jsp?MainCIType=Hardware&txtChangeNo=Change-20180704000252&txtProp...

CI DataType Property

ID: TestLinux04

Name: TestLinux04

Version: 1.0

Description: Test Server Linux Redhat

Memory:

cpuSpeed:

diskSpace:

IPAddress: 10.0.0.4

Location:

ProductBrand: IBM

ProductName: IBM LinuxOne

SupplyDate:

SupportBy: Macro DENZI

costInstallation: 100000

costMaintenance: 500000

CIType: Hardware

Status: Active

รูปที่ 5.6 หน้าจอกรอกข้อมูลเซิร์ฟเวอร์

## Ontology-Based Configuration Management System

[Home](#)

### New Project - Add Hardware

Add Hardware for Project\_Loan.

Select Hardware Type:

Server Add New Add Existing

Server:TestLinux04  
 Server:ProductionLinux1

Next Step

รูปที่ 5.7 หน้าจอผลของการกรอกข้อมูลส่วนฮาร์ดแวร์

เมื่อกดปุ่ม next step ส่วนถัดไปจะเป็นส่วนของการเพิ่มข้อมูลเอกสารดังแสดงในรูปที่ 5.8 จะมีลิสต์ของประเภทเอกสารที่ต้องการเพิ่ม ซึ่งรูปที่ 5.9 แสดงการกรอกข้อมูลเอกสารแผนการทดสอบ รูปที่ 5.10 แสดงการกรอกข้อมูลเอกสารความต้องการของผู้ใช้งาน รูปที่ 5.11 แสดงการกรอกข้อมูลเอกสารข้อกำหนดความต้องการ

## Ontology-Based Configuration Management System

[Home](#)

### New Project - Add Document

Add Document for Project\_Loan.

Select Document Type:

TestPlan Add New Add Existing

TestPlan  
 TestCase  
 TestItem  
 SRS  
 URD  
 DesignSpec  
 RequirementItem

Back Next Step

รูปที่ 5.8 หน้าจอส่วนของการเพิ่มข้อมูลเอกสาร

OntologyCM - Mozilla Firefox  
localhost:8080/OntoCMS/cinew.jsp?MainCIType=Document&txtChangeNo=Change-20180704000252&txtProp

### Add New TestPlan

CI DataType Property

ID: TESTLoanAPlan  
Name: TESTLoanAPlan  
Version: 1.0  
Description: Test Plan for Loan system with the effective rate feature

featuresToBeTested: loan report , loan calculator function  
featuresnotToBeTested: batch export data to other system  
itemPassFailCriteria: satisfy the standard requirements for system pass/fail stated in AAA company procedure  
schedule: UAT 1-10 May 2018  
scope: Test new calculation in Loan system  
documentName: TESTLoanAPlan  
documentVersion: 1.0  
objective: Define activities, test case and scope for testing  
CIType: Document  
Status: Active

รูปที่ 5.9 หน้าจอส่วนของการกรอกข้อมูลเอกสารแผนการทดสอบ

OntologyCM - Mozilla Firefox  
localhost:8080/OntoCMS/cinew.jsp?MainCIType=Document&txtChangeNo=Change-20180704000252&txtProp

### Add New URD

CI DataType Property

ID: URDLoanA  
Name: URDLoanA  
Version: 1.0  
Description: URD for Loan system with the effective rate feature

businessAnalyst: Tirapon L.  
projectManager: Tirawan K.  
projectOnwer: Nuanpan S.  
projectScope: Enhancement the interest fee.  
userRequirements:  
documentName: URD loanA  
documentVersion:  
objective: add the new calculation for the interest fee  
CIType: Document  
Status: Active

Submit

รูปที่ 5.10 หน้าจอส่วนของการกรอกข้อมูลเอกสารความต้องการของผู้ใช้งาน

OntologyCM - Mozilla Firefox

localhost:8080/OntoCMS/cinew.jsp?MainCIType=Document&txtChangeNo=Change-20180704000252&txtProp

### Add New RequirementItem

CI DataType Property

ID: useEffectiveInterestRateFromMLR

Name: useEffectiveInterestRateFromMLR

Version: 1.0

Description: The loan system can use the MLR rate with subtraction or addition by using percentage

businessRuleOrCondition: refer to the bank's standard MLR from setEffectiveRate

contributor: Bantoon L.

postRequire: nent period and how many times to finish the installment ,provide total interest amount

preRequire: Get MLR rate from bank's standard

requirementType: User Requirement

documentName:

documentVersion: 1.0

objective:

CIType: Document

Status: Active

รูปที่ 5.11 หน้าจอส่วนของการกรอกข้อมูลเอกสารข้อกำหนดความต้องการ

เมื่อทำการกด next step จะแสดงหน้าจอ ดังภาพที่ 5.12 เป็นส่วนของการเพิ่มข้อมูลซอฟต์แวร์อื่นๆ เช่น โปรแกรมโมดูล ระบบปฏิบัติการ ระบบฐานข้อมูล เครือข่ายภายนอกและเครือข่ายภายใน

## Ontology-Based Configuration Management System

Home

### New Project - Add OtherItem

Add OtherItem for Project\_Loan.

Select Type:

ApplicationModule Add New Add Existing

ApplicationModule

OS

DBMS

Middleware

Wan

Lan

Back Next Step

รูปที่ 5.12 หน้าจอส่วนของการเพิ่มข้อมูลซอฟต์แวร์และเครือข่าย

เมื่อทำการกด next step จะเป็นส่วนของการสร้างความสัมพันธ์ระหว่างซีไอดังแสดงในรูปที่ 5.13 ซึ่งแสดงการกำหนดความสัมพันธ์ของซีไอชื่อ TestLinux04 มีความสัมพันธ์ชื่อ resides กับ ซีไอชื่อ LanDatacenter เป็นต้น เมื่อกำหนดความสัมพันธ์ครบหมดแล้วกดปุ่ม complete จะแสดงภาพรวมของซอฟต์แวร์ที่ให้บริการทางธุรกิจ Loan ดังรูปที่ 5.14 โดยแสดงเป็นรูปแบบเส้นกราฟลากความสัมพันธ์ของแต่ละซีไอแต่ทั้งนี้เส้นกราฟมีข้อจำกัดที่แสดงความสัมพันธ์ได้เพียงหนึ่งความสัมพันธ์จากบนลงล่างเท่านั้น

## Ontology-Based Configuration Management System

[Home](#)

### New Project - Add Relation

Business System : Project\_Loan.

Select Object for add relation

Server:TestLinux04

Server:ProductionLinux1

TestPlan:TESTLoanAPlan

URD:URDLoanA

RequirementItem:useEffectiveInterestRateFromM

SRS:SRSLoanA

RequirementItem:serverSpecification

RequirementItem:setEffectvieRate

TestCase:Testcase1

TestItem:Case1checkRateOutPeriod

TestItem:Case1checkDate

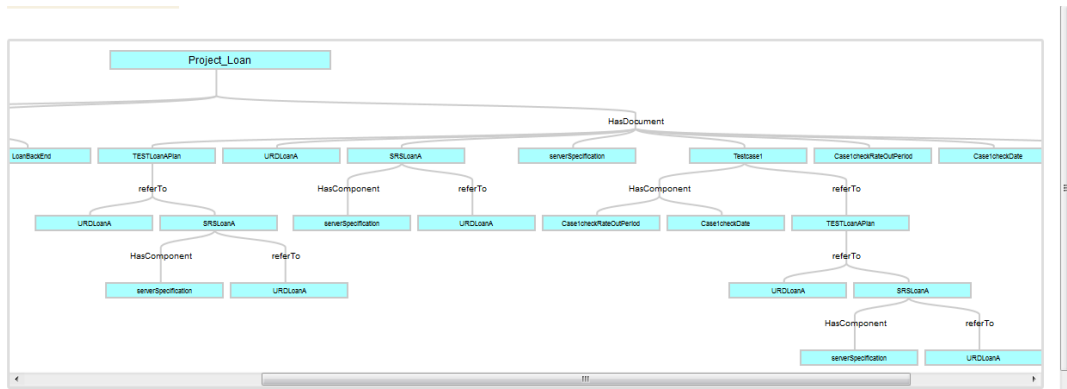
DesignSpec:setEffectiveRate

OS:LinuxRedhat7

DBMS:Oracle9i

Middleware:ApacheTomcat8

รูปที่ 5.13 หน้าจอส่วนของการกำหนดความสัมพันธ์ของซีไอ



รูปที่ 5.14 หน้าจอหลังจากสร้าง Business System สำเร็จ

### 5.2.3 หน้า View Existing Business System

เมื่อทำการเพิ่มข้อมูล Business System ใหม่เสร็จแล้วก็จะเข้ามาตรวจสอบ Business System ที่เพิ่งสร้างมาโดยเข้าหน้าจอ View Existing Business System หน้าจอจะลิสต์ Business System ทั้งหมดในระบบจัดการโครงแบบมาให้เลือก ดังแสดงในรูปที่ 5.15

## Ontology-Based Configuration Management System



### Business System

[Project InternetBanking](#)  
[Project ATM](#)  
[Project MobileBanking](#)  
[obiBusinessSystem](#)  
[Project Loan](#)

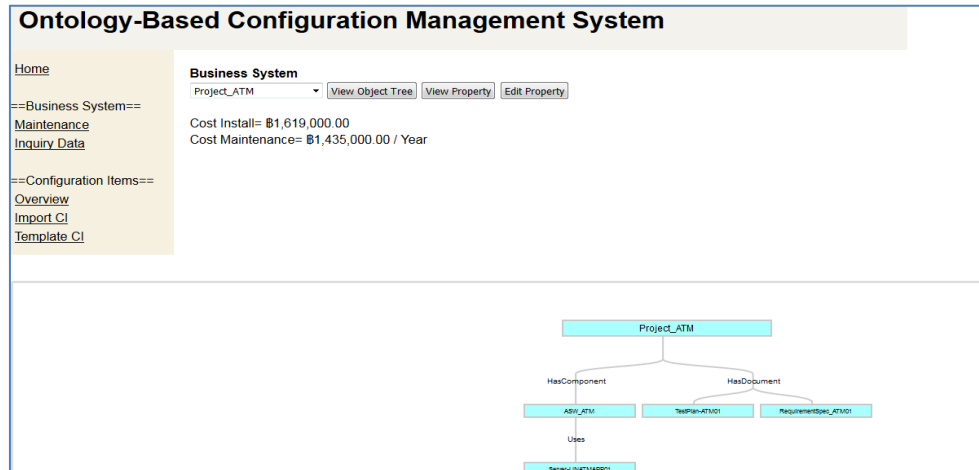
รูปที่ 5.15 หน้าจอหน้า View Existing Business System

### 5.2.4 หน้า Maintenance

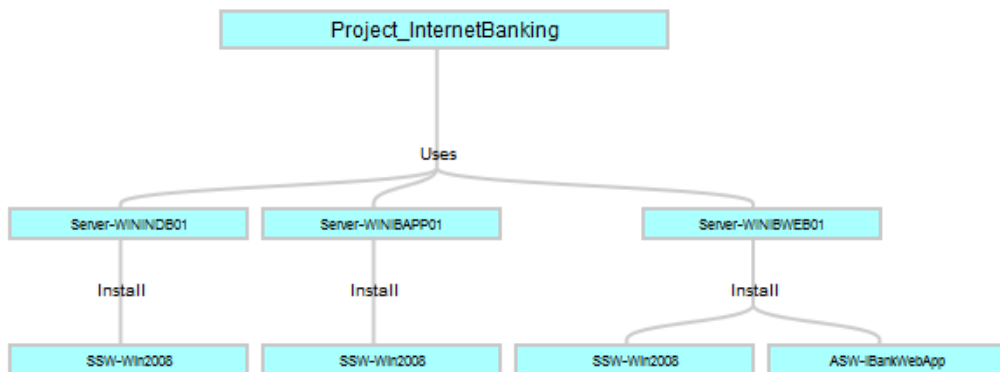
หลังจากเลือก Business System แล้วจะเข้ามายังหน้าจอ Maintenance ซึ่งเป็นหน้าแสดงข้อมูลและมีส่วนจัดการ Business System ที่ได้เลือกไว้แล้ว ตามรูปที่ 5.16 ระบบจะแสดง



ข้อมูลของ Business System และยังแสดงกราฟที่มีข้อมูลองค์ประกอบต่างๆ ของ Business System ดังแสดงตามรูปที่ 5.17



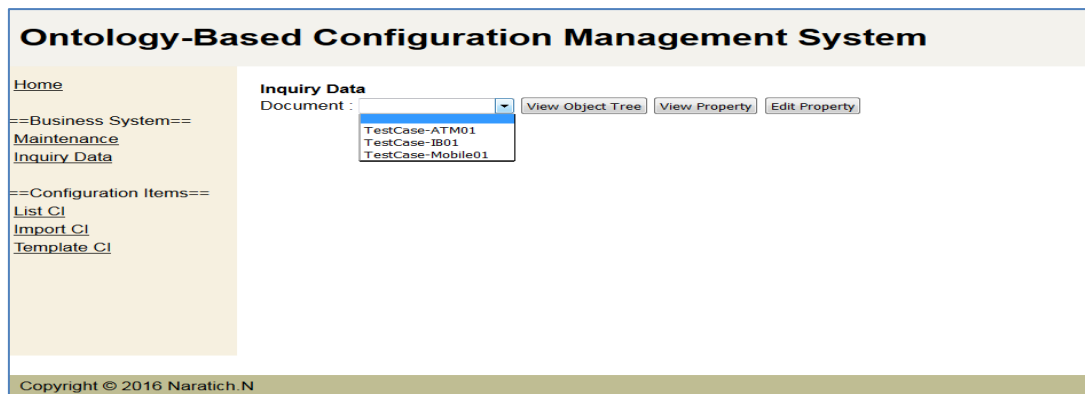
รูปที่ 5.16 หน้าจอแสดงข้อมูลของ Business System



รูปที่ 5.17 ตัวอย่างกราฟที่แสดงข้อมูลองค์ประกอบต่างๆ ของ Business System

### 5.2.5 หน้า Inquiry Data

Inquiry Data เป็นหน้าที่ใช้ออกรายงานชิ้นงานที่อยู่ใน Business System ดังแสดงในรูปที่ 5.18 และตัวอย่างแฟ้มเอกซ์เซลที่ส่งออกในรูปที่ 5.19



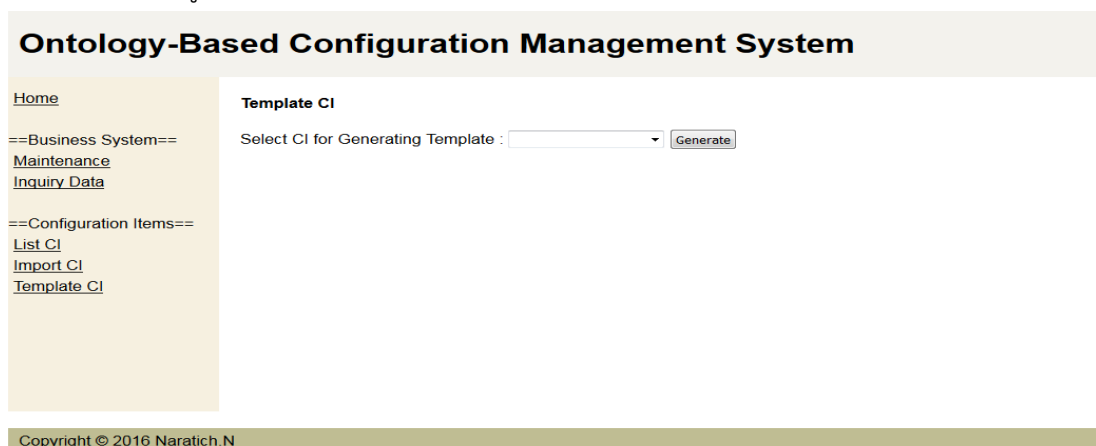
รูปที่ 5.18 ตัวอย่างหน้า Inquiry Data

| ID | Name       | Version    | Description | PersonSupport | CostInstall | CostMaintenance |
|----|------------|------------|-------------|---------------|-------------|-----------------|
| 2  | WINIBAPP03 | WINIBAPP01 | 1.1         |               | 120000.0    | 20000.0         |
| 3  |            |            |             |               |             |                 |
| 4  |            |            |             |               |             |                 |

รูปที่ 5.19 ตัวอย่างการส่งออกข้อมูลซีไอเป็นแฟ้มเอกซ์เซล

### 5.2.6 หน้าจอ Template CI

ก่อนจะทำการนำเข้าซีไอจะเข้าหน้าจอนี้เพื่อสร้างโครงร่างสำหรับนำเข้าข้อมูลซีไอ วิธีการคือเลือกประเภทซีไอและกดปุ่ม generate แสดงดังรูปที่ 5.20 เมื่อลองสร้างแฟ้มต้นแบบเอกซ์เซลซีไอ Server จะได้ดังรูปที่ 5.21



รูปที่ 5.20 หน้าจอ Template CI

|   | A  | B    | C       | D           | E        | F          | G             | H      | I   | J           | K               |
|---|----|------|---------|-------------|----------|------------|---------------|--------|-----|-------------|-----------------|
| 1 | ID | Name | Version | Description | Location | CreateDate | LicenseDetail | Status | SLA | CostInstall | CostMaintenance |
| 2 |    |      |         |             |          |            |               |        |     |             |                 |
| 3 |    |      |         |             |          |            |               |        |     |             |                 |
| 4 |    |      |         |             |          |            |               |        |     |             |                 |
| 5 |    |      |         |             |          |            |               |        |     |             |                 |

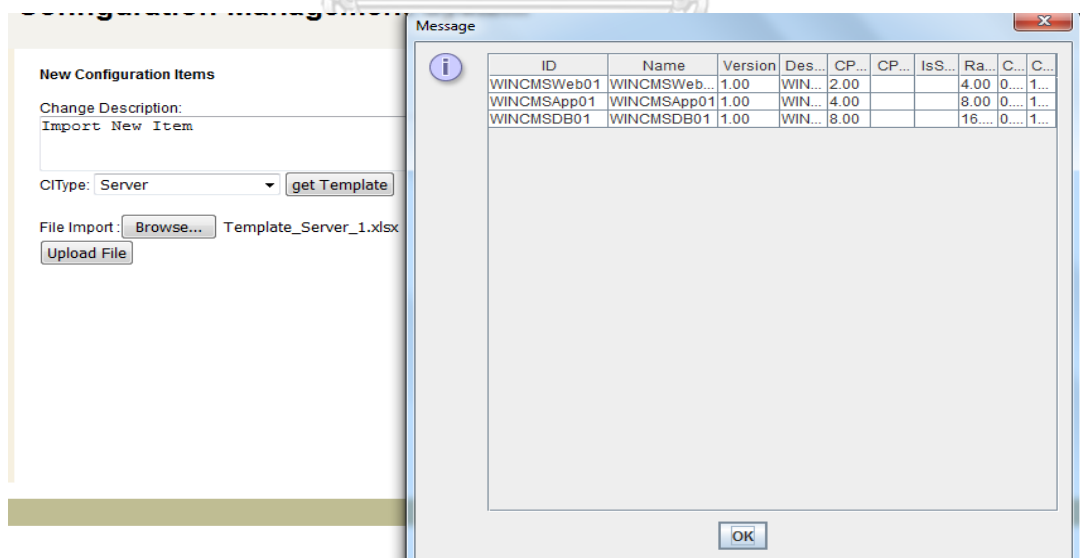
รูปที่ 5.21 ตัวอย่างเพิ่มต้นแบบเอกซ์เซลซีไอ Server

### 5.2.7 หน้าจอ Import CI

หน้าจอ Import CI จะมีฟังก์ชันการใช้งานเพื่อนำเข้าข้อมูลซีไอจากแฟ้มเอกซ์เซลซึ่งสามารถนำเข้าข้อมูลจากแฟ้มเอกซ์เซล ดังแสดงในรูปที่ 5.22 ทำการนำเข้าแฟ้มเอกซ์เซลซีไอ เมื่อเสร็จแล้วจะแสดงผลดังรูปที่ 5.23

|   | A           | B           | C       | D           | E        | F          | G             | H      | I   | J           | K               |
|---|-------------|-------------|---------|-------------|----------|------------|---------------|--------|-----|-------------|-----------------|
| 1 | ID          | Name        | Version | Description | Location | CreateDate | LicenseDetail | Status | SLA | CostInstall | CostMaintenance |
| 2 | WINCMSWeb01 | WINCMSWeb01 | 1.0     | WINCMSWeb01 |          | 11/20/2016 |               | Active |     | 0           | 10000           |
| 3 | WINCMSApp01 | WINCMSApp01 | 1.0     | WINCMSApp01 |          | 11/20/2016 |               | Active |     | 0           | 10000           |
| 4 | WINCMSDB01  | WINCMSDB01  | 1.0     | WINCMSDB01  |          | 11/20/2016 |               | Active |     | 0           | 10000           |
| 5 |             |             |         |             |          |            |               |        |     |             |                 |

รูปที่ 5.22 ตัวอย่างเพิ่มเอกซ์เซลซีไอ Server ที่กรอกข้อมูลแล้ว



รูปที่ 5.23 หน้าจอแสดงผลการใช้งานฟังก์ชัน Import CI

### 5.2.8 หน้าจอ List CI

เมื่อเราเพิ่มหรือแก้ไขซีไอแล้วเราสามารถดูข้อมูลซีไอทั้งหมดของ Business System ที่เราเลือกไว้ได้ในหน้านี้ จากหน้าจอในรูปที่ 5.24 จะแสดงซีไอแต่ละประเภทและมีลิงค์ให้ สร้างซีไอ นำเข้าซีไอ และ ค้นหาซีไอ

## Ontology-Based Configuration Management System

[Home](#)

==Business System==

[Maintenance](#)


[Inquiry Data](#)

==Configuration Items==

[List CI](#)

[Import CI](#)

[Template CI](#)



### Configuration Items

|                          |   |
|--------------------------|---|
| Application_Module (0)   | <a href="#">New</a> <a href="#">Import</a> <a href="#">Search</a> |
| Application_Software (4) | <a href="#">New</a> <a href="#">Import</a> <a href="#">Search</a> |
| ChangeRequest (0)        | <a href="#">New</a> <a href="#">Import</a> <a href="#">Search</a> |
| DBMS (3)                 | <a href="#">New</a> <a href="#">Import</a> <a href="#">Search</a> |
| Firewall (2)             | <a href="#">New</a> <a href="#">Import</a> <a href="#">Search</a> |

รูปที่ 5.24 หน้าจอ List CI

## บทที่ 6

### สรุปผลงานวิจัยและข้อเสนอแนะ

#### 6.1 สรุปผลงานวิจัย

การจัดเก็บชิ้นส่วนโครงแบบของทรัพยากรการบริการไอทีที่จะช่วยให้การจัดการบริการเทคโนโลยีสารสนเทศมีประสิทธิภาพมากขึ้นโดยเฉพาะอย่างยิ่งถ้านำมาใช้กับการจัดการโครงแบบโดยการจัดการโครงแบบคือการสร้างและบำรุงรักษาความเข้ากันได้ของชิ้นงานในระบบโดยมีกระบวนการที่ช่วยจัดเก็บและจัดการข้อมูลซีไอให้เป็นปัจจุบันอยู่เสมอและมีการควบคุมเวอร์ชันการเปลี่ยนแปลงรวมทั้งตรวจสอบความถูกต้องโดยมีฐานข้อมูลการจัดการโครงแบบเป็นฐานข้อมูลหลักของการจัดการด้านเทคโนโลยีสารสนเทศซึ่งเก็บรวบรวมรายละเอียดขององค์ประกอบขององค์กรที่ใช้ในการจัดการบริการเทคโนโลยีสารสนเทศและการสนับสนุนการบริการ ผู้วิจัยได้เสนอแนวคิดที่จะการจัดเก็บชิ้นส่วนโครงแบบของทรัพยากรการบริการไอทีในรูปแบบของข้อมูลออนโทโลยี

ขั้นตอนการพัฒนาเริ่มจากการใช้โครงสร้างแบบแยกย่อยช่วยในการระบุซีไอและเมื่อระบุซีไอได้แล้วนำซีไอที่ได้ไปทำการการออกแบบแผนภาพคลาสของซีไอ โดยทำการออกแบบคลาส คุณสมบัติของคลาส และ ความสัมพันธ์ระหว่างคลาส จากนั้นใช้แผนภาพคลาสของซีไอที่ได้มาเป็นต้นแบบในการออกแบบโครงร่างออนโทโลยีโดยใช้โปรแกรมโปรเทเจ ซึ่งจะได้ผลลัพธ์คือโครงร่างออนโทโลยีของซีไอในรูปแบบของภาษาอาวล์

จากนั้นใช้ข้อมูลของไอทีลเพื่อออกแบบคุณสมบัติชนิดข้อมูล เช่น ชื่อ รายละเอียดซีไอ เวอร์ชัน ค่าใช้จ่าย ค่าบำรุงรักษา สถานะ เป็นต้น และคุณสมบัติแบบวัตถุของซีไอ และยังมีการควบคุมเวอร์ชันของซีไอโดยเก็บข้อมูลการเปลี่ยนแปลงและข้อมูลเดิมไว้

ออนโทโลยีที่ออกแบบสำหรับระบบจัดการโครงแบบมีจำนวนคลาส 26 คลาส คือ ITInfrastructure, Hardware, Software, Network, Document, Server, Firewall, Router, Switch, Wan, Lan, BusinessSystem, Application Software, Application Module, OS, DBMS, Middleware, Software Package, Test Plan, TestCase, Test Item, SRS, Requirement Item, URD, Design Specification และ ChangeRequest จากนั้นใช้โปรแกรมโปรเทเจในการสร้างโครงร่างข้อมูลออนโทโลยีและสร้างข้อมูลโดยได้ข้อมูลจำนวน 1,225 โหนด จากนั้นนำชุดข้อมูลนำเข้าบนอาปาเซจินาฟูเซกิ

อาปาเซจินาได้มีบริการส่วนต่อประสานโปรแกรมประยุกต์ให้สามารถสืบค้นชุดข้อมูลออนโทโลยีได้ ผู้พัฒนาได้พัฒนาเว็บแอปพลิเคชันด้วยภาษาจาวาทำการออกแบบคลาสเพื่อสืบค้นกับ

ฐานข้อมูลออนไลน์โดยใช้นาฟุเซกิ โดยที่โปรแกรมการจัดการโครงแบบ สามารถสร้างซีไอ, แก้ไขซีไอ, ลบซีไอ และสืบค้นข้อมูลซีไอได้

ในการสืบค้นและใช้งานระบบของกรณีศึกษาสามารถเห็นภาพรวมของระบบซึ่งมีประโยชน์ในการใช้งานวิเคราะห์ซึ่งเป็นไปตามผลลัพธ์ที่คาดหวังไว้

## 6.2 ประโยชน์ที่ได้รับ

- 1) ช่วยให้การจัดการบริการเทคโนโลยีสารสนเทศในองค์กรมีทางเลือกที่สร้างระบบโดยใช้ออนไลน์แทนการจัดเก็บแบบเดิมคือ RDBMS อย่างเดียว
- 2) เพิ่มประโยชน์ในการนำข้อมูลไปวิเคราะห์ต่อไปได้ เช่น แก้ไขปัญหาในระบบสารสนเทศหรือทำนายผลกระทบที่จะเกิดขึ้นในอนาคตได้

## 6.3 ข้อจำกัดของระบบ

1. ระบบยังมีส่วนของหน้าที่ (Role) แค่ Admin และ User
2. ระบบยังมีสเกลที่เล็กไม่ครอบคลุมคุณสมบัติทั้งหมดแต่รองรับที่เพิ่มเพิ่มเติมได้

## 6.4 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

1. เพิ่มให้ระบบมีการบูรณาการกับ การจัดการอินซิเด็นท์ (Incident Management) โดยออกแบบระบบให้รองรับกระบวนการจัดการอินซิเด็นท์และออกแบบซีไอรายละเอียดอินซิเด็นท์ที่สามารถโยงความสัมพันธ์ไปยังซีไออื่นๆ ได้
2. เพิ่มซีไอให้ระบบรองรับกับเอกสารต่างๆ ได้มากขึ้น
3. พัฒนาให้ระบบรองรับกับการนำเข้าข้อมูลจากระบบอื่นเข้าระบบได้ได้ง่าย
4. พัฒนาระบบจัดเก็บโครงแบบอัตโนมัติเพื่อรวบรวมขึ้นโครงแบบจากงานปัจจุบันที่ใช้เข้าฐานข้อมูล

## รายการอ้างอิง

1. OGC, *ITIL Core Books*. 2007, UK: TSO.
2. Ward, C., Aggarwal, V. ,Buco, M.,Olsson, E. ,Weinberger S.,IBM Research Division,Thomas J. Watson Research Center, *Integrated Change and Configuration Management*. IBM Systems Journal 2007. **46**(3).
3. Bruegge, B., Dutoit, A.H. , *Object-Oriented Software Engineering Using UML, Patterns, and Java™*. 2010.
4. OGC, *ITIL - Service Transition*. 2007, London: TSO (The Stationery Office).
5. OGC, *ITIL Service Support*. 2000.
6. Pressman, R.S., *Software engineering: a practitioner's approach* 5th ed. 2001.
7. Liu, L., Özsu M.T., *Ontology, Encyclopedia of Database Systems*. Vol. 1. Springer.
8. *Web Ontology Language (OWL)*. 2012-12-11; Available from: <https://www.w3.org/2001/sw/wiki/OWL>.
9. Martin, D. *OWL-S: Semantic Markup for Web Services* Available from: <https://www.w3.org/Submission/OWL-S/>.
10. Horridge, M., *Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools* University of Manchester.
11. Prud'hommeaux, E., Seaborne, A. *SPARQL Query Language for RDF*. Available from: <https://www.w3.org/TR/rdf-sparql-query/>.
12. IEEE, *IEEE Standard for Software Test Documentation*. 1998(IEEE Std 829-1998).
13. Kiranandana, S., *Statistical information technology : data in information systems*. 1998.
14. Dean, T., *Network+ Guide to Networks*. Sixth Edition ed. 2013.
15. Stair, R., Reynolds, G., *Principles of Information Systems,A Managerial Approach*. Ninth Edition ed. 2010.
16. Lheureux, B., Schulte, R. , Natis, Y., McCoy, D., Gassman, B., Sinur, J., Thompson, J. , Pezzini, M. , Kenney, F., Friedman, T. , Gilbert,M. , Phifer, G. .

- Who's Who in Middleware, Gartner RAS Core Strategic Analysis Report R-22-2153*. 29 March 2004; Available from: <https://www-01.ibm.com/software/info/websphere/partners4/articles/gartner/garwho.html>.
17. Brenner, M., Gillmeister, M., *Designing CMDB data models with good utility and limited complexity* in *International Symposium on Network Operations and Management Symposium (NOMS)*. 2014, IEEE: Krakow p. 1-15.
  18. Baioco, G., Costa, A., Calvi, C.Z., Garcia, A.S. *IT service management and governance modeling an ITSM Configuration process: A foundational ontology approach*. 2009. IEEE
  19. Lerlertvanich, R., Vatanawood, W., *Facade Layer for Apache Jena*. *ARPN Journal of Systems and Software*, 2012. **2**(11): p. 350-353.
  20. Aissam, B., Mustapha, B., *Automatic Generation of OWL Ontologies from UML Class Diagrams Based on Meta-Modelling and Graph Grammars*. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2014. **6**(8).





ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

## ประวัติผู้เขียนวิทยานิพนธ์

นายนราทิช ฌ ลำปาง เกิดวันที่ 25 กุมภาพันธ์ พ.ศ. 2523 ที่จังหวัดแพร่ สำเร็จ การศึกษาระดับปริญญาตรีหลักสูตรวิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2544 เข้าการศึกษาต่อ ระดับปริญญาโทบัณฑิต ปีการศึกษา 2557 หลักสูตรวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขา วิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์ มหาวิทยาลัย

ประสบการณ์ทำงาน บริษัทไอพีเอ็มโซลูชันส์ดีเวลอปี่ ปัจจุบันทำงานบริษัทกสิกรซอฟต์แวร์ ในเครือกสิกรบิสซิเนส-เทคโนโลยีกรุ๊ป ตำแหน่งวิศวกรซอฟต์แวร์

