

การออกแบบและพัฒนาส่วนขยายยูติไลไอเพื่อสนับสนุนการโฆษณา  
และการค้นหาบริการโดยใช้คุณสมบัติของบริการ



นาย นิปรกรณ์ชัย ทุมประสิทธิ์

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

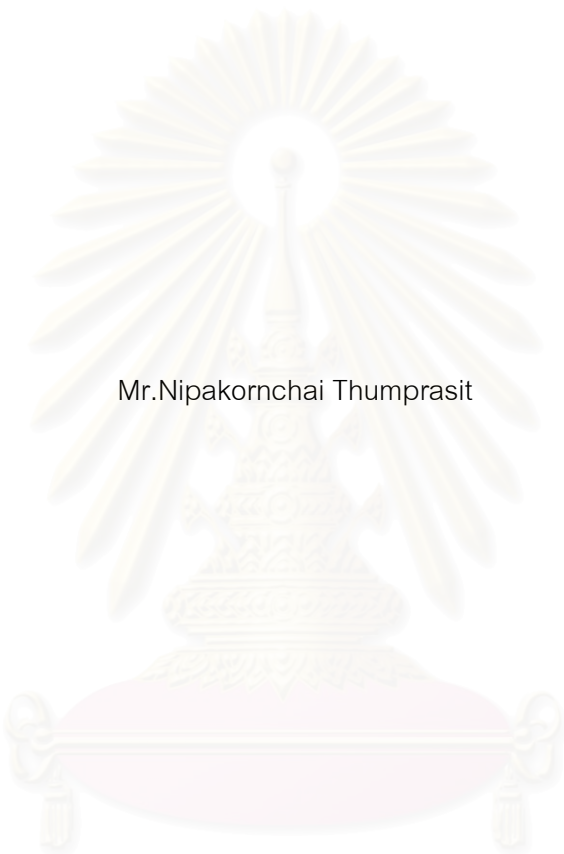
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2546

ISBN 974-17-5547-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THE DESIGN AND DEVELOPMENT OF UDDI EXTENSION TO SUPPORT  
SERVICE ADVERTISING AND SEARCHING BY SERVICE PROPERTIES



Mr.Nipakornchai Thumprasit

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2003

ISBN 974-17-5547-3

หัวข้อวิทยานิพนธ์ การออกแบบและพัฒนาส่วนขยายยูดีดีไอเพื่อสนับสนุนการโฆษณาและการ  
ค้นหาบริการโดยใช้คุณสมบัติของบริการ  
โดย นาย นิปกรณชัย ทุมประสิทธิ์  
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์  
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้  
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร.ดิเรก ลาวัญย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ  
(อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... กรรมการ  
(อาจารย์ ดร.วีระ เหมืองสิน)

นิพนธ์ชัย ทুমประสิทธิ์ : การออกแบบและพัฒนาส่วนขยายยูดีดีไอเพื่อสนับสนุนการโฆษณาและการค้นหาบริการโดยใช้คุณสมบัติของบริการ. (THE DESIGN AND DEVELOPMENT OF UDDI EXTENSION TO SUPPORT SERVICE ADVERTISING AND SEARCHING BY SERVICE PROPERTIES) อ.ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ทวิศย์ เสนีวงศ์ ณ อยุธยา, 97 หน้า. ISBN 974-17-5547-3.

การแข่งขันการให้บริการต่างๆ ในปัจจุบันมีแนวโน้มสูงขึ้น ทำให้องค์กรธุรกิจต้องทำการโฆษณาบริการของตนทางอินเทอร์เน็ตเพื่อให้ผู้บริโภคทราบบริการของตนได้ จึงเกิดการรวมตัวกันจัดตั้งเป็นคลังข้อมูลของการโฆษณาบริการของเว็บเซอร์วิส กำหนดวิธีการโฆษณาและค้นหาบริการร่วมกัน เรียกว่า ยูดีดีไอ แต่ปัญหาที่พบก็คือ ยูดีดีไอกำหนดชุดของคุณสมบัติสำหรับการโฆษณาบริการใดๆ ไว้ ซึ่งชุดคุณสมบัติดังกล่าวมีความจำกัด คือไม่สามารถโฆษณาคุณสมบัติอื่นนอกเหนือจากที่มีกำหนดไว้แล้วได้ นอกจากนี้การค้นหาบริการจากยูดีดีไอนั้นยังเป็นการค้นหาจากชื่อบริการเป็นหลัก เมื่อมีการโฆษณาบริการจำนวนมากขึ้น การค้นหาจากชื่อบริการอย่างเดียวอาจจะได้ข้อมูลจำนวนมาก ซึ่งมีทั้งที่ตรงความต้องการและไม่ตรงความต้องการ จึงทำให้ต้องเสียเวลาในการมาคัดกรองข้อมูลเพื่อให้ได้บริการที่ตนต้องการ เมื่อเป็นเช่นนี้น่าจะมีการระบุคุณสมบัติของบริการเพิ่มเติมได้ เพื่อเป็นการให้ข้อมูลเพิ่มเพื่อการแสดงความแตกต่างกันของแต่ละบริการได้ดียิ่งขึ้น รวมทั้งการค้นหาบริการน่าจะสามารถใช้คุณสมบัติที่เพิ่มเติมขึ้นเป็นเงื่อนไขในการค้นหาได้

งานวิจัยนี้ได้ทำการออกแบบและพัฒนาส่วนขยายยูดีดีไอ เพื่อให้รองรับการโฆษณาและการค้นหาบริการโดยการใช้คุณสมบัติบริการเป็นเงื่อนไข ซึ่งเป็นการเพิ่มคุณลักษณะเฉพาะให้กับบริการที่จะทำการโฆษณาในยูดีดีไอ ด้วยการปรับเปลี่ยนโครงสร้างข้อมูลเกี่ยวกับบริการของยูดีดีไอ โดยการเพิ่มโครงสร้างข้อมูลเกี่ยวกับคุณสมบัติของบริการเข้าไปในโครงสร้างข้อมูลบริการเดิม และเพิ่มส่วนต่อประสานโปรแกรมเพื่อให้สามารถระบุข้อมูลลงในส่วนที่เพิ่มขยายนี้ รวมทั้งนำข้อมูลดังกล่าวมาใช้ในการค้นหาบริการโดยการใช้คุณสมบัติบริการที่เพิ่มเติมเป็นเงื่อนไขในการค้นหาด้วย ซึ่งจะทำให้การกรองข้อมูลทำได้ละเอียดขึ้น

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์.....

สาขาวิชา ..... วิทยาศาสตร์คอมพิวเตอร์.....

ปีการศึกษา ..... 2546.....

ลายมือชื่อนิสิต .....

ลายมือชื่ออาจารย์ที่ปรึกษา .....

## 4371437721 : MAJOR COMPUTER SCIENCE

KEY WORD : UDDI / SERVICE PROPERTY / WEB SERVICES

NIPAKORNCHAI THUMPRASIT: THE DESIGN AND DEVELOPMENT OF UDDI EXTENSION TO SUPPORT SERVICE ADVERTISING AND SEARCHING BY SERVICE PROPERTIES. THESIS ADVISOR: ASSIST. PROF DR. TWITTIE SENIVONGSE, 97 pp. ISBN 974-17-5547-3.

With high competition rate, businesses will advertise their services on the Internet to attract service consumers. This leads to a setup of the standard service registry called UDDI which is a place for advertising and querying for Web Services. UDDI has restrictions in that UDDI does not allow advertisement of additional service properties apart from the set of properties defined in the standard. Moreover, UDDI allows only a simple query such as query by service name. This may result in inaccurate search, especially when there are a great number of service providers, and service consumers will have a burden to differentiate relevant services by themselves. In this situation, UDDI should be extended to allow publishing of additional properties that can give more details to differentiate different services and can be used to define constraints in service query.

This research designs and develops an extension to UDDI so that it can accommodate advertising and querying with additional service properties. To achieve this, UDDI information structure is extended to support more properties, and program interfaces are added for defining these properties or using them as constraints in the queries. These additional properties will provide better filtering at query time.

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department .....Computer Engineering..... Student's signature .....

Field of study .....Computer Science..... Advisor's signature .....

Academic year .....2003.....

## กิตติกรรมประกาศ

งานวิจัยและวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องมาจากการให้คำปรึกษา คำแนะนำ และความช่วยเหลือจาก ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้ข้อคิดเห็นและชี้แนะงานวิจัยนี้เป็นอย่างดี

ขอขอบคุณ อาจารย์ ดร.ยรรยง เต็งอำนวยการ อาจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ และ อาจารย์ ดร.วีระ เหมืองสิน กรรมการวิทยานิพนธ์ ที่กรุณาให้คำแนะนำและตรวจสอบแก้ไขต้นฉบับวิทยานิพนธ์นี้

ขอขอบคุณ เพื่อนๆ ร่วมรุ่น พี่ๆ และน้องๆ ทั้งในและนอกสถาบัน ที่ให้ความช่วยเหลือ คำแนะนำดีๆ ทั้งในงานวิจัยและอื่นๆ รวมทั้งกำลังใจที่มีมาไม่ขาดสาย

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา มารดา น้อง ที่ให้การสนับสนุนในด้านต่างๆ ให้กำลังใจและเข้าใจแก่ผู้วิจัยเป็นอย่างดีเสมอมา



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ .....	ช
สารบัญตาราง .....	ฌ
สารบัญรูปภาพ .....	ญ
บทที่	
1. บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของงานวิจัย .....	2
1.3 ขอบเขตของงานวิจัย .....	2
1.4 ขั้นตอนและวิธีดำเนินงานวิจัย .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	3
1.6 โครงสร้างวิทยานิพนธ์ .....	3
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	4
2.1 ทฤษฎีที่เกี่ยวข้อง .....	4
2.2 งานวิจัยที่เกี่ยวข้อง .....	7
3. การออกแบบส่วนขยายยูดีดีไอ .....	10
3.1 การออกแบบคุณสมบัติบริการ .....	11
3.2 การปรับเปลี่ยนโครงสร้างข้อมูลของยูดีดีไอ (UDDI Data Structure) .....	16
3.2.1 โครงสร้างของบิสเนสเซอร์วิส (businessService) .....	18
3.2.2 โครงสร้างของถุงคุณสมบัติบริการ (servicePropertyBag) .....	19
3.3 การขยายข้อกำหนดของส่วนต่อประสานโปรแกรมยูดีดีไอ (UDDI API Specification) ..	29
3.3.1 การขยายส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา (Inquiry API Function) .....	29
3.3.2 การขยายส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณา (Publishing API Function) .....	42
4. ต้นแบบส่วนขยายยูดีดีไอ .....	48

บทที่	หน้า
4.1 คลาสหลักต้นแบบส่วนขยายยูติไลตี้	48
4.2 ส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา	50
4.3 ส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณา	53
4.4 ฐานข้อมูลส่วนขยายยูติไลตี้	56
4.5 เครื่องมือในการพัฒนาต้นแบบ	58
5. การทดสอบต้นแบบส่วนขยายยูติไลตี้	59
5.1 สภาวะที่ใช้ในการทดสอบ	60
5.2 การทดสอบการโฆษณาบริการพร้อมคุณสมบัติ	61
5.3 การทดสอบการค้นหาบริการด้วยคุณสมบัติบริการ	68
6. สรุปผลการวิจัยและข้อเสนอแนะ	72
6.1 สรุปผลการวิจัย	72
6.2 ปัญหาและข้อจำกัดของงานวิจัย	73
6.3 ข้อเสนอแนะ	73
รายการอ้างอิง	75
ภาคผนวก	76
ภาคผนวก ก	77
ประวัติผู้เขียนวิทยานิพนธ์	97



## สารบัญตาราง

ณ

ตาราง	หน้า
ตารางที่ 3.1 คุณลักษณะความจำเป็นของข้อมูล (mode) .....	20
ตารางที่ 3.2 ชนิดของลักษณะข้อมูล (valueTypeRef) .....	21
ตารางที่ 3.3 ชนิดของข้อมูล (type) .....	22



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาพประกอบ	หน้า
รูปที่ 2.1 คุณสมบัติพื้นฐานของยูดีดีไอ .....	4
รูปที่ 2.2 โครงสร้างข้อมูลในยูดีดีไอ .....	5
รูปที่ 3.1 การเพิ่มส่วนขยายของยูดีดีไอเพื่อรองรับคุณสมบัติของบริการ (Service Properties)..	10
รูปที่ 3.2 โครงสร้างของบิซิเนสเซอร์วิส ตามข้อกำหนดยูดีดีไอ .....	17
รูปที่ 3.3 โครงสร้างข้อมูลของยูดีดีไอที่เพิ่มคุณสมบัตินบริการ .....	17
รูปที่ 3.4 โครงสร้างของบิซิเนสเซอร์วิสที่เพิ่มคุณสมบัตินบริการ .....	18
รูปที่ 3.5 โครงสร้างของคุณสมบัติบริการ .....	19
รูปที่ 3.6 โครงสร้างของ propertyBag .....	20
รูปที่ 3.7 โครงสร้างของ propertyReference .....	20
รูปที่ 3.8 โครงสร้างของ valueReference .....	21
รูปที่ 3.9 โครงสร้างของ simpleType.....	22
รูปที่ 3.10 โครงสร้างของ structType และ fieldStruct .....	23
รูปที่ 3.11 โครงสร้างของ arrayType .....	24
รูปที่ 3.12 โครงสร้างของ enumType .....	25
รูปที่ 3.13 โครงสร้างของ unionType.....	25
รูปที่ 3.14 โครงสร้างของ propertySet.....	26
รูปที่ 3.15 โครงสร้างของ propertySetBag .....	27
รูปที่ 3.16 โครงสร้างของ propertySetFamily .....	28
รูปที่ 3.17 โครงสร้างของ propertySetParent .....	28
รูปที่ 3.18 โครงสร้างของ propertySetChild .....	29
รูปที่ 3.19 โครงสร้างของ find_property .....	30
รูปที่ 3.20 โครงสร้างของ propertyCollection .....	30
รูปที่ 3.21 โครงสร้างของ propertyList .....	31
รูปที่ 3.22 โครงสร้างของ propertyInfo.....	31
รูปที่ 3.23 โครงสร้างของ find_propertySet.....	32
รูปที่ 3.24 โครงสร้างของ propertySetCollections .....	33
รูปที่ 3.25 โครงสร้างของ propertySetList .....	33
รูปที่ 3.26 โครงสร้างของ propertySetInfo.....	34

## สารบัญญภาพ (ต่อ)

ฎ

ภาพประกอบ	หน้า
รูปที่ 3.27 โครงสร้างของ propertyChild .....	34
รูปที่ 3.28 โครงสร้างของ find_serviceProperty .....	35
รูปที่ 3.29 โครงสร้างของ propertyValueInfo .....	36
รูปที่ 3.30 โครงสร้างของ propertySetValue .....	36
รูปที่ 3.31 โครงสร้างของ servicePropertyList .....	37
รูปที่ 3.32 โครงสร้างของ servicePropertyInfo .....	37
รูปที่ 3.33 โครงสร้างของ servicePropertySet .....	38
รูปที่ 3.34 โครงสร้างของ get_propertyDetail .....	39
รูปที่ 3.35 โครงสร้างของ propertyDetail .....	40
รูปที่ 3.36 โครงสร้างของ get_propertySetDetail .....	40
รูปที่ 3.37 โครงสร้างของ propertySetDetail .....	41
รูปที่ 3.38 โครงสร้างของ get_servicePropertyDetail .....	41
รูปที่ 3.39 โครงสร้างของ servicePropertyDetail .....	42
รูปที่ 3.40 โครงสร้างของ save_property .....	43
รูปที่ 3.41 โครงสร้างของ save_propertySet .....	43
รูปที่ 3.42 โครงสร้างของ save_serviceProperty .....	44
รูปที่ 3.43 โครงสร้างของ delete_property .....	44
รูปที่ 3.44 โครงสร้างของ delete_propertySet .....	45
รูปที่ 3.45 โครงสร้างของ delete_serviceProperty .....	45
รูปที่ 3.46 โครงสร้างของ businessServiceKey .....	46
รูปที่ 3.47 โครงสร้างของ propertySetBagKey .....	46
รูปที่ 4.1 โครงสร้างข้อมูลของบริการในยูดีดีไอที่แสดงคุณสมบัติและเซตของคุณสมบัติ .....	48
รูปที่ 4.2 คลาสหลักต้นแบบส่วนขยายยูดีดีไอ .....	49
รูปที่ 4.3 ส่วนข้อมูลส่วนขยายยูดีดีไอ .....	57
รูปที่ 5.1 ส่วนต่อประสานโปรแกรมของส่วนขยายยูดีดีไอ .....	59
รูปที่ 5.2 การเพิ่มข้อมูลบริการตามข้อกำหนดยูดีดีไอ .....	62
รูปที่ 5.3 ข้อมูลบริการเชิงโครงสร้างตามข้อกำหนดยูดีดีไอ .....	62
รูปที่ 5.4 การเพิ่มข้อมูลคุณสมบัติบริการตามส่วนขยายยูดีดีไอ .....	63

## สารบัญญภาพ (ต่อ)

	ฎ หน้า
ภาพประกอบ	
รูปที่ 5.5 การเพิ่มเซตของคุณสมบัติ room price .....	64
รูปที่ 5.6 การเพิ่มคุณสมบัติเป็นสมาชิกเซต room price.....	64
รูปที่ 5.7 การค้นหาเซตของคุณสมบัติ room price เพื่อใช้ซ้ำ .....	68
รูปที่ 5.8 การค้นหาข้อมูลบริการตามข้อกำหนดยูดีดีไอเดิม.....	68
รูปที่ 5.9 รายการบริการที่มีชื่อว่า Hotel Reservation .....	69
รูปที่ 5.10 โครงสร้างข้อมูลของการค้นหาตามส่วนขยายยูดีดีไอ (find_serviceProperty) .....	70
รูปที่ 5.11 ข้อมูลที่ได้จากการค้นหาข้อมูลด้วยส่วนขยายยูดีดีไอโดยระบุเงื่อนไข .....	70



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

องค์กรต่าง ๆ ที่มีระบบคอมพิวเตอร์และระบบสารสนเทศของตนเองต่างมีความต้องการที่จะเพิ่มศักยภาพและประสิทธิภาพในการดำเนินธุรกิจ จึงมีการประสานประโยชน์ร่วมกัน โดยการแลกเปลี่ยนข้อมูลข่าวสารและบริการระหว่างกันเพื่อเพิ่มมูลค่าของสินค้าและบริการของตน จากการแลกเปลี่ยนข้อมูลหรือบริการที่อยู่ในวงจำกัดเฉพาะองค์กรที่ตกลงเป็นพันธมิตรกัน ปัจจุบันได้กลายมาเป็นการให้บริการแก่ผู้รับบริการโดยทั่วไปในรูปแบบของ เว็บเซอร์วิส (Web Services) [1]

เว็บเซอร์วิส เป็นสถาปัตยกรรมของการให้บริการบนเวปไซต์ไวด์เว็บ โดยบริการจะถูกพัฒนาในรูปแบบของโปรแกรมที่ถูกเรียกใช้งานผ่านโพรโทคอลเอชทีทีพีบนเครือข่ายอินเทอร์เน็ต โครงสร้างของข้อความ (Message) ที่เป็นการเรียกใช้บริการจะอยู่ในรูปของเอกสารเอ็กซ์เอ็มแอล (XML: Extensible Markup Language) [2] ซึ่งเป็นไปตามข้อกำหนดของโพรโทคอลโซฟ (SOAP: Simple Object Access Protocol) [3] ทำให้การเรียกใช้งานบริการสามารถทำได้บนโพรโทคอลที่แพร่หลายและไม่จำกัดอยู่ที่สถาปัตยกรรมใดโดยเฉพาะเหมือนแต่ก่อน

เมื่อองค์กรต่าง ๆ ได้พัฒนาบริการของตนขึ้นมาเพื่อให้บริการกับองค์กรอื่น ๆ ปัญหาที่ตามมาก็คือองค์กรอื่น ๆ จะทราบได้อย่างไรว่ามีบริการอะไรอยู่ที่ใดบ้าง จากปัญหาที่เกิดขึ้นจึงมีองค์กรด้านธุรกิจและคอมพิวเตอร์ได้มาทำข้อตกลงร่วมกันเพื่อกำหนดลักษณะของการเรียกใช้บริการร่วมกัน โดยเรียกว่า ยูดีดีไอ (UDDI: Universal Description, Discovery, and Integration) [4][5] โดยกำหนดวิธีการโฆษณาบริการซึ่งอยู่บนพื้นฐานของเอกสารเอ็กซ์เอ็มแอล รวมทั้งวิธีการค้นหาบริการ วิธีการเรียกใช้บริการ และการจัดการข้อมูลบริการ โดยสามารถเรียกใช้ยูดีดีไอนี้โดยการผูกมัด (Bind) ไว้กับโปรแกรมประยุกต์ภายในองค์กรที่มีอยู่ได้

ปัญหาหนึ่งที่พบในยูดีดีไอคือ ยูดีดีไอมีโครงสร้างของคำโฆษณาบริการที่ตายตัวและไม่รองรับการระบุคุณสมบัติอื่นใดให้กับบริการนอกเหนือจากที่ได้กำหนดไว้ ทำให้ไม่สามารถค้นหาบริการโดยกำหนดเงื่อนไขตามคุณสมบัติอื่น ๆ นั้นได้ ในการค้นหาจึงอาจได้ผลลัพธ์กลับมาเป็นจำนวนมาก ทำให้ต้องมาทำการคัดเลือกบริการจากผลลัพธ์จำนวนมากนั้นอีกต่อหนึ่ง หากสามารถระบุคุณสมบัติอื่น ๆ ให้กับบริการและค้นหาโดยกำหนดเงื่อนไขตามคุณสมบัติของบริการได้ จะทำให้ผลลัพธ์ที่ได้กลับมาจะมีจำนวนลดลง และเป็นผลลัพธ์ที่ตรงตามความต้องการของผู้รับบริการมากขึ้น งานวิจัยนี้จึงมีแนวคิดที่จะพัฒนาส่วนขยายให้กับยูดีดีไอเพื่อให้รองรับการ

โฆษณาบริการโดยมีการระบุคุณสมบัติต่าง ๆ เพิ่มเติมให้กับบริการ และให้สามารถทำการค้นหาบริการโดยระบุเงื่อนไขตามคุณสมบัติของบริการที่เพิ่มเติมขึ้นมาได้

## 1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาส่วนขยายของยูดีดีไอให้สนับสนุนการโฆษณาคุณสมบัติของบริการได้เพิ่มเติมจากที่กำหนดไว้โดยยูดีดีไอมาตรฐาน รวมทั้งการใช้คุณสมบัติของบริการที่เพิ่มเติมนั้นเป็นเงื่อนไขในการค้นหา

## 1.3 ขอบเขตของการวิจัย

1. รูปแบบคำอธิบายของยูดีดีไอที่จะเป็นพื้นฐานในการพัฒนา จะเป็นไปตามโครงสร้างข้อมูลของยูดีดีไอ รุ่นที่ 1.0 ตาม UDDI Open Draft Specification 30 September 2000 เป็นอย่างน้อย
2. เอกสารเอ็กซ์เอ็มแอลที่ใช้ระบุคำอธิบายบริการจะเป็นไปตามข้อกำหนดของเอ็กซ์เอ็มแอล รุ่นที่ 1.0 ตาม W3C Recommendation 10-Feb-98 เป็นอย่างน้อย
3. ยูดีดีไอ รีจิสทรี ที่ใช้ในงานวิจัยนี้จะเป็นไปตามข้อกำหนดของยูดีดีไอ รุ่นที่ 1.0 เป็นอย่างน้อย

## 1.4 ขั้นตอนและวิธีการดำเนินงานวิจัย

1. ศึกษาทฤษฎีที่เกี่ยวข้องกับเว็บเซอร์วิส เช่น เอ็กซ์เอ็มแอล ดับเบิลยูเอสดีแอล โซพ เป็นต้น
2. ศึกษาทฤษฎีของยูดีดีไอ
3. ศึกษาทดลองเครื่องมือต่างๆ ที่จะนำมาใช้ในการพัฒนา
4. ออกแบบโครงสร้างข้อมูลและส่วนต่อประสานโปรแกรมยูดีดีไอ
5. พัฒนาคู่มือส่วนขยายของยูดีดีไอ รีจิสทรี
6. พัฒนาโปรแกรมที่เรียกใช้ส่วนขยายของยูดีดีไอ รีจิสทรี
7. ทดสอบและปรับปรุงแก้ไขต้นแบบ
8. สรุปผลและข้อเสนอแนะ
9. จัดทำรายงานวิทยานิพนธ์

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ข้อกำหนดสำหรับการกำหนดคุณสมบัติของบริการในยูดีดีไอ เพื่อให้สามารถโฆษณาคุณสมบัติอื่น ๆ และค้นหาบริการโดยระบุคุณสมบัติของบริการเหล่านั้นเป็นเงื่อนไขได้
2. ได้ต้นแบบในการสร้างส่วนต่อประสานโปรแกรม เพื่อใช้ในการโฆษณาและค้นหาบริการโดยระบุคุณสมบัติของบริการเป็นเงื่อนไขในการค้นหาได้
3. สามารถนำข้อมูลคำอธิบายบริการที่อยู่ในระบบอื่น เช่น ในเทรดเดอร์ของคอร์บา มาทำการโฆษณาไว้ที่ยูดีดีไอ ริจิสทรี ที่ได้เพิ่มขยายตามงานวิจัยนี้ได้ อันจะเป็นการสนับสนุนการทำงานข้ามระบบ โดยให้บริการของคอร์บาสามารถถูกค้นพบในลักษณะของเว็บเซอร์วิสได้

### 1.6 โครงสร้างวิทยานิพนธ์

ในบทต่อไปของวิทยานิพนธ์นี้จะกล่าวถึงทฤษฎีที่นำมาประยุกต์ใช้และงานวิจัยที่เกี่ยวข้อง ส่วนในบทที่ 3 จะกล่าวถึงการออกแบบส่วนขยายของยูดีดีไอ ในบทที่ 4 จะกล่าวถึงการพัฒนาระบบส่วนขยายยูดีดีไอ ในบทที่ 5 จะกล่าวถึงการประยุกต์ใช้ส่วนขยายยูดีดีไอ ซึ่งเป็นส่วนขยายที่ได้ในบทที่ 3 และต้นแบบในบทที่ 4 ส่วนในบทสุดท้ายนั้นจะเป็นการสรุปผลของงานวิทยานิพนธ์ และข้อเสนอแนะในการพัฒนาต่อไป

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 2

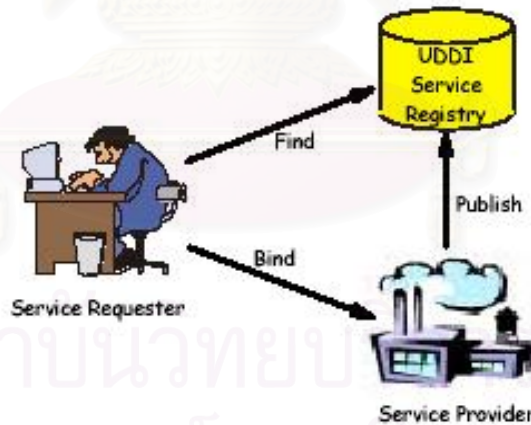
### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 ยูดีดีไอ (Universal Description, Discovery, and Integration (UDDI)) [4][5]

ยูดีดีไอ เป็นข้อกำหนดในการโฆษณาบริการของเว็บเซอร์วิส โดยมียูดีดีไอ รีจิสทรี (UDDI Registry) เป็นคลังที่เก็บข้อมูลการโฆษณาบริการไว้ เพื่อให้การค้นหบริการสามารถทำได้ง่าย และเป็นรูปแบบเดียวกัน โครงสร้างพื้นฐานของการโฆษณาบริการเป็นไปตามรูปแบบของภาษา เอ็กซ์เอ็มแอล ทำให้ไม่ขึ้นกับระบบปฏิบัติการ ภาษาคอมพิวเตอร์ หรือสถาปัตยกรรมคอมพิวเตอร์ ยูดีดีไอรุ่นที่ 1.0 ได้กำหนดคุณสมบัติพื้นฐานไว้ 3 ลักษณะ ดังรูปที่ 2.1 คือ

1. การโฆษณา (Publish) ผู้ให้บริการเว็บเซอร์วิส สามารถทำการโฆษณาบริการของตนตามรูปแบบและรายละเอียดที่กำหนด
2. การค้นหา (Find) ผู้รับบริการ สามารถทำการค้นหบริการที่ต้องการได้
3. การผูกมัด (Bind) ผู้รับบริการ ทำการขอใช้บริการเว็บเซอร์วิสจากผู้ให้บริการ หลังจากที่ได้ค้นหบริการที่ต้องการได้แล้ว



รูปที่ 2.1 คุณสมบัติพื้นฐานของยูดีดีไอ

การโฆษณาบริการตามมาตรฐานของยูดีดีไอ ซึ่งอยู่ในรูปของเอกสารเอ็กซ์เอ็มแอลนั้นจะประกอบด้วยข้อมูล 3 ส่วน คือ

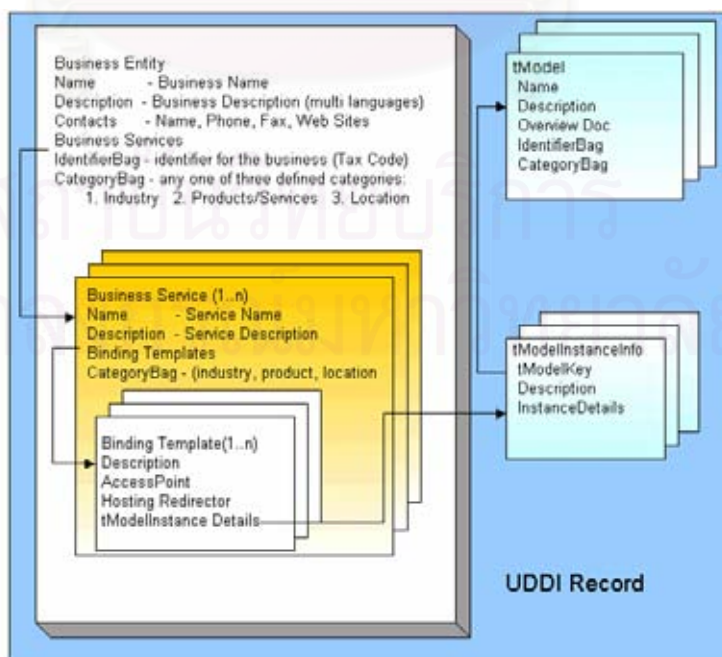
1. สมุดหน้าขาว (White Pages) เป็นส่วนที่ประกอบด้วยข้อมูลขององค์กรที่เป็นเจ้าของบริการที่จะทำการโฆษณา เช่น ชื่อขององค์กร ที่อยู่ หมายเลขโทรศัพท์ ข้อมูลที่เกี่ยวข้องกับธุรกิจขององค์กร เป็นต้น



2. สมุดหน้าเหลือง (Yellow Pages) เป็นส่วนที่ประกอบด้วยข้อมูลของบริการที่องค์กรนั้นให้บริการ และการจัดหมวดหมู่ของธุรกิจ โดยอ้างอิงถึงการจัดหมวดหมู่ที่เป็นมาตรฐาน เช่น NAICS, UNSPSC, ISO 3166 เป็นต้น
3. สมุดหน้าเขียว (Green Pages) เป็นส่วนที่ประกอบด้วยข้อมูลในเชิงเทคนิคของการเรียกใช้บริการเว็บไซต์นั้น ๆ

ข้อมูลที่ประกอบใน 3 ส่วนดังกล่าวข้างต้น จะแสดงโครงสร้างได้ดังรูปที่ 2.2 ซึ่งประกอบด้วย

1. ข้อมูลทางด้านธุรกิจ (Business Information) ได้แก่ บิสิเนสเอ็นทิตี (Business Entity) ซึ่งเป็นข้อมูลในส่วนของสมุดหน้าขาว
2. ข้อมูลการบริการ (Service Information) ได้แก่ บิสิเนสเซอร์วิส (Business Service) เป็นข้อมูลส่วนที่แสดงบริการที่มีอยู่และส่วนที่เกี่ยวข้องกับการจัดหมวดหมู่ของธุรกิจ ซึ่งจะเป็นข้อมูลในส่วนของสมุดหน้าเหลือง
3. ข้อมูลสำหรับการผูกมัด (Binding Information) ได้แก่ ไลนิงเทมเพลต (Binding Template) เป็นข้อมูลรายละเอียดที่สำคัญในการเรียกใช้งานเว็บไซต์ ซึ่งเป็นข้อมูลในส่วนของสมุดหน้าเขียว
4. ข้อมูลเกี่ยวกับข้อกำหนดคุณลักษณะของเว็บไซต์ ได้แก่ ทีโมเดล (tModel) ซึ่งเป็นข้อกำหนดในการเรียกใช้เว็บไซต์



รูปที่ 2.2 โครงสร้างข้อมูลในยูดีดีไอ

การค้นหบริการสามารถเรียกใช้บริการตามมาตรฐานของยูดีดีไอได้ดังนี้

1. find\_Business เป็นการค้นหาโดยระบุข้อมูลเกี่ยวกับธุรกิจ เช่น ชื่อธุรกิจ โดยผลลัพธ์ที่ได้ คือ ข้อมูลเกี่ยวกับธุรกิจนั้น ๆ รวมไปถึงบริการที่ธุรกิจนั้นให้บริการด้วย
2. find\_Service เป็นการค้นหาโดยระบุข้อมูลเกี่ยวกับบริการ เช่น ชื่อบริการ โดยผลลัพธ์ที่ได้คือ ข้อมูลที่แสดงว่ามีธุรกิจใดบ้างที่มีบริการที่ค้นหา
3. find\_tModel เป็นการค้นหาโดยระบุชื่อที่โมเดล

### 2.1.2 ดับเบิลยูเอสดีแอล (Web Service Description Language (WSDL)) [9][10]

ดับเบิลยูเอสดีแอล เป็นภาษาสำหรับการบรรยายเว็บเซอร์วิส โดยมีโครงสร้างพื้นฐานคือ ภาษาเอ็กซ์เอ็มแอล ซึ่งจะบรรยายถึงวิธีการติดต่อเรียกใช้บริการผ่านเครือข่ายเวิลด์ไวด์เว็บ การดำเนินการ (Operation) ที่บริการรองรับ และข้อความที่ใช้ร้องขอบริการ ทั้งนี้เพื่อเป็นการโฆษณาถึงวิธีการเรียกใช้งานบริการที่มีอยู่ โดยรายละเอียดประกอบไปด้วย

1. Type ได้แก่ การกำหนดชนิดของข้อมูล
2. Message ได้แก่ ลักษณะของข้อความที่เกี่ยวข้องกับการเรียกใช้บริการ
3. Operation ได้แก่ การดำเนินการ
4. Port Type ได้แก่ ชนิดของพอร์ตในการติดต่อบริการ โดยเปรียบเสมือนส่วนต่อประสานของบริการ
5. Binding ได้แก่ โพรโทคอลและรูปแบบของข้อมูลที่จะใช้ในการติดต่อบริการ
6. Port ได้แก่ พอร์ตซึ่งกำหนดการผูกมัดตำแหน่งที่อยู่ในเครือข่ายเข้าด้วยกัน
7. Service ได้แก่ บริการ

ผู้ให้บริการสามารถบรรยายบริการตามรูปแบบของดับเบิลยูเอสดีแอลหรือไม่ก็ได้ หากใช้ดับเบิลยูเอสดีแอลก็จะระบุไว้ที่สมุดหน้าเขียวของบริการ โดยหลังจากที่ผู้รับบริการทำการค้นหาบริการจากยูดีดีไอ วิธีสทรีแล้ว เมื่อจะเรียกใช้บริการจะทำการร้องขอไปยังผู้ให้บริการโดยตรง โดยส่งคำร้องขอตามข้อกำหนดในดับเบิลยูเอสดีแอลของบริการนั้น

### 2.1.3 เอ็กซ์เอ็มแอล (Extensible Markup Language (XML)) [2]

เอ็กซ์เอ็มแอล เป็นภาษาที่ถูกออกแบบมาเพื่อใช้สำหรับอธิบายข้อมูล โดยมีพื้นฐานมาจากภาษาเอสจีเอ็มแอล (SGML: Standard Generalized Markup Language) แต่ได้ทำการลดทอนไวยากรณ์ที่มีความซับซ้อนของภาษาเอสจีเอ็มแอล และสร้างข้อกำหนดของไวยากรณ์ที่ง่ายต่อการเขียน เอกสารเอ็กซ์เอ็มแอลจะมีการอธิบายข้อมูลในรูปแบบเป็นลำดับชั้น (Hierarchy) เหมือนต้นไม้ (Tree) โดยจะสามารถสร้างโปรแกรมประยุกต์ เพื่อทำการประมวลผลเอกสารได้โดยไม่ต้องขึ้นกับสถาปัตยกรรมคอมพิวเตอร์ และยังใช้เป็นพื้นฐานในการใช้งานหลายๆ อย่าง เช่น ใช้ในการบรรยายดับเบิลยูเอสดีแอล หรือยูดีดีไอ เป็นต้น

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 บริการเทรดเดอร์ (Trading Object Service) [6]

บริการเทรดเดอร์ เป็นบริการที่ใช้ในการโฆษณาและการค้นหาบริการของระบบคอร์บา (CORBA) ผู้ให้บริการ (Exporter) สามารถมาทำการโฆษณาบริการไว้ที่เทรดเดอร์ และเมื่อวัตถุที่เป็นผู้รับบริการ (Importer) ต้องการเรียกใช้บริการก็จะทำการร้องขอมายังเทรดเดอร์ให้ทำการค้นหาได้

ข้อมูลบริการในเทรดเดอร์ จะระบุ

1. ชนิดของบริการ (Service Type)
2. ส่วนต่อประสานของบริการ (Interface)
3. คุณสมบัติต่างๆ ของบริการ (Properties)

เช่น บริการที่มีชนิด Bank มีส่วนต่อประสานตาม Bank.idl และมีคุณสมบัติ bankname=SCB และ interestrate=0.05 การระบุค่าโฆษณาในลักษณะนี้ทำให้ผู้รับบริการสามารถระบุเงื่อนไขต่างๆ ในการค้นหาได้ เช่น ต้องการบริการใดๆ ที่มีชนิด Bank หรือต้องการบริการชนิด Bank ที่มีอัตราดอกเบี้ยมากกว่า 4% ได้ ทำให้การค้นหาเป็นไปอย่างยืดหยุ่นและผู้รับบริการได้รับบริการที่ตรงตามความต้องการ

บริการเทรดเดอร์กำหนดโครงสร้างในการโฆษณาคุณสมบัติต่างๆ ของบริการในลักษณะของชื่อคุณสมบัติ (Property Name) และชนิดข้อมูลของคุณสมบัติ (Property Type) โดยไม่กำหนดตายตัวว่าบริการควรโฆษณาด้วยคุณสมบัติใดบ้าง ซึ่งตรงกันข้ามกับกรณีของยูดีดีไอ ดังนั้นผู้ให้บริการจึงสามารถนิยามคุณสมบัติใหม่ๆ สำหรับโฆษณาบริการบนเทรดเดอร์ได้ตามที่เห็นเหมาะสม ผู้วิจัยจะได้ใช้แนวทางดังกล่าวของเทรดเดอร์ ในการพัฒนาส่วนขยายให้กับยูดีดีไอ

## 2.2.2 WebTrader: Discovery and Programmed Access to Web-Based Services [7]

แนวคิดในงานวิจัยนี้ ได้นำเสนอการค้นหาค้นหาบริการผ่านเว็ลด์ไวด์เว็บโดยเว็บเทรดเดอร์ (Web Trader) ผู้ให้บริการจะทำการโฆษณาบริการของตนโดยใช้หน้าโฆษณาบริการ (Service Advertisement Page (SAP)) ที่เป็นเอกสารที่อยู่ในรูปแบบของเอ็กซ์เอ็มแอล โดยเอกสารนี้จะประกอบด้วยข้อมูล 3 ส่วน คือ

1. ส่วนต่อประสาน
2. เมตาเดตา (Metadata) ที่เป็นคุณสมบัติ
3. คำสำคัญในการค้นหา (Search Keywords)

ผู้รับบริการสามารถทำการค้นหาค้นหาบริการผ่านทางเว็บเทรดเดอร์ โดยทำการส่งเอกสารความต้องการซึ่งเป็นเอกสารเอ็กซ์เอ็มแอลที่มีส่วนหนึ่งเป็นเอกสารเอ็กซ์เอ็มแอล และมีโครงสร้างตามคำอธิบายชนิดเอกสารที่เรียกว่า ดีทีดี (Document Type Description (DTD)) ซึ่งคล้ายกับการโฆษณาของผู้ให้บริการ จากนั้นเว็บเทรดเดอร์จะทำการเปรียบเทียบและค้นหาค้นหาบริการที่ตรงกับความต้องการที่ร้องขอมา และส่งข้อมูลสำหรับส่วนการติดต่อกับบริการนั้นคืนมาให้ผู้รับบริการ

งานวิจัยชิ้นนี้สามารถที่จะทำการสร้างตัวกลางในการค้นหาค้นหาบริการบนเว็ลด์ไวด์เว็บโดยไม่ขึ้นกับสถาปัตยกรรมระบบกระจายใดๆ โดยการนำเสนอคำอธิบายบริการในรูปแบบของเอกสารเอ็กซ์เอ็มแอลแทนที่คำอธิบายของแต่ละสถาปัตยกรรม เช่น คอร์บา (CORBA) จาวา อาร์เอ็มไอ (Java RMI) หรือ ซีจีไอ-สคริปต์ (CGI-Script) เป็นต้น

## 2.2.3 A Design and Development of an Extension to Jini Lookup Service to Support Non-Exact Match Search [8]

แนวคิดในงานวิจัยนี้ นำเสนอการปรับปรุงให้บริการลูคัฟของระบบจินี มีความสามารถเพิ่มขึ้นในการค้นหาค้นหาบริการ โดยการระบุเงื่อนไขตามคุณสมบัติของบริการเพื่อคัดเลือกบริการนั้น จะไม่จำกัดอยู่ที่การเปรียบเทียบค่าตรงกันเท่านั้น โดยจะเพิ่มขยายลักษณะการบันทึกรายการที่บริการลูคัฟเพื่อให้สะดวกมากขึ้นในการเปรียบเทียบค่า อันจะทำให้ผู้รับบริการสามารถระบุเงื่อนไขและค้นหาค้นหาบริการได้อย่างยืดหยุ่นมากขึ้น

งานวิจัยดังกล่าวทำการออกแบบส่วนประกอบต่างๆ ของส่วนเพิ่มขยายบริการลูคัฟ เพื่อเพิ่มความสามารถในการค้นหาค้นหาบริการดังนี้

1. บริการผู้ช่วยลुकัฟ (Lookup Helper Service) และส่วนจัดเก็บข้อมูลบริการ โดยจะทำการปรับปรุงรูปแบบการจัดเก็บข้อมูลบริการ ซึ่งแต่เดิมจะจัดเก็บในลักษณะของวัตถุแมร์แชล (Marshaled Objects) ให้จัดเก็บในลักษณะของวัตถุอันแมร์แชล (Unmarshalled Objects) ด้วยเพื่อนำไปใช้ในการเปรียบเทียบตามเงื่อนไขในการค้นหาที่มีหลากหลายรูปแบบมากขึ้น

2. ออกแบบแผ่นแบบ (Template) ให้สามารถรองรับการค้นหาบริการในรูปแบบที่หลากหลายขึ้น โดยสามารถระบุตัวดำเนินการแบบต่าง ๆ ที่ใช้ในการระบุเงื่อนไขสำหรับการค้นหาบริการได้ โดยตัวดำเนินการที่สามารถระบุเพิ่มเติม ได้แก่ ตัวดำเนินการเปรียบเทียบ การเป็นสมาชิกในเซต สายอักขระย่อย ค่าข้อมูลสูงสุดและค่าข้อมูลต่ำสุด



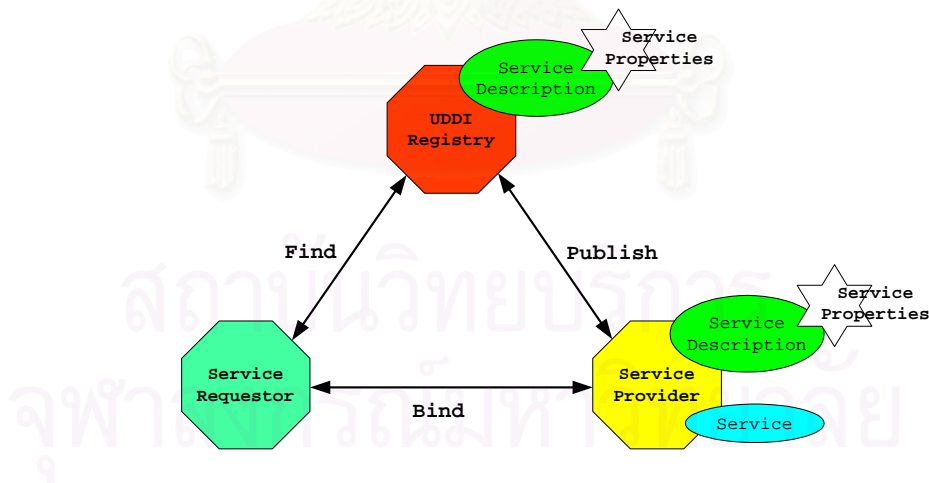
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

### บทที่ 3

#### การออกแบบส่วนขยายยูดีดีไอ

ในบทนี้จะกล่าวถึงการออกแบบโครงสร้างของส่วนขยายยูดีดีไอ ซึ่งประกอบด้วยโครงสร้างข้อมูลของคุณสมบัติของบริการ เพื่อให้รองรับคุณสมบัติของบริการที่กำหนดเพิ่มเติมขึ้นมาจากที่กำหนดไว้ในมาตรฐานยูดีดีไอ รวมไปถึงส่วนต่อประสานโปรแกรมที่สนับสนุนการโฆษณาและค้นหาบริการโดยใช้คุณสมบัติของบริการที่กำหนดเพิ่มขึ้นมานั้น

เนื่องจากการค้นหาบริการตามมาตรฐานยูดีดีไอนั้น สามารถกระทำได้เพียงการค้นหาตามชื่อธุรกิจ หรือชื่อของบริการเท่านั้น ทำให้ในบางครั้งจะได้ผลลัพธ์ของการค้นหาเป็นจำนวนมาก เพราะคุณสมบัติของบริการที่ใช้เป็นเงื่อนไขในการกรอง (Filter) บริการยังไม่ละเอียดนัก และการโฆษณาบริการยังไม่สามารถทำการกำหนดคุณสมบัติเฉพาะของบริการเพิ่มจากที่กำหนดไว้ในมาตรฐานยูดีดีไอได้ ในงานวิจัยนี้จึงต้องการขยายความสามารถของการโฆษณาบริการให้สามารถโฆษณาคุณสมบัติเฉพาะตัวของบริการได้ และสามารถทำการค้นหาบริการโดยการระบุคุณสมบัติเฉพาะเหล่านี้ได้ อันจะทำให้การกรองข้อมูลทำได้ละเอียดขึ้น โดยคุณสมบัติเหล่านี้องค์กรผู้ให้บริการจะเป็นผู้กำหนดให้กับบริการของตน และผู้รับบริการสามารถติดต่อยูดีดีไอเพื่อค้นหาบริการโดยระบุเงื่อนไขตามคุณสมบัติของบริการเหล่านี้ได้ ดังในรูปที่ 3.1



รูปที่ 3.1 การเพิ่มส่วนขยายของยูดีดีไอเพื่อรองรับคุณสมบัติของบริการ (Service Properties)

จากรูปที่ 3.1 แต่เดิมนั้นผู้ให้บริการ (Service Provider) มีบริการ (Service) ที่ต้องการโฆษณา ก็จะมีการสร้างข้อมูลโฆษณาบริการตามข้อกำหนดของยูดีดีไอ (Service Description) จากนั้นจะส่งข้อมูลเหล่านี้ไปเก็บไว้ที่ยูดีดีไอ รีจิสตรี ซึ่งเป็นข้อมูลที่เป็นไปตามข้อกำหนดเดิมของยู

ดีดีไอซึ่งยังไม่รองรับการอธิบายคุณสมบัติอื่นๆ (Service Properties) ของบริการ ดังนั้นในการเพิ่มความสามารถของยูดีดีไอให้รองรับคุณสมบัติของบริการเพิ่มเติมได้ จึงต้องทำการกำหนดส่วนโครงสร้างข้อมูลที่สามารถอธิบายถึงคุณสมบัติของบริการแต่ละบริการ และทำการขยายโครงสร้างส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณาและการค้นหาบริการ ข้อมูลคุณสมบัติบริการที่ประกาศเพิ่มเติม จะถูกส่งไปเก็บไว้ที่ยูดีดีไอ รีจิสตรี เพื่อรอให้ผู้รับบริการ (Service Requester) มาทำการค้นหาบริการโดยใช้คุณสมบัติเหล่านี้เป็นเงื่อนไขในการค้นหาได้

### 3.1 การออกแบบคุณสมบัติบริการ

คุณสมบัติของบริการนั้นเป็นข้อมูลที่สามารถแสดงถึงคุณลักษณะของบริการได้ ซึ่งจะช่วยแยกแยะบริการต่างๆ ออกจากกัน ในบริการหนึ่งๆ นั้นสามารถมีได้หลายคุณสมบัติ โดยแต่ละคุณสมบัติที่มีอยู่ในบริการจะต้องมีความสัมพันธ์เกี่ยวข้องกับบริการนั้นๆ ไม่เช่นนั้นคุณสมบัติดังกล่าวจะไม่ได้ชี้ถึงคุณลักษณะของบริการได้เลย และอาจจะทำให้เกิดการสับสนกับบริการนั้นได้ เช่น บริการโอนเงินอัตโนมัติ มีคุณสมบัติที่เกี่ยวข้องคือ ค่าธรรมเนียมในการใช้บริการการโอนเงินจำนวนเงินที่สามารถโอนได้สูงสุด เป็นต้น คุณสมบัติเหล่านี้เป็นคุณสมบัติที่ชี้เฉพาะคุณลักษณะของบริการได้ ส่วนตัวอย่างของคุณสมบัติที่ไม่เกี่ยวข้องกับบริการโอนเงินอัตโนมัติเช่น การเรียกเก็บเงินปลายทางนั้นไม่ได้สื่อถึงการให้บริการโอนเงินอัตโนมัติ ถ้านำมาโฆษณาเป็นคุณสมบัติของบริการการโอนเงินอัตโนมัติจะทำให้สับสนว่า บริการโอนเงินอัตโนมัติไปปลายทางแล้ว ทำไมจึงต้องมีการเรียกเก็บเงินที่ปลายทางอีก แต่ในงานวิจัยชิ้นนี้ไม่ได้ทำการกำหนดว่าบริการใดควรมีคุณสมบัติใดบ้างจึงจะเหมาะสม หากแต่ต้องการที่จะขยายข้อกำหนดของยูดีดีไอให้สามารถทำการระบุคุณสมบัติให้กับบริการได้ เพื่อให้สามารถใช้คุณสมบัติเหล่านั้นเป็นเงื่อนไขในการค้นหาบริการให้ได้ตรงตามความต้องการมากขึ้น ซึ่งจะเป็นการลดภาระในการคัดเลือกข้อมูลบริการจำนวนมากที่ได้จากการค้นหา

บริการเทรดเดอร์ ซึ่งเป็นบริการที่ใช้ในการโฆษณาและการค้นหาบริการของระบบคอร์บานั้น สามารถนำมาใช้เป็นแนวคิดในการพัฒนาส่วนขยายยูดีดีไอได้ โดยการโฆษณาบริการภายในบริการเทรดเดอร์นั้น ต้องทำการโฆษณา 2 ส่วน คือ ชนิดของบริการ (Service Type) และ ข้อเสนอบริการ (Service Offer)

ชนิดของบริการเป็นการกำหนดส่วนต่อประสานและชุดของคุณสมบัติสำหรับบริการหนึ่งๆ โดยมีรูปแบบดังนี้

```
service <ServiceTypeName>[:<BaseServiceTypeName>
[,<BaseServiceTypeName>]*]{
interface <InterfaceTypeName>;
[[mandatory] [readonly] property <IDLType> <PropertyName>;]*
}
```

ตัวอย่างชนิดของบริการจองห้องพักโรงแรม ประกอบด้วยส่วนต่อประสานไอดีแอลของบริการ และชุดของคุณสมบัติ ซึ่งประกอบไปด้วยคุณสมบัติคือ ชื่อโรงแรม ส่วนลดค่าบริการ ความสามารถในการจองห้องพักแบบออนไลน์ได้ โดยแสดงได้ดังนี้

```
Service HotelReservationType
{
interface Hotel;
mandatory property string Hotel_Name;
property int discount;
mandatory property boolean onlineBooking;
};
```

จากชนิดของบริการ HotelReservation ข้างต้นสามารถนำมาโฆษณาเป็นข้อเสนอบริการได้ ตัวอย่างเช่น โรงแรม Hyatt ซึ่งต้องการโฆษณาบริการจองห้องพักโรงแรม ก็สามารถที่จะทำการอ้างถึงชนิดของบริการและทำการกำหนดค่าของคุณสมบัติต่างๆ ของตน ตามที่มีกำหนดไว้ในชนิดของบริการHotelReservation ดังนี้

```
HotelReservationType HotelReservation = new HotelReservation();
HotelReservation.Hotel_Name("Hyatt");
HotelReservation.discount(5);
HotelReservation.onlineBooking(true);
```

จากการที่สามารถกำหนดคุณสมบัติให้กับบริการได้ จึงได้นำแนวคิดนี้มาพัฒนาเป็นส่วนขยายยูทิลิตี้ โดยนำเอาคุณสมบัติไปโฆษณาเป็นคุณสมบัติของบริการ เพื่อแสดงคุณลักษณะเฉพาะของบริการได้ โดยสามารถที่จะนำเอาคุณสมบัติใดๆ ที่เกี่ยวข้องและสื่อถึงบริการไปอธิบายเป็นคุณสมบัติก็ได้ หรือสามารถทำการออกแบบเป็นเซตของคุณสมบัติ เพื่อให้องค์กรธุรกิจสามารถอธิบายบริการของตนโดยใช้เซตของคุณสมบัติที่มีการโฆษณาไว้ก่อนได้ โดยเซตของคุณสมบัตินั้น จะประกอบไปด้วยคุณสมบัติที่เกี่ยวข้องกับบริการหนึ่งและมีความสัมพันธ์เกี่ยวเนื่องกันระหว่างคุณสมบัติเหล่านั้นด้วย

ส่วนขยายยูทิลิตี้ไอนั้น ประกอบด้วย คุณสมบัติ เซตของคุณสมบัติ และคุณสมบัติของบริการ ดังรายละเอียดต่อไปนี้

- คุณสมบัติ (Property) เป็นข้อมูลเกี่ยวกับแง่มุมหรือลักษณะหนึ่งๆ ของสิ่งที่สนใจ ตัวอย่างเช่น ส่วนลด (Discount) เป็นคุณสมบัติที่อธิบายการให้ส่วนลดโดยทั่วไป โดยที่ยังไม่ได้ระบุว่าส่วนลดนี้เป็นส่วนลดของอะไร หรือใช้กับบริการใด โครงสร้างของคุณสมบัติจะประกอบด้วย ชื่อคุณสมบัติ ค่าอธิบายคุณสมบัติ ชนิดของข้อมูล ชนิดลักษณะของข้อมูล คุณลักษณะความจำเป็นของข้อมูล
  - ชนิดของข้อมูล เป็นการกำหนดถึงชนิดของค่าข้อมูลของคุณสมบัติ เช่น char, string, integer เป็นต้น



- ชนิดลักษณะของข้อมูล เป็นการกำหนดถึงชนิดของลักษณะของข้อมูลว่า ลักษณะข้อมูลของคุณสมบัติเป็นแบบใด เช่น simple, array, structure เป็นต้น
- คุณลักษณะความจำเป็นของข้อมูล เป็นการกำหนดถึงความจำเป็นของคุณสมบัตินี้ เช่น optional, mandatory, readonly เป็นต้น ซึ่งบ่งชี้ว่า ต้องทำการกำหนดค่าให้กับคุณสมบัตินี้หรือไม่หรือแก้ไขค่าคุณสมบัตินี้ได้หรือไม่

ตัวอย่างคุณสมบัติ ส่วนลดค่าบริการ ที่อธิบายโดยใช้โครงสร้างข้อมูลของส่วนขยายยูทิลิตี้ได้ดังนี้

```
<propertyReference mode="OPTIONAL">
  <name>discount</name>
  <description>discount for service</description>
  <valueReference valueTypeRef="simpleType">
    <simpleType>
      <simpleValue type="integer"/>
    </simpleType>
  </valueReference>
</propertyReference>
```

- เซตของคุณสมบัติ (propertySet) เป็นชุดของคุณสมบัติที่มีความเกี่ยวเนื่องกัน และมีความสัมพันธ์กัน ซึ่งสามารถร่วมกันบ่งชี้คุณลักษณะของสิ่งที่สนใจได้ ในงานวิจัยนี้ไม่ได้กำหนดถึงเซตของคุณสมบัติที่บริการหนึ่งๆ ควรจะมี หากแต่ต้องการสร้างโครงสร้าง ที่สามารถรองรับการกำหนดเซตของคุณสมบัติได้ โดยโครงสร้างของเซตของคุณสมบัติประกอบด้วย ชื่อของเซตของคุณสมบัติ คำอธิบาย และถูกรวบรวมคุณสมบัตินั้นในเซต

ตัวอย่างของเซตของคุณสมบัติ เช่น เซตของคุณสมบัติของบริการจองห้องพัก โรงแรม เป็นต้น สามารถอธิบายโดยใช้โครงสร้างข้อมูลของส่วนขยายยูทิลิตี้ได้ดังนี้

```
<propertySet mode="OPTIONAL">
  <name>HotelReservationType</name>
  <description>serviceType of Hotel Reservation</description>
  <propertyBag>
    <propertyReference mode="MANDATORY">
      <name>Hotel_Name</name>
      <description>Hotel Name</description>
      <valueReference valueTypeRef="simpleType">
        <simpleType>
          <simpleValue type="string"/>
        </simpleType>
      </valueReference>
    </propertyReference>
  </propertyBag>
</propertySet>
```

```

    </valueReference>
  </propertyReference>
  <propertyReference mode="OPTIONAL">
    <name>discount</name>
    <description>discount for service</description>
    <valueReference valueTypeRef="simpleType">
      <simpleType>
        <simpleValue type="integer"/>
      </simpleType>
    </valueReference>
  </propertyReference>
  <propertyReference mode="MANDATORY">
    <name>onlineBooking</name>
    <description>available for online booking</description>
    <valueReference valueTypeRef="simpleType">
      <simpleType>
        <simpleValue type="boolean"/>
      </simpleType>
    </valueReference>
  </propertyReference>
</propertyBag>
</propertySet>

```

จากตัวอย่างจะพบว่าเซตของคุณสมบัตินั้นจะมีโครงสร้างที่คล้ายคลึงกับชนิดของบริการ (serviceType) ของบริการเทอร์เรเตอร์ ยกเว้นส่วนต่อประสานเท่านั้นที่ส่วนขยายยูทิลิตี้ไม่ได้รองรับ เนื่องจากในยูทิลิตี้มาตรฐานมีส่วนที่ทำหน้าที่ในการอธิบายถึงส่วนต่อประสานของบริการอยู่แล้ว คือส่วนอ้างอิงไปยังที่โมเดลของดับเบิลยูเอสดีแอล

- คุณสมบัติของบริการ (servicePropertyBag) เป็นการนำเอา คุณสมบัติ และ/หรือเซตของคุณสมบัติ มาอธิบายถึงคุณลักษณะของบริการที่ต้องการโฆษณา โดยกำหนดค่าข้อมูลให้กับคุณสมบัตินั้นๆ เพื่อเป็นการแสดงถึงคุณลักษณะของบริการของตน ในส่วนขยายยูทิลิตี้สามารถที่จะทำการอ้างถึงคุณสมบัตินั้นๆ หรือเซตของคุณสมบัติได้หลายตัว ซึ่งในส่วนนี้จะมีความคล้ายคลึงกับข้อเสนอบริการในบริการเทอร์เรเตอร์ กล่าวคือ องค์การธุรกิจจะต้องนำเอาคุณสมบัตินั้นๆ มากำหนดค่าข้อมูลให้กับบริการ

ตัวอย่างคุณสมบัติของบริการ เช่น โรงแรม Hyatt ต้องการโฆษณาบริการจองห้องพักโรงแรม พร้อมโฆษณาคุณสมบัตินั้นๆ เป็นต้น สามารถอธิบายโดยใช้โครงสร้างข้อมูลของส่วนขยายยูทิลิตี้ได้ดังนี้

```

<servicePropertyBag>
  <propertyBag>
    <propertyReference mode="MANDATORY">
      <name>Hotel_Name</name>
      <description>Hotel Name</description>
      <valueReference valueTypeRef="simpleType">

```

```

        <simpleType>
            <simpleValue type="string">Hyatt</simpleValue>
        </simpleType>
    </valueReference>
</propertyReference>
<propertyReference mode="OPTIONAL">
    <name>discount</name>
    <description>discount for service</description>
    <valueReference valueTypeRef="simpleType">
        <simpleType>
            <simpleValue type="integer">5</simpleValue>
        </simpleType>
    </valueReference>
</propertyReference>
<propertyReference mode="MANDATORY">
    <name>onlineBooking</name>
    <description>available for online booking</description>
    <valueReference valueTypeRef="simpleType">
        <simpleType>
            <simpleValue type="boolean">true</simpleValue>
        </simpleType>
    </valueReference>
</propertyReference>
</propertyBag>
<propertySet>
    <name>room price</name>
    <description>price in US Dollars</description>
    <propertyBag>
        <propertyReference mode="OPTIONAL">
            <name>single-bed room</name>
            <description>price for single-bed room per
day</description>
            <valueReference valueTypeRef="simpleType">
                <simpleType>
                    <simpleValue type="currency">75</simpleValue>
                </simpleType>
            </valueReference>
        </propertyReference>
        <propertyReference mode="OPTIONAL">
            <name>double-bed room</name>
            <description>price for double-bed room per
day</description>
            <valueReference valueTypeRef="simpleType">
                <simpleType>
                    <simpleValue type="currency">99</simpleValue>
                </simpleType>
            </valueReference>
        </propertyReference>
        <propertyReference mode="OPTIONAL">
            <name>extra-bed</name>
            <description>price for extra-bed per day</description>
            <valueReference valueTypeRef="simpleType">
                <simpleType>
                    <simpleValue type="currency">40</simpleValue>
                </simpleType>
            </valueReference>
        </propertyReference>
    </propertyBag>

```

</propertySet>  
</ servicePropertyBag>

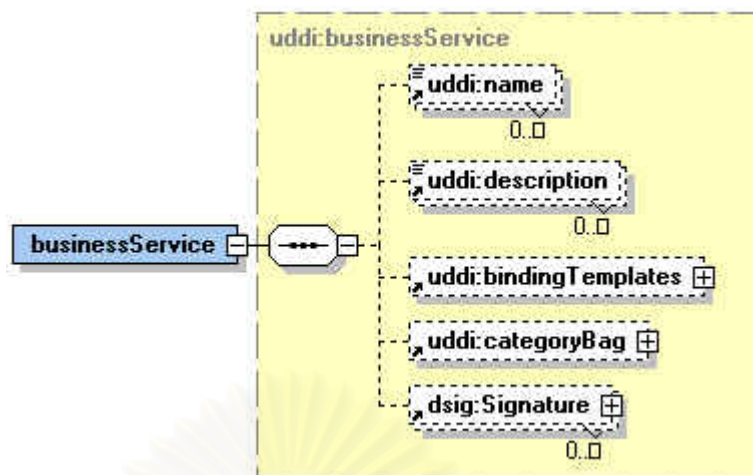
จากตัวอย่างทั้งเซตของคุณสมบัติ และคุณสมบัติบริการนั้น ในส่วนขยายยูดีดีไอในงานวิจัยนี้ ยังไม่ได้กำหนดว่าคุณสมบัติใดควรบรรจุอยู่ในเซตของคุณสมบัติหรือควรเป็นคุณสมบัติเดี่ยวๆ ที่ประกาศให้กับบริการ แต่งานวิจัยมุ่งเน้นที่การพัฒนาโครงสร้างเพื่อรองรับการประกาศคุณสมบัติและเซตของคุณสมบัติมากกว่า โดยเปิดโอกาสให้ผู้ให้บริการเลือกทำการประกาศเองเพื่อความยืดหยุ่น ซึ่งในตัวอย่างข้างต้น คุณสมบัติได้แสดงการประกาศคุณสมบัติต่างๆ ของบริการจองโรงแรมของ Hyatt เป็นคุณสมบัติเดี่ยวๆ ยกเว้นข้อมูลราคาห้องพักแบบต่างๆ ที่ได้กำหนดไว้เป็นเซตของคุณสมบัติ

คุณสมบัตินี้ของบริการเป็นการเพิ่มคุณสมบัติให้กับบริการ จึงต้องนำโครงสร้างนี้ไปผูกกับโครงสร้างบริการของยูดีดีไอเดิม ได้แก่ ส่วนของบิสซิเนสเซอร์วิส โดยงานวิจัยนี้จะทำการขยายยูดีดีไอในส่วนของโครงสร้างของส่วนต่อประสานการค้นหา และการโฆษณา เพื่อให้สามารถนำโครงสร้างของคุณสมบัติไปประกาศเป็นคุณสมบัติให้กับบริการได้ และทำการค้นหาโดยใช้คุณสมบัติเหล่านั้นเป็นเงื่อนไขในการค้นหาได้

คุณสมบัตินี้ที่เพิ่มให้กับบริการนั้น ทำให้โครงสร้างของบิสซิเนสเซอร์วิสเปลี่ยนไป แต่การเพิ่มเติมนั้นทำในส่วนขยายโดยไม่กระทบกับยูดีดีไอเดิม นอกจากนี้โครงสร้างที่จะกำหนดในส่วนขยายยูดีดีไอนี้ จะอ้างอิงถึงวิธีการกำหนดโครงสร้างจากโครงสร้างของยูดีดีไอเดิม เพื่อให้สอดคล้องกัน

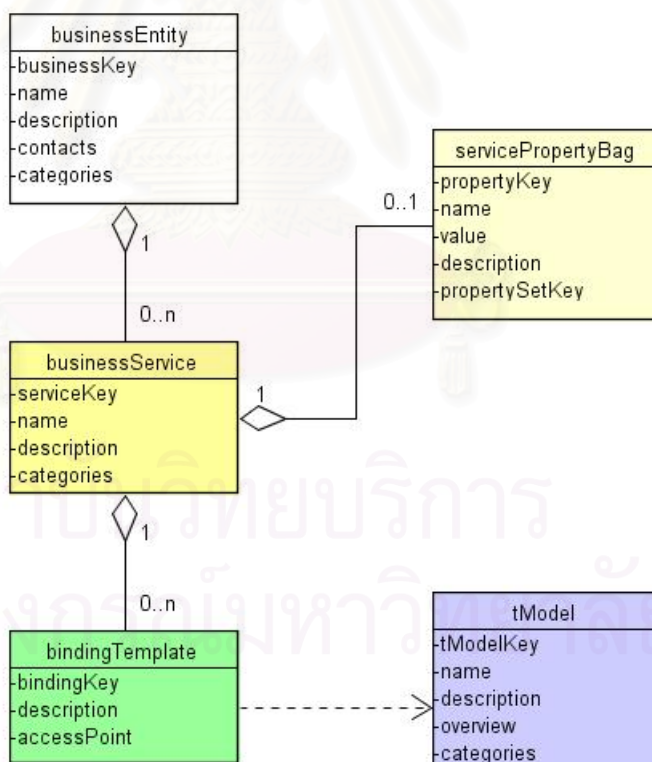
### 3.2 การปรับเปลี่ยนโครงสร้างข้อมูลของยูดีดีไอ (UDDI Data Structure)

โครงสร้างข้อมูลหลักตามข้อกำหนดของยูดีดีไอ [4] นั้น ประกอบด้วย 4 ส่วนคือ (1) บิสซิเนสเอ็นทิตี (businessEntity) เป็นอิลิเมนต์ระดับบนสุด ซึ่งเก็บข้อมูลโฆษณาเกี่ยวกับธุรกิจทั้งหมดที่ผู้ให้บริการโฆษณา (2) บิสซิเนสเซอร์วิส (businessService) เป็นอิลิเมนต์ในระดับถัดมา ซึ่งเก็บข้อมูลโฆษณาเกี่ยวกับบริการแต่ละบริการขององค์กรธุรกิจนั้นๆ (3) ไบนดิงเทมเพลต (bindingTemplate) เป็นอิลิเมนต์ที่อยู่ภายใต้อิลิเมนต์บิสซิเนสเซอร์วิส ซึ่งแสดงรายละเอียดในการเรียกใช้บริการ และ (4) ทีโมเดล (tModel) เป็นอิลิเมนต์ที่แทรกอยู่ตามอิลิเมนต์ต่างๆ ที่กล่าวมาแล้วเพื่อแสดงข้อมูลรายการอ้างอิงทางเทคนิค ในส่วนของโครงสร้างข้อมูลของบิสซิเนสเซอร์วิสตามข้อกำหนดยูดีดีไอที่อธิบายถึงบริการที่ธุรกิจให้บริการนั้น เป็นส่วนที่ต้องทำการขยายให้สามารถทำการรองรับการโฆษณาคุณสมบัติของบริการได้ โดยมีโครงสร้างเดิมดังแสดงในรูปที่ 3.2



รูปที่ 3.2 โครงสร้างของบิสิเนสเซอร์วิส ตามข้อกำหนดยูดีดีไอ

เพื่อให้โครงสร้างข้อมูลสามารถรองรับการโฆษณาคุณสมบัติของบริการได้ จึงทำการเพิ่มโครงสร้างข้อมูลของส่วนที่เรียกว่า ถุงคุณสมบัติบริการ (servicePropertyBag) เพื่อให้รองรับคุณสมบัติของบริการที่เพิ่มเติม โดยมีแนวคิดดังรูปที่ 3.3

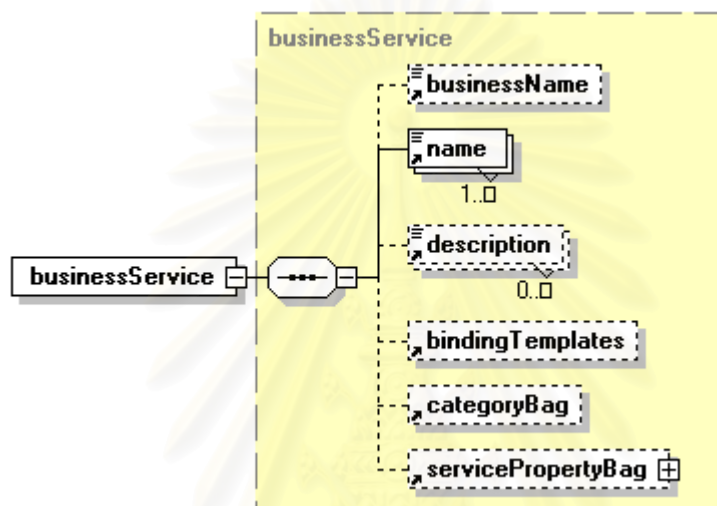


รูปที่ 3.3 โครงสร้างข้อมูลของยูดีดีไอที่เพิ่มถุงคุณสมบัติบริการ

โครงสร้างข้อมูลส่วนขยายยูดีดีไอ สามารถอธิบายลักษณะโครงสร้างที่สำคัญ (แสดงข้อกำหนดเชิงโครงสร้างส่วนขยายยูดีดีไอทั้งหมด ในภาคผนวก ก) ได้ดังนี้

### 3.2.1 โครงสร้างของบิสิเนสเซอร์วิส (businessService)

เนื่องจากคุณสมบัติของบริการที่เพิ่มขึ้นมานั้น เป็นคุณสมบัติที่จะอธิบายลักษณะเฉพาะตัวของแต่ละบริการ ผู้ให้บริการสามารถที่จะโฆษณาข้อมูลเหล่านั้นไว้ในคำอธิบายบริการที่ตนมีอยู่ได้ ดังนั้นจึงทำการเพิ่มโครงสร้างของคุณสมบัติของบริการเข้าไปในบิสิเนสเซอร์วิสเดิมของยูดีดีไอ แต่เนื่องจากคุณสมบัติของบริการอาจมีได้หลายตัว จึงบรรจุคุณสมบัติเหล่านั้นไว้ในถุงคุณสมบัติบริการ (servicePropertyBag) ดังในรูปที่ 3.4



รูปที่ 3.4 โครงสร้างของบิสิเนสเซอร์วิสที่เพิ่มถุงคุณสมบัติบริการ

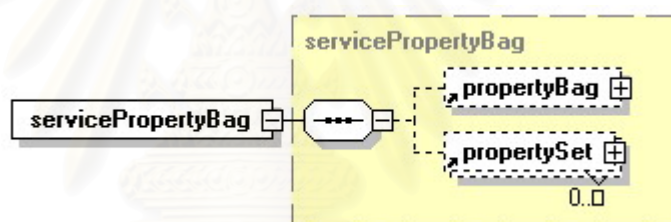
โครงสร้างข้อมูลของบิสิเนสเซอร์วิส ประกอบด้วย

- แอตทริบิวต์
  - serviceKey แสดงรหัสประจำตัวของบริการ (อยู่ในมาตรฐานยูดีดีไอเดิม)
  - businessKey แสดงรหัสประจำตัวของธุรกิจ (อยู่ในมาตรฐานยูดีดีไอเดิม)
- อิลิเมนต์
  - businessName แสดงชื่อของธุรกิจ ซึ่งจะระบุหรือไม่มีก็ได้ (อยู่ในมาตรฐานยูดีดีไอเดิม)
  - name แสดงชื่อของบริการ (อยู่ในมาตรฐานยูดีดีไอเดิม)
  - description แสดงคำอธิบายบริการ ซึ่งจะระบุหรือไม่มีก็ได้ (อยู่ในมาตรฐานยูดีดีไอเดิม)

- bindingTemplates แสดงเซตของคุณสมบัติสำหรับการผูกมัดบริการ ซึ่งจะระบุหรือไม่ก็ได้ (อยู่ในมาตรฐานยูดีดีไอเดิม)
- categoryBag แสดงการจำแนกประเภทบริการ ซึ่งจะระบุหรือไม่ก็ได้ (อยู่ในมาตรฐานยูดีดีไอเดิม)
- servicePropertyBag แสดงถุงใส่คุณสมบัติของบริการ ซึ่งจะระบุหรือไม่ก็ได้ (เพิ่มขยายจากมาตรฐานยูดีดีไอ)

### 3.2.2 โครงสร้างของถุงคุณสมบัติบริการ (servicePropertyBag)

ถุงคุณสมบัติบริการ เป็นส่วนขยายยูดีดีไอที่เพิ่มคุณสมบัติบริการ ซึ่งประกอบด้วยคุณสมบัติของบริการที่ผู้ให้บริการระบุรายละเอียดเพื่อกำหนดลักษณะบริการของตน โดยคุณสมบัติบริการสามารถเพิ่มให้กับบริการได้โดยตรง หรือกำหนดเซตของคุณสมบัติที่กำหนดคุณสมบัติหลายๆ คุณสมบัติที่มีความเกี่ยวข้องในเรื่องเดียวกันไว้แล้วก็ได้ มีโครงสร้างข้อมูลดังรูปที่ 3.5



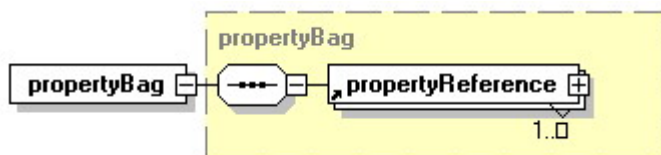
รูปที่ 3.5 โครงสร้างของถุงคุณสมบัติบริการ

โครงสร้างของถุงคุณสมบัติบริการ ประกอบด้วย

- อิลิเมนต์
  - propertyBag แสดงกลุ่มของคุณสมบัติบริการ
  - propertySet แสดงเซตของคุณสมบัติบริการที่มีการกำหนดกลุ่มของคุณสมบัติไว้แล้ว

#### 3.2.2.1 โครงสร้างของ propertyBag

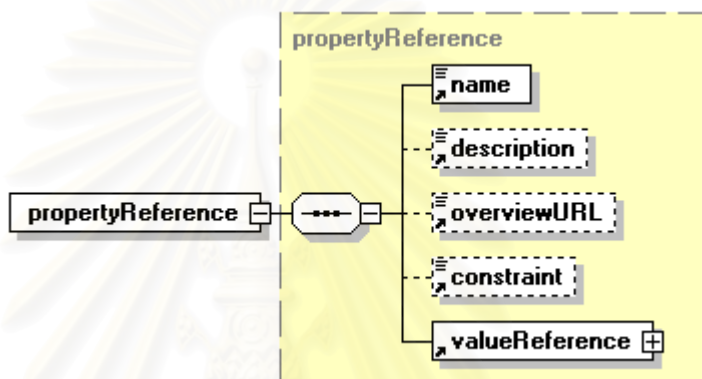
เป็นส่วนที่แสดงข้อมูลคุณสมบัติบริการที่ผู้ให้บริการสามารถกำหนดคุณสมบัติของบริการของตนได้ โดยกำหนดในอิลิเมนต์การอ้างถึงคุณสมบัติ (propertyReference) ดังรูปที่



รูปที่ 3.6 โครงสร้างของ propertyBag

### 3.2.2.2 โครงสร้างของ propertyReference

เป็นส่วนที่แสดงรายละเอียดคุณลักษณะของคุณสมบัติของบริการ ดังในรูปที่ 3.7



รูปที่ 3.7 โครงสร้างของ propertyReference

โครงสร้างของ propertyReference ประกอบด้วย

- แอตทริบิวต์
  - propertyKey แสดงรหัสประจำตัวของคุณสมบัติบริการ
  - mode แสดงคุณลักษณะความจำเป็นของโครงสร้างข้อมูล ซึ่งปรับใช้มาจาก [6] ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 คุณลักษณะความจำเป็นของข้อมูล (mode)

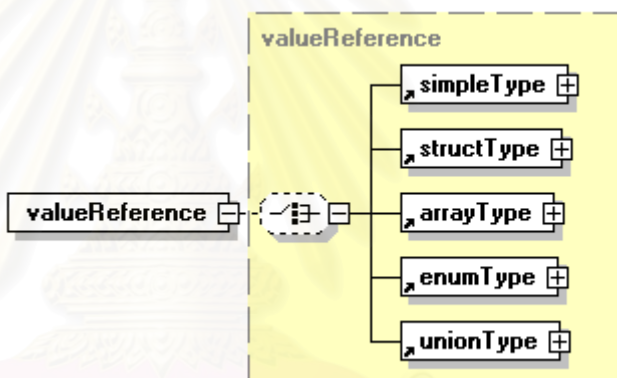
mode	คำอธิบาย
OPTIONAL	จะระบุใช้โครงสร้างนี้หรือไม่ก็ได้
READONLY	โครงสร้างเป็นโครงสร้างแบบอ่านอย่างเดียว
MANDATORY	ต้องแสดงโครงสร้างข้อมูลนี้
MANDATORY_READONLY	ต้องแสดงโครงสร้างข้อมูลนี้ และเป็นโครงสร้างแบบอ่านอย่างเดียว



- อีลิเมนต์
  - name แสดงชื่อของคุณสมบัติบริการ
  - description แสดงคำอธิบายของคุณสมบัติบริการ
  - overviewURL แสดงที่อยู่บนอินเทอร์เน็ตของคำอธิบายลักษณะทั่วไปของคุณสมบัติบริการ
  - constraint แสดงข้อบังคับของคุณสมบัติบริการ
  - valueReference แสดงลักษณะของค่าข้อมูลของคุณสมบัติบริการ

### 3.2.2.3 โครงสร้างของ valueReference

เป็นส่วนที่แสดงข้อมูลเกี่ยวกับลักษณะข้อมูลของคุณสมบัติของบริการ ซึ่งได้แก่ ชนิดข้อมูล (Data Type) ของค่าข้อมูล โดยปรับใช้มาจาก [6] ดังรูปที่ 3.8



รูปที่ 3.8 โครงสร้างของ valueReference

โครงสร้างของ valueReference ประกอบด้วย

- แอตทริบิวต์
  - valueTypeRef แสดงชนิดของลักษณะข้อมูล ดังแสดงในตารางที่

3.2

ตารางที่ 3.2 ชนิดของลักษณะข้อมูล (valueTypeRef)

valueTypeRef	คำอธิบาย
simpleType	ลักษณะข้อมูลเป็นแบบอย่างง่าย
structType	ลักษณะข้อมูลเป็นแบบมีโครงสร้าง
arrayType	ลักษณะข้อมูลเป็นแบบอาร์เรย์

ตารางที่ 3.2 ชนิดของลักษณะข้อมูล (valueTypeRef) (ต่อ)

valueTypeRef	คำอธิบาย
enumType	ลักษณะข้อมูลเป็นแบบแจกจ่าย
unionType	ลักษณะข้อมูลเป็นแบบยูเนียน

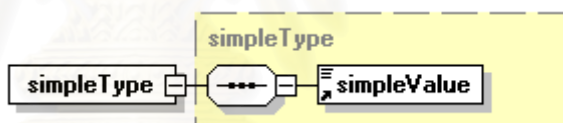
- อิลิเมนต์

สามารถระบุได้เพียงอย่างใดอย่างหนึ่งเท่านั้น โดยเลือกจากชนิดโครงสร้างของลักษณะข้อมูล ดังนี้

- simpleType แสดงโครงสร้างของลักษณะข้อมูลแบบอย่างง่าย
- structType แสดงโครงสร้างของลักษณะข้อมูลแบบมีโครงสร้าง
- arrayType แสดงโครงสร้างของลักษณะข้อมูลแบบอาร์เรย์
- enumType แสดงโครงสร้างของลักษณะข้อมูลแบบแจกจ่าย
- unionType แสดงโครงสร้างของลักษณะข้อมูลแบบยูเนียน

### 3.2.2.4 โครงสร้างของ simpleType

เป็นส่วนที่แสดงข้อมูลของลักษณะคุณสมบัติบริการแบบอย่างง่าย ดังรูปที่ 3.9



รูปที่ 3.9 โครงสร้างของ simpleType

โครงสร้างของ simpleType ประกอบด้วย

- แอตทริบิวต์

- simpleKey แสดงรหัสประจำตัวของโครงสร้างลักษณะข้อมูลแบบอย่างง่าย

- อิลิเมนต์

- simpleValue แสดงค่าของคุณสมบัติบริการ โดยมีแอตทริบิวต์เป็น type ซึ่งเป็นการกำหนดชนิดของข้อมูล ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 ชนิดของข้อมูล (type)

type	คำอธิบาย
char	ข้อมูลแบบตัวอักษรหนึ่งตัว

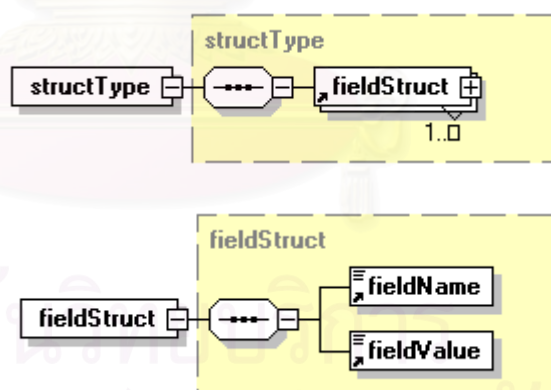
ตารางที่ 3.3 ชนิดของข้อมูล (type) (ต่อ)

type	คำอธิบาย
string	ข้อมูลแบบชุดของตัวอักษร
byte	ข้อมูลแบบตัวเลขจำนวนเต็มแบบไบนารี
integer	ข้อมูลแบบตัวเลขจำนวนเต็มอินทิเจอร์
double	ข้อมูลแบบตัวเลขแบบดับเบิล
float	ข้อมูลแบบตัวเลขแบบฟลอยด์
currency	ข้อมูลแบบตัวเลขแบบหน่วยเงิน
date	ข้อมูลแบบวันที่อย่างเดียว
time	ข้อมูลแบบเวลาอย่างเดียว
datetime	ข้อมูลแบบวันที่และเวลา
boolean	ข้อมูลแบบตรรกะ

### 3.2.2.5 โครงสร้างของ structType

เป็นส่วนที่แสดงลักษณะข้อมูลของคุณสมบัติบริการแบบมีโครงสร้าง ดังรูปที่

3.10



รูปที่ 3.10 โครงสร้างของ structType และ fieldStruct

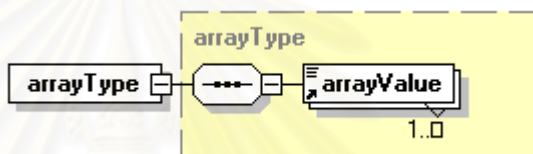
โครงสร้างของ structType ประกอบด้วย

- แอตทริบิวต์
  - structKey แสดงรหัสประจำตัวของโครงสร้างลักษณะข้อมูลแบบมีโครงสร้าง

- อิลิเมนต์
  - fieldStruct แสดงชุดลักษณะข้อมูลแบบมีโครงสร้าง ซึ่งประกอบด้วย
    - อิลิเมนต์ fieldName แสดงชื่อของฟิลด์ข้อมูล ซึ่งมีแอตทริบิวต์เป็น fieldKey แสดงรหัสประจำตัวของฟิลด์ข้อมูล
    - อิลิเมนต์ fieldValue แสดงค่าของคุณสมบัติบริการ โดยมีแอตทริบิวต์เป็น type ซึ่งเป็นการกำหนดชนิดของข้อมูล ดังแสดงในตารางที่ 3.3

### 3.2.2.6 โครงสร้างของ arrayType

เป็นส่วนที่แสดงลักษณะข้อมูลของคุณสมบัติบริการแบบอาร์เรย์ ดังรูปที่ 3.11



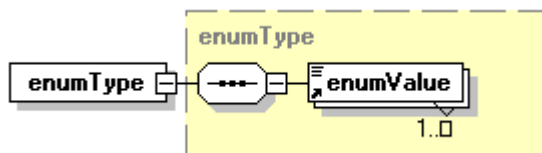
รูปที่ 3.11 โครงสร้างของ arrayType

โครงสร้างของ arrayType ประกอบด้วย

- แอตทริบิวต์
  - arrayKey แสดงรหัสประจำตัวของโครงสร้างลักษณะข้อมูลแบบอาร์เรย์หนึ่งมิติ
  - type แสดงชนิดของข้อมูลของอิลิเมนต์ในอาร์เรย์ ดังแสดงในตารางที่ 3.3
  - no\_of\_element แสดงจำนวนของอาร์เรย์อิลิเมนต์ทั้งหมด
- อิลิเมนต์
  - arrayValue แสดงค่าของคุณสมบัติบริการ โดยมีแอตทริบิวต์ดังนี้
    - index\_no แสดงลำดับที่ของอาร์เรย์อิลิเมนต์

### 3.2.2.7 โครงสร้างของ enumType

เป็นส่วนแสดงลักษณะข้อมูลของค่าคุณสมบัตินี้แบบแจกแจง ดังรูปที่ 3.12



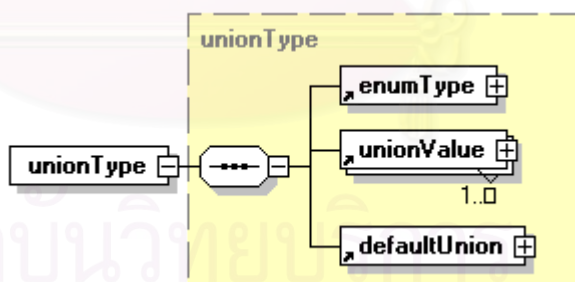
รูปที่ 3.12 โครงสร้างของ enumType

โครงสร้างของ enumType ประกอบด้วย

- แอตทริบิวต์
  - enumKey แสดงรหัสประจำตัวของโครงสร้างลักษณะข้อมูลแบบ  
แจกจ่าย
- อีลิเมนต์
  - enumValue แสดงค่าของคุณสมบัติบริการ โดยมีแอตทริบิวต์ดังนี้
    - valueRef แสดงค่าแจกจ่ายเพื่ออ้างอิงค่าข้อมูลในอีลิเมนต์  
enumValue

### 3.2.2.8 โครงสร้างของ unionType

เป็นส่วนแสดงลักษณะข้อมูลของค่าคุณสมบัติบริการแบบยูเนียน ซึ่งสามารถทำการผสมรูปแบบของโครงสร้างลักษณะของค่าข้อมูลหลายๆ รูปแบบเข้าด้วยกันได้ ดังรูปที่ 3.13



รูปที่ 3.13 โครงสร้างของ unionType

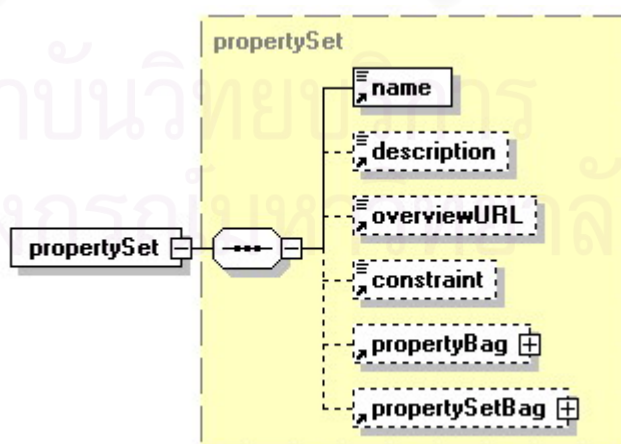
โครงสร้างของ unionType ประกอบด้วย

- แอตทริบิวต์
  - unionKey แสดงรหัสประจำตัวของโครงสร้างลักษณะข้อมูลแบบ  
ยูเนียน

- อิลิเมนต์
  - enumType แสดงข้อมูลของค่าคุณสมบัติบริการแบบแฉงนั้บ ตามหัวข้อที่ 3.2.2.7
  - unionValue แสดงชุดของข้อมูลที่มีโครงสร้างแสดงลักษณะข้อมูลตามหัวข้อที่ 3.2.2.3 โดยมีแอดทริบิวท์ ดังนี้
    - valueTypeRef แสดงค่าตามตารางที่ 3.2
    - unionValueKey แสดงรหัสประจำตัวของค่าของข้อมูลยูเนียน
    - enumValueRef แสดงค่าตัวเลขจำนวนเต็มเพื่ออ้างอิงค่าข้อมูลแบบแฉงนั้บ
  - defaultUnion แสดงข้อมูลที่มีโครงสร้างตามหัวข้อที่ 3.2.2.3 ซึ่งเป็นค่าที่กำหนดให้เป็นค่าเริ่มต้น โดยมีแอดทริบิวท์ ดังนี้
    - valueTypeRef แสดงค่าตามตารางที่ 3.2
    - defaultUnionKey แสดงรหัสประจำตัวของค่าของข้อมูลยูเนียน ที่เป็นค่าเริ่มต้น

### 3.2.2.9 โครงสร้างของ propertySet

เป็นส่วนที่แสดงข้อมูลคุณสมบัติบริการโดยมีการกำหนดเซตของคุณสมบัติไว้แล้ว ผู้ให้บริการสามารถเลือกเซตของคุณสมบัติที่จะใช้เพื่อระบุรายละเอียดของบริการตามกลุ่มของคุณสมบัติที่อยู่ในเซตของคุณสมบัติได้ โครงสร้างจะเป็นดังรูปที่ 3.14



รูปที่ 3.14 โครงสร้างของ propertySet

โครงสร้างของ propertySet ประกอบด้วย

- แอตทริบิวต์
  - propertySetKey แสดงรหัสประจำตัวของเซตของของคุณสมบัติบริการ
  - mode แสดงคุณลักษณะความจำเป็นของโครงสร้างข้อมูลนี้ ดังแสดงในตารางที่ 3.1
- อิลิเมนต์
  - name แสดงชื่อของเซตของคุณสมบัติบริการ
  - description แสดงคำอธิบายของเซตของคุณสมบัติบริการ
  - overviewURL แสดงที่อยู่บนอินเทอร์เน็ตของคำอธิบายลักษณะทั่วไปของเซตของคุณสมบัติบริการ
  - constraint แสดงข้อบังคับของเซตของคุณสมบัติบริการ
  - propertyBag แสดงกลุ่มของคุณสมบัติบริการ ดังในหัวข้อที่ 3.2.2.1
  - propertySetBag แสดงกลุ่มของเซตของคุณสมบัติบริการ

### 3.2.2.10 โครงสร้างของ propertySetBag

เป็นส่วนที่แสดงชุดเซตของกลุ่มของคุณสมบัติบริการ (รูปที่ 3.15) ซึ่งประกอบด้วยอิลิเมนต์ชุดครอบครัวของเซตของคุณสมบัติบริการ (propertySetFamily)

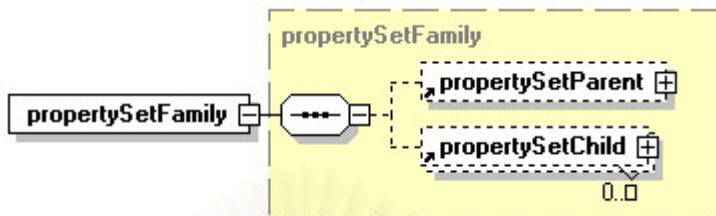


รูปที่ 3.15 โครงสร้างของ propertySetBag

### 3.2.2.11 โครงสร้างของ propertySetFamily

เป็นส่วนที่แสดงชุดครอบครัวของเซตของคุณสมบัติบริการ ซึ่งแสดงถึงเซตของคุณสมบัติที่เป็นพ่อแม่ และแสดงเซตของคุณสมบัติของลูกได้ ดังรูปที่ 3.16 ประกอบด้วย

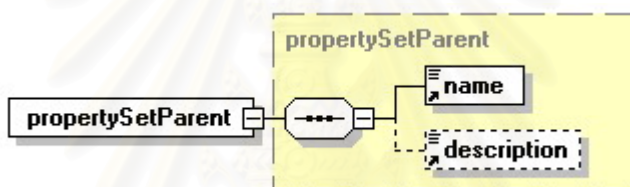
- อีลิเมนต์
  - propertySetParent แสดงเซตของคุณสมบัติบริการของพ่อแม่
  - propertySetChild แสดงชุดของเซตของคุณสมบัติบริการของลูก



รูปที่ 3.16 โครงสร้างของ propertySetFamily

### 3.2.2.12 โครงสร้างของ propertySetParent

เป็นส่วนที่แสดงการอ้างถึงเซตของคุณสมบัติบริการของพ่อแม่ ดังรูปที่ 3.17



รูปที่ 3.17 โครงสร้างของ propertySetParent

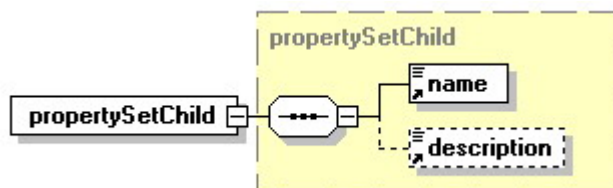
โครงสร้างของ propertySetParent ประกอบด้วย

- แอตทริบิวต์
  - propertySetKey แสดงรหัสประจำตัวของเซตของคุณสมบัติบริการของพ่อแม่
  - mode แสดงคุณลักษณะความจำเป็นของโครงสร้างข้อมูลนี้ ดังแสดงในตารางที่ 3.1
- อีลิเมนต์
  - name แสดงชื่อของเซตของคุณสมบัติบริการของพ่อแม่
  - description แสดงคำอธิบายของเซตของคุณสมบัติบริการของพ่อแม่

### 3.2.2.13 โครงสร้างของ propertySetChild

เป็นส่วนที่แสดงการอ้างถึงเซตของคุณสมบัติบริการของลูก ดังรูปที่ 3.18





รูปที่ 3.18 โครงสร้างของ propertySetChild

โครงสร้างของ propertySetChild ประกอบด้วย

- แอตทริบิวต์
  - propertySetKey แสดงรหัสประจำตัวของเซตของคุณสมบัติบริการของลูก
  - mode แสดงคุณลักษณะความจำเป็นของโครงสร้างข้อมูลนี้ ดังแสดงในตารางที่ 3.1
- อิลิเมนต์
  - name แสดงชื่อของเซตของคุณสมบัติบริการของลูก
  - description แสดงคำอธิบายของเซตของคุณสมบัติบริการของลูก

### 3.3 การขยายข้อกำหนดของส่วนต่อประสานโปรแกรมยูดีดีไอ (UDDI API Specification)

เนื่องจากส่วนต่อประสานโปรแกรมตามข้อกำหนดของยูดีดีไอนั้น ยังไม่สามารถที่จะใช้ทำการค้นหาบริการโดยอาศัยคุณสมบัติบริการที่เพิ่มเข้าไปได้ ดังนั้นจึงต้องทำการขยายข้อกำหนดเดิมของยูดีดีไอ โดยการเพิ่มโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรมยูดีดีไอ ดังนี้

#### 3.3.1 การขยายส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา (Inquiry API Function)

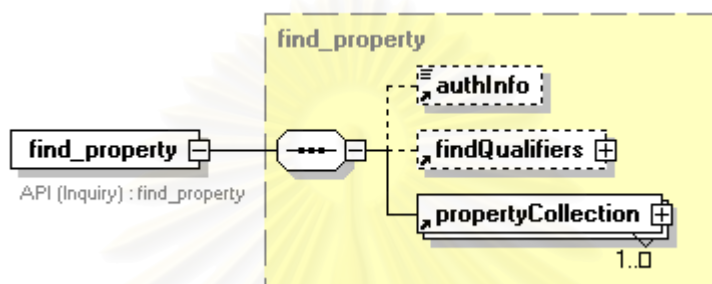
เพื่อให้สามารถทำการค้นหาบริการโดยใช้คุณสมบัติบริการที่เพิ่มขึ้นมาเป็นเงื่อนไขในการค้นหาได้ จึงขยายโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรมของยูดีดีไอ วิธีจิสตรี ดังนี้

- find\_property ผลลัพธ์ที่ได้คือ propertyList
- find\_propertySet ผลลัพธ์ที่ได้คือ propertySetList
- find\_serviceProperty ผลลัพธ์ที่ได้คือ servicePropertyList
- get\_propertyDetail ผลลัพธ์ที่ได้คือ propertyDetail
- get\_propertySetDetail ผลลัพธ์ที่ได้คือ propertySetDetail

- get\_servicePropertyDetail ผลลัพธ์ที่ได้คือ servicePropertyDetail

### 3.3.1.1 การค้นหาคุณสมบัติ (find\_property)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการค้นหาคุณสมบัติซึ่งได้ระบุเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.19 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายการคุณสมบัติ (propertyList)

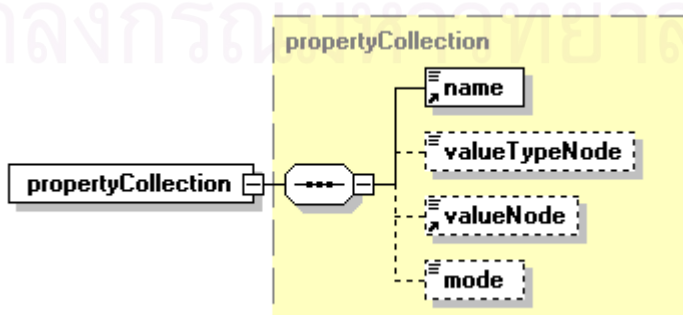


รูปที่ 3.19 โครงสร้างของ find\_property

- อิลิเมนต์
  - authinfo ข้อมูลของผู้ที่จะทำการค้นหา ซึ่งจะระบุหรือไม่ก็ได้
  - findQualifiers ระบุเงื่อนไขในการค้นข้อมูลตามรูปแบบของข้อกำหนดยูดีดีไอ ซึ่งจะระบุหรือไม่ก็ได้
  - propertyCollection ระบุคุณสมบัติที่ต้องการเป็นเงื่อนไขในการค้นหา

#### 3.3.1.1.1 โครงสร้างของ propertyCollection

เป็นส่วนที่กำหนดข้อมูลของคุณสมบัติ เพื่อใช้เป็นเงื่อนไขในการค้นหา ดังแสดงในรูปที่ 3.20



รูปที่ 3.20 โครงสร้างของ propertyCollection

- อิลิเมนต์
  - name กำหนดชื่อของคุณสมบัติ
  - valueTypeNode กำหนดชนิดของลักษณะของค่าข้อมูล ดังแสดงในตารางที่ 3.2
  - valueNode กำหนดค่าของคุณสมบัติ
  - mode กำหนดคุณลักษณะความจำเป็นของค่าข้อมูล ดังแสดงในตารางที่ 3.1

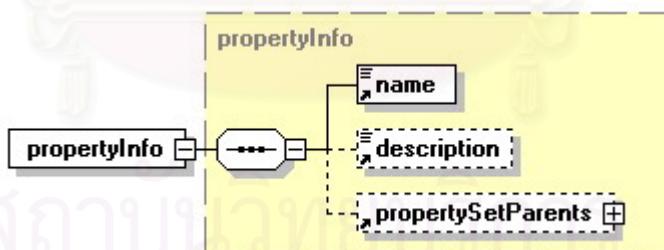
### 3.3.1.1.2 โครงสร้างของรายการคุณสมบัติ (propertyList)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการค้นหาคุณสมบัติ (find\_property) โดยมีส่วนประกอบดังรูปที่ 3.21



รูปที่ 3.21 โครงสร้างของ propertyList

ส่วนอิลิเมนต์ propertyInfos ประกอบด้วยชุดข้อมูลของคุณสมบัติ (propertyInfo) ดังรูปที่ 3.22

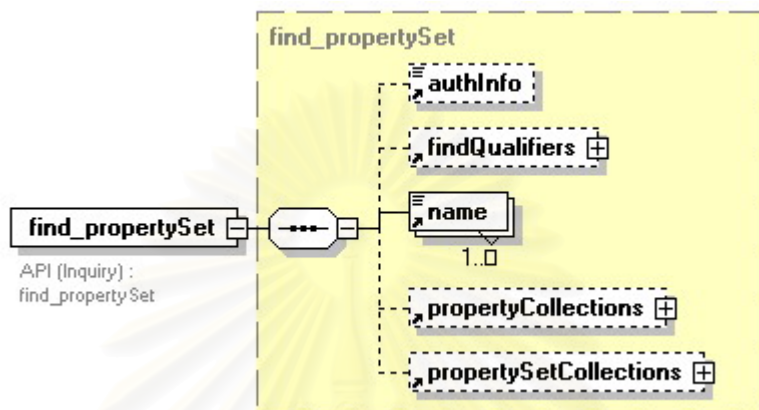


รูปที่ 3.22 โครงสร้างของ propertyInfo

- อิลิเมนต์
  - name ชื่อของคุณสมบัติ
  - description แสดงคำอธิบายคุณสมบัติ ซึ่งจะระบุหรือไม่ก็ได้
  - propertySetParents เป็นชุดเซตของคุณสมบัติ เพื่อแสดงว่าคุณสมบัตินี้อยู่ในเซตของคุณสมบัติใดบ้าง แสดงข้อมูลดังรูปที่ 3.17

### 3.3.1.2 การค้นหาเซตของคุณสมบัติ (find\_propertySet)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการค้นหาเซตของคุณสมบัติซึ่งได้ระบุเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.23 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายการเซตของคุณสมบัติ (propertySetList)

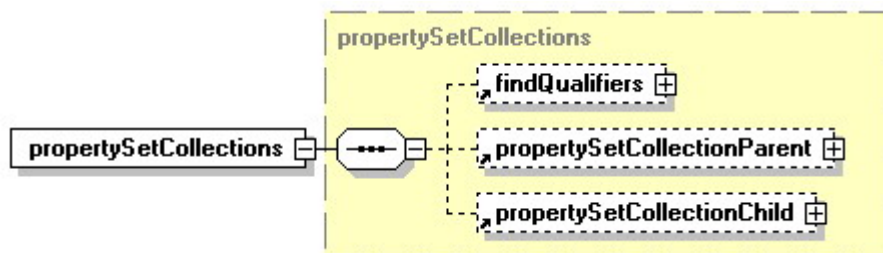


รูปที่ 3.23 โครงสร้างของ find\_propertySet

- แอตทริบิวต์
  - serviceKey ระบุรหัสประจำตัวของบริการที่ต้องการค้นหา ซึ่งจะระบุหรือไม่ก็ได้
- อิลิเมนต์
  - authinfo ข้อมูลของผู้ที่จะทำการค้นหา ซึ่งจะระบุหรือไม่ก็ได้
  - findQualifiers ระบุเงื่อนไขในการค้นข้อมูลตามรูปแบบของข้อกำหนดยูดีดีไอ ซึ่งจะระบุหรือไม่ก็ได้
  - name กำหนดชื่อของเซตของคุณสมบัติที่ต้องการค้นหา
  - propertyCollections ระบุชุดของคุณสมบัติเป็นเงื่อนไขในการค้นหา โดยคุณสมบัติ 1 ชุด คือ propertyCollection ดังแสดงในหัวข้อที่ 3.2.1.1.1
  - propertySetCollections ระบุชุดของเซตของคุณสมบัติเป็นเงื่อนไขในการค้นหา

#### 3.3.1.2.1 โครงสร้างของ propertySetCollections

เป็นส่วนที่กำหนดชุดพ่อแม่และลูกของเซตของคุณสมบัติ



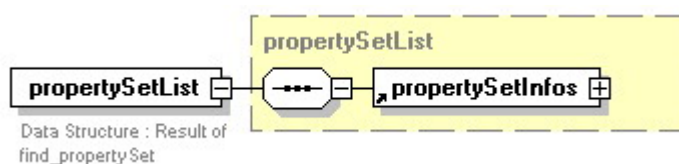
รูปที่ 3.24 โครงสร้างของ propertySetCollections

- อิลิเมนต์

- findQualifiers ใช้กำหนดเงื่อนไขในการค้นข้อมูลตามรูปแบบของข้อกำหนดยูดีดีไอ ซึ่งจะระบุหรือไม่ก็ได้
- propertySetCollectionParent ใช้กำหนดเซตของที่เป็นพ่อแม่ของเซตของที่สนใจ ซึ่งประกอบด้วย
  - อิลิเมนต์ name ชื่อเซตของพ่อแม่ซึ่งสามารถกำหนดได้หลายชื่อ
  - อิลิเมนต์ mode กำหนดคุณลักษณะความจำเป็นของเซตของพ่อแม่ ดังแสดงในตารางที่ 3.1
- propertySetCollectionChild ใช้กำหนดเซตของที่เป็นลูกของเซตของคุณสมบัติที่สนใจ ซึ่งประกอบด้วย
  - อิลิเมนต์ name ชื่อเซตของคุณสมบัติลูกซึ่งสามารถกำหนดได้หลายชื่อ
  - อิลิเมนต์ mode กำหนดคุณลักษณะความจำเป็นของเซตของคุณสมบัติลูก ดังแสดงในตารางที่ 3.1

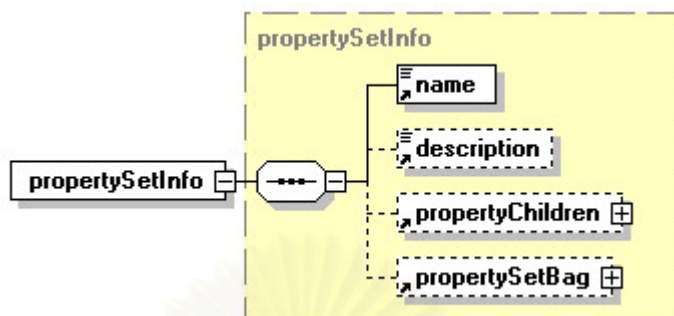
### 3.3.1.2.2 โครงสร้างของรายการเซตของคุณสมบัติ (propertySetList)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการค้นหาเซตของคุณสมบัติ (find\_propertySet) โดยมีส่วนประกอบดังรูปที่ 3.25



รูปที่ 3.25 โครงสร้างของ propertySetList

อิลิเมนต์ `propertySetInfos` ซึ่งเป็นชุดของเซตของคุณสมบัติ (`propertySetInfo`) มีส่วนประกอบดังรูปที่ 3.26

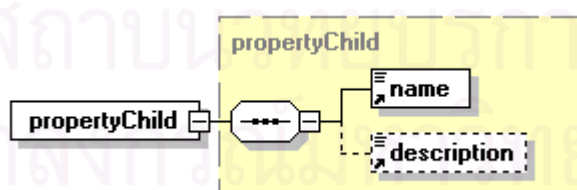


รูปที่ 3.26 โครงสร้างของ `propertySetInfo`

- อิลิเมนต์
  - `name` ชื่อของเซตของคุณสมบัติ
  - `description` แสดงคำอธิบายเซตของคุณสมบัติ
  - `propertyChildren` เป็นชุดคุณสมบัติที่ประกอบในเซตของคุณสมบัติ (`propertyChild`) แสดงข้อมูลดังรูปที่ 3.27
  - `propertySetBag` เป็นชุดเซตของคุณสมบัติ เพื่อแสดงว่าคุณสมบัตินี้ อยู่ในเซตของคุณสมบัติใดบ้าง (`propertySetFamily`) ซึ่งมี ส่วนประกอบตามหัวข้อที่ 3.2.2.11

### 3.3.1.2.3 โครงสร้างของ `propertyChild`

แสดงรายการคุณสมบัติ ดังรูปที่ 3.27

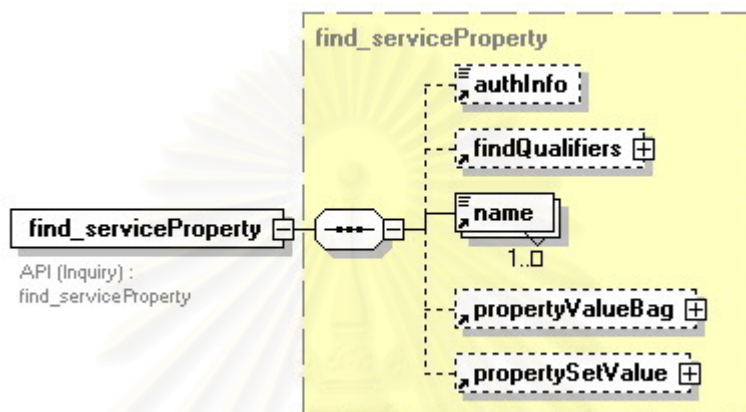


รูปที่ 3.27 โครงสร้างของ `propertyChild`

- อิลิเมนต์
  - `name` แสดงชื่อของคุณสมบัติ
  - `description` แสดงคำอธิบายของคุณสมบัติ

### 3.3.1.3 การค้นหาบริการโดยใช้คุณสมบัติบริการ (find\_serviceProperty)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการค้นหาบริการโดยใช้คุณสมบัติบริการเป็นเงื่อนไขในการค้นหา โดยมีส่วนประกอบดังรูปที่ 3.28 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายการบริการพร้อมกับคุณสมบัติบริการ (servicePropertyList)

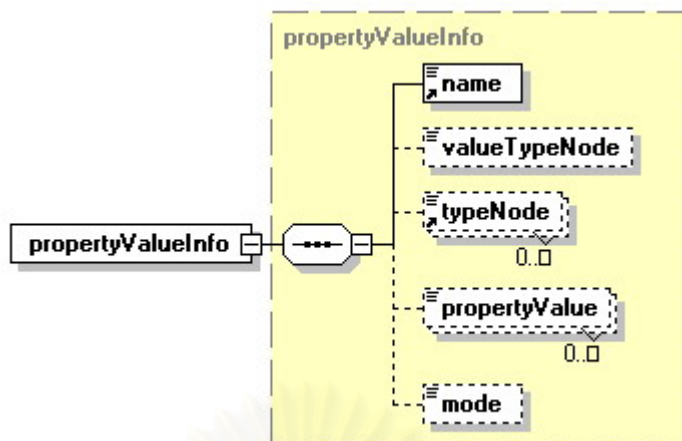


รูปที่ 3.28 โครงสร้างของ find\_serviceProperty

- แอตทริบิวต์
  - businessKey ระบุรหัสประจำตัวของธุรกิจที่ต้องการค้นหา ซึ่งจะระบุหรือไม่ก็ได้
- อิลิเมนต์
  - authinfo ข้อมูลของผู้ที่จะทำการค้นหา ซึ่งจะระบุหรือไม่ก็ได้
  - findQualifiers ใช้กำหนดเงื่อนไขในการค้นหาข้อมูลตามรูปแบบของข้อกำหนดยูดีดีไอ ซึ่งจะระบุหรือไม่ก็ได้
  - name ระบุชื่อของบริการที่ต้องการค้นหา ซึ่งสามารถระบุได้หลายชื่อ
  - propertyValueBag ระบุชุดของคุณสมบัติที่ต้องการค้นหา (propertyValueInfo) ซึ่งจะระบุหรือไม่ก็ได้
  - propertyValueInfo ระบุคุณสมบัติของบริการที่ต้องการค้นหา ซึ่งจะระบุหรือไม่ก็ได้

#### 3.3.1.3.1 โครงสร้างของ propertyValueInfo

แสดงคุณลักษณะของคุณสมบัติ ดังรูปที่ 3.29



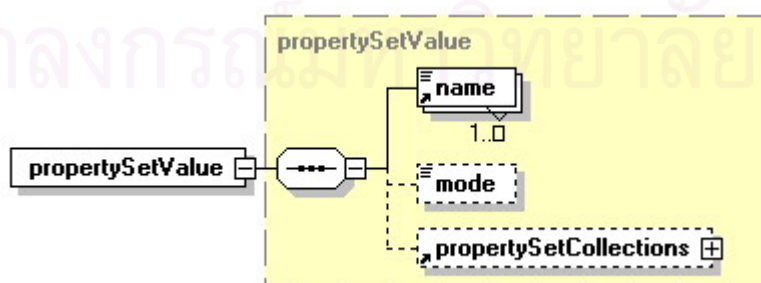
รูปที่ 3.29 โครงสร้างของ `propertyValueInfo`

- อิลิเมนต์

- `name` ระบุชื่อคุณสมบัติที่เป็นของบริการที่ต้องการค้นหา
- `valueTypeNode` ระบุชนิดของลักษณะของคุณสมบัติบริการ ซึ่งประกอบด้วยค่าดังแสดงในตารางที่ 3.2
- `typeNode` ระบุชนิดของข้อมูล ดังแสดงในตารางที่ 3.3 ซึ่งสามารถระบุได้หลายค่า
- `propertyValue` ระบุค่าของคุณสมบัติบริการที่ต้องการค้นหา ซึ่งสามารถระบุได้หลายค่า
- `mode` ระบุคุณลักษณะความจำเป็นของคุณสมบัติบริการ ดังแสดงในตารางที่ 3.1

### 3.3.1.3.2 โครงสร้างของ `propertySetValue`

แสดงคุณลักษณะของเซตของคุณสมบัติบริการที่ต้องการค้นหา ดังรูปที่ 3.30



รูปที่ 3.30 โครงสร้างของ `propertySetValue`

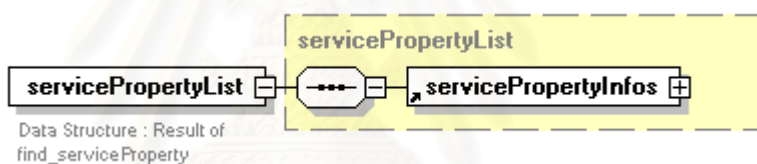


- อีลิเมนต์

- name ระบุชื่อเซตของคุณสมบัติบริการซึ่งบริการที่ต้องการค้นหาใช้  
โหมงมาอยู่ โดยจะสามารถระบุได้หลายชื่อ
- mode ระบุคุณลักษณะความจำเป็นของเซตของคุณสมบัติบริการ  
ดังแสดงในตารางที่ 3.1
- propertySetCollections ระบุชุดพ่อแม่และลูกของเซตของ  
คุณสมบัตินี้ ดังแสดงในหัวข้อที่ 3.2.1.2.1 ซึ่งจะระบุหรือไม่ก็ได้

### 3.3.1.3.3 โครงสร้างของรายการบริการพร้อมกับคุณสมบัติบริการ (servicePropertyList)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการค้นหา  
บริการโดยใช้คุณสมบัติบริการ (find\_servicePproperty) โดยมีส่วนประกอบดัง  
รูปที่ 3.31

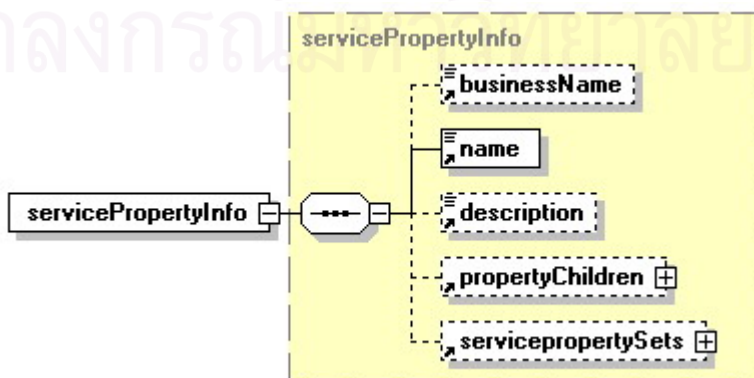


รูปที่ 3.31 โครงสร้างของ servicePropertyList

อีลิเมนต์ servicePropertyInfos ซึ่งเป็นชุดของข้อมูลของคุณสมบัติบริการ  
(servicePropertyInfo) มีส่วนประกอบดังรูปที่ 3.32

### 3.3.1.3.4 โครงสร้างของ servicePropertyInfo

แสดงข้อมูลเกี่ยวกับบริการที่ค้นหาพบ ดังรูปที่ 3.32

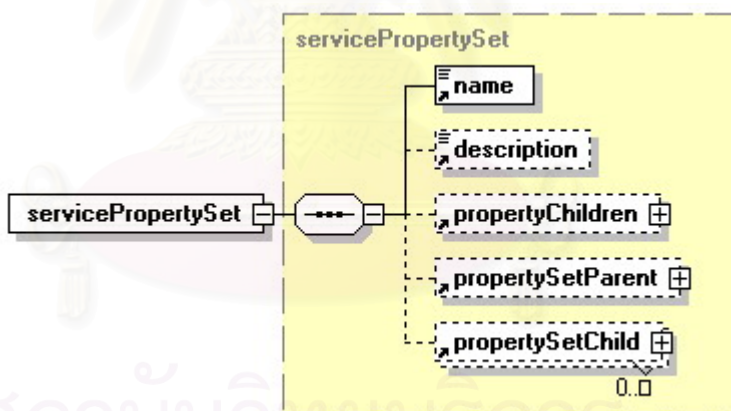


รูปที่ 3.32 โครงสร้างของ servicePropertyInfo

- แอตทริบิวต์
  - businessKey แสดงรหัสประจำตัวของธุรกิจ
  - serviceKey แสดงรหัสประจำตัวของบริการ
- อีลิเมนต์
  - businessName แสดงชื่อของธุรกิจ ซึ่งเป็นเจ้าของบริการที่ได้จากการค้นหา
  - name แสดงชื่อบริการที่ต้องการค้นหา
  - description แสดงคำอธิบายของบริการ
  - propertyChildren แสดงชุดคุณสมบัติบริการ (propertyChild) ตามหัวข้อที่ 3.2.1.2.3
  - servicePropertySets แสดงชุดคุณสมบัติที่ปรากฏในเซตของคุณสมบัติ servicePropertySet ดังรูปที่ 3.33

### 3.3.1.3.5 โครงสร้างของ servicePropertySet

แสดงรายการของเซตของคุณสมบัติของบริการที่ค้นหาได้



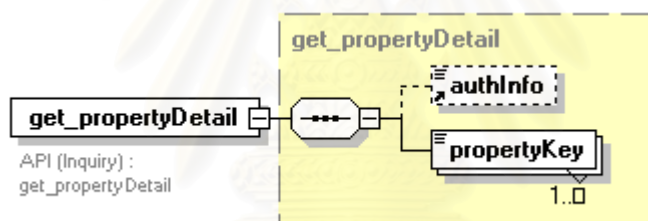
รูปที่ 3.33 โครงสร้างของ servicePropertySet

- แอตทริบิวต์
  - propertySetKey แสดงรหัสประจำตัวของเซตของคุณสมบัติ
  - mode ระบุคุณลักษณะความจำเป็นของเซตของคุณสมบัติบริการ ดังแสดงในตารางที่ 3.1
- อีลิเมนต์
  - name แสดงชื่อเซตของคุณสมบัติบริการ

- description แสดงคำอธิบายของเซตของคุณสมบัติบริการ
- propertyChildren แสดงชุดของคุณสมบัติบริการ (propertyChild) ซึ่งบรรจุอยู่ในเซตของคุณสมบัติบริการ ตามหัวข้อที่ 3.2.1.2.3
- propertySetParent แสดงเซตของคุณสมบัติของพ่อแม่คุณสมบัตินี้บริการ ตามหัวข้อที่ 3.2.2.12
- propertySetChild แสดงชุดเซตของคุณสมบัติของลูกคุณสมบัตินี้บริการ ตามหัวข้อที่ 3.2.2.13

#### 3.3.1.4 การเรียกดูรายละเอียดคุณสมบัตินี้ (get\_propertyDetail)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการเรียกดูรายละเอียดคุณสมบัตินี้โดยใช้รหัสประจำตัวคุณสมบัตินี้เป็นเงื่อนไขในการเรียกดูข้อมูล โดยมีส่วนประกอบดังรูปที่ 3.34 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายละเอียดคุณสมบัตินี้ (propertyDetail)

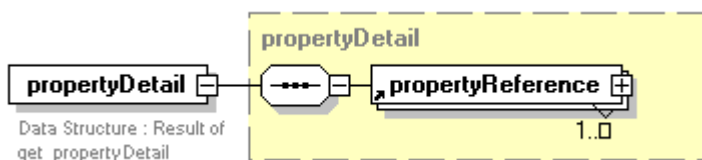


รูปที่ 3.34 โครงสร้างของ `get_propertyDetail`

- อิลิเมนต์
  - `authInfo` ข้อมูลของผู้ที่จะทำการเรียกดูข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - `propertyKey` ระบุรหัสประจำตัวคุณสมบัตินี้ที่ต้องการเรียกดูข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

##### 3.3.1.4.1 โครงสร้างของรายละเอียดคุณสมบัตินี้ (propertyDetail)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการเรียกดูรายละเอียดคุณสมบัตินี้ (`get_propertyDetail`) ตามรหัสประจำตัวคุณสมบัตินี้ที่ต้องการ โดยมีส่วนประกอบดังรูปที่ 3.35



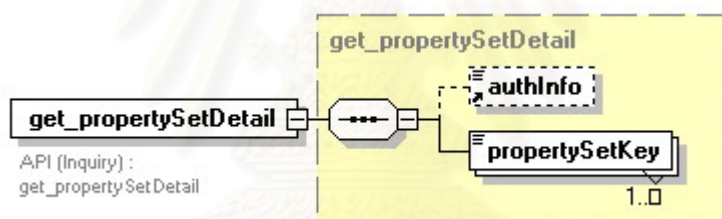
รูปที่ 3.35 โครงสร้างของ propertyDetail

- อิลิเมนต์

- propertyReference แสดงชุดรายละเอียดคุณลักษณะคุณสมบัติตามหัวข้อที่ 3.2.2.2

### 3.3.1.5 การเรียกดูรายละเอียดเซตของคุณสมบัติ (get\_propertySetDetail)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการเรียกดูรายละเอียดเซตของคุณสมบัติโดยใช้รหัสประจำตัวเซตของคุณสมบัติเป็นเงื่อนไขในการเรียกดูข้อมูลโดยมีส่วนประกอบดังรูปที่ 3.36 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายละเอียดคุณลักษณะเซตของคุณสมบัติ (propertySetDetail)



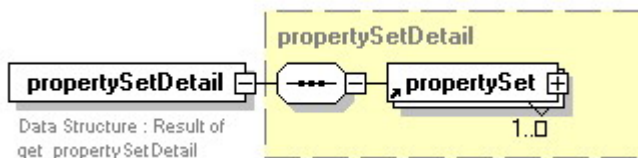
รูปที่ 3.36 โครงสร้างของ get\_propertySetDetail

- อิลิเมนต์

- authinfo ข้อมูลของผู้ที่จะทำการเรียกดูข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
- propertySetKey ระบุรหัสประจำตัวเซตของคุณสมบัติที่ต้องการเรียกดูข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

#### 3.3.1.5.1 โครงสร้างของรายละเอียดคุณลักษณะของเซตของคุณสมบัติ (propertySetDetail)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการเรียกดูรายละเอียดเซตของคุณสมบัติ (get\_propertySetDetail) ตามรหัสประจำตัวเซตของคุณสมบัติที่ต้องการ โดยมีส่วนประกอบดังรูปที่ 3.37

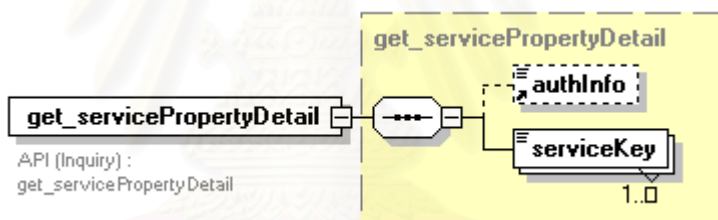


รูปที่ 3.37 โครงสร้างของ propertySetDetail

- อิลิเมนต์
  - propertySet แสดงชุดรายละเอียดคุณลักษณะของเซตของคุณสมบัติ ตามหัวข้อที่ 3.2.2.9

### 3.3.1.6 การเรียกดูรายละเอียดบริการพร้อมคุณสมบัติ (get\_servicePropertyDetail)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการเรียกดูรายละเอียดบริการพร้อมคุณสมบัติโดยใช้รหัสประจำตัวของบริการเงื่อนไขในการเรียกดูข้อมูลโดยมีส่วนประกอบดังรูปที่ 3.38 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของรายละเอียดบริการพร้อมคุณสมบัติ (servicePropertyDetail)

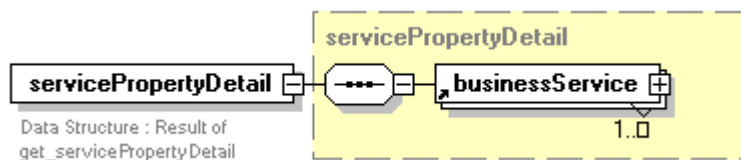


รูปที่ 3.38 โครงสร้างของ get\_servicePropertyDetail

- อิลิเมนต์
  - authInfo ข้อมูลของผู้ที่จะทำการเรียกดูข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - serviceKey ระบุรหัสประจำตัวบริการที่ต้องการเรียกดูข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

#### 3.3.1.6.1 โครงสร้างของรายละเอียดบริการพร้อมคุณสมบัติ (servicePropertyDetail)

เป็นผลลัพธ์ที่ได้คืนจากการเรียกใช้งานส่วนต่อประสานโปรแกรมการเรียกดูรายละเอียดบริการพร้อมคุณสมบัติ (get\_servicePropertyDetail) ตามรหัสประจำตัวบริการที่ต้องการ โดยมีส่วนประกอบดังรูปที่ 3.39



รูปที่ 3.39 โครงสร้างของ servicePropertyDetail

- อีลิเมนต์
  - businessService แสดงชุดรายละเอียดบริการพร้อมคุณสมบัติ ตามหัวข้อที่ 3.2.1

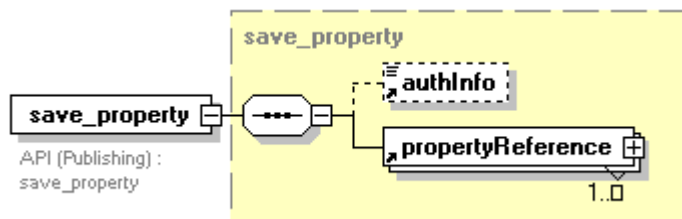
### 3.3.2 การขยายส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณา (Publishing API Function)

เพื่อให้สามารถทำการโฆษณาคุณสมบัติบริการที่เพิ่มเติมขึ้นมาไว้ในยูทิลิตี้โอ รีจิสตรี จึงได้ขยายโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรมให้ผู้ให้บริการสามารถทำการโฆษณาบริการและเพิ่มข้อมูลคุณสมบัติของบริการได้ ดังนี้

- save\_property ผลลัพธ์ที่ได้คือ save\_property
- save\_propertySet ผลลัพธ์ที่ได้คือ save\_propertySet
- save\_serviceProperty ผลลัพธ์ที่ได้คือ save\_serviceProperty
- delete\_property
- delete\_propertySet
- delete\_serviceProperty

#### 3.3.2.1 การโฆษณาคุณสมบัติ (save\_property)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการโฆษณาคุณสมบัติไว้ในยูทิลิตี้โอ รีจิสตรี โดยมีส่วนประกอบดังรูปที่ 3.40 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของ save\_property ซึ่งถ้าเป็นการโฆษณาคุณสมบัติใหม่ จะมีรหัสประจำตัวคุณสมบัติเพิ่มเติมเข้ามา

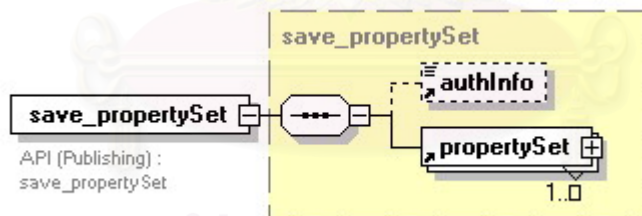


รูปที่ 3.40 โครงสร้างของ save\_property

- อิลิเมนต์
  - authinfo ข้อมูลของผู้ที่จะทำการโฆษณาข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - propertyReference ระบุรายละเอียดของคุณสมบัติที่ต้องการโฆษณา ตามหัวข้อที่ 3.2.2.2 ซึ่งสามารถทำการระบุได้มากกว่า 1 คุณสมบัตินี้

### 3.3.2.2 การโฆษณาเซตของคุณสมบัติ (save\_propertySet)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการโฆษณาเซตของคุณสมบัติไว้ในยูทิลิตี้โอ รี่จิสตรี โดยมีส่วนประกอบดังรูปที่ 3.41 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของ save\_propertySet ซึ่งถ้าเป็นการโฆษณาเซตของคุณสมบัติใหม่ จะมีรหัสประจำตัวเซตของคุณสมบัติเพิ่มเติมเข้ามา

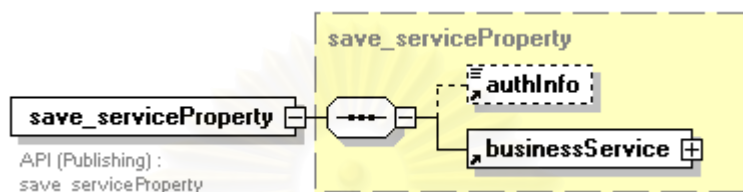


รูปที่ 3.41 โครงสร้างของ save\_propertySet

- อิลิเมนต์
  - authinfo ข้อมูลของผู้ที่จะทำการโฆษณาข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - propertySet ระบุรายละเอียดของเซตของคุณสมบัติที่ต้องการโฆษณา ตามหัวข้อที่ 3.2.2.9 ซึ่งสามารถทำการระบุได้มากกว่า 1 เซตของคุณสมบัติ

### 3.3.2.3 การโฆษณาบริการพร้อมคุณสมบัติ (save\_serviceProperty)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการโฆษณาบริการพร้อมคุณสมบัติไว้ในยูดีดีไอ รีจิสตรี โดยมีส่วนประกอบดังรูปที่ 3.42 ผลลัพธ์ที่คืนกลับมาจะอยู่ในโครงสร้างของ save\_serviceProperty ซึ่งถ้าเป็นการโฆษณาบริการใหม่ จะมีรหัสประจำตัวบริการเพิ่มเติมเข้ามา

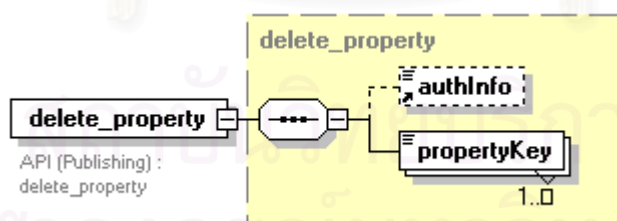


รูปที่ 3.42 โครงสร้างของ save\_serviceProperty

- อิลิเมนต์
  - `authInfo` ข้อมูลของผู้ที่จะทำการโฆษณาข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - `businessService` ระบุรายละเอียดของบริการพร้อมคุณสมบัติที่ต้องการโฆษณา ตามหัวข้อที่ 3.2.1

### 3.3.2.4 การลบคุณสมบัติ (delete\_property)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการลบคุณสมบัติโดยใช้รหัสประจำตัวคุณสมบัติเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.43



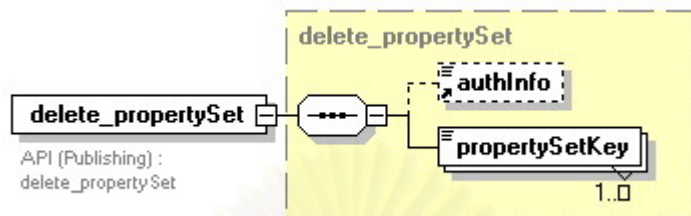
รูปที่ 3.43 โครงสร้างของ delete\_property

- อิลิเมนต์
  - `authInfo` ข้อมูลของผู้ที่จะทำการลบข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - `propertyKey` ระบุรหัสประจำตัวคุณสมบัติที่ต้องการลบข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า



### 3.3.2.5 การลบเซตของคุณสมบัติ (delete\_propertySet)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการลบเซตของคุณสมบัติ โดยใช้รหัสประจำตัวเซตของคุณสมบัติเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.44

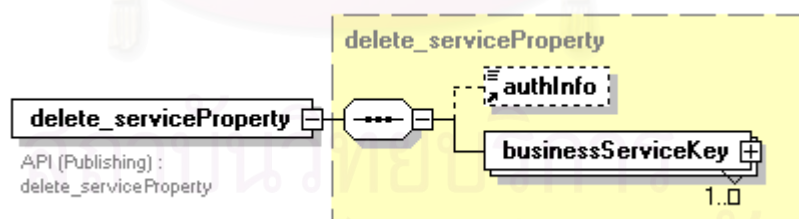


รูปที่ 3.44 โครงสร้างของ delete\_propertySet

- อิลิเมนต์
  - authInfo ข้อมูลของผู้ที่จะทำการลบข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - propertySetKey ระบุรหัสประจำตัวเซตของคุณสมบัติที่ต้องการลบข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

### 3.3.2.6 การลบคุณสมบัติของบริการ (delete\_serviceProperty)

เป็นโครงสร้างข้อมูลของส่วนต่อประสานโปรแกรม ที่ใช้ในการลบคุณสมบัติของบริการ โดยใช้รหัสประจำตัวบริการและคุณสมบัตินั้นเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.45

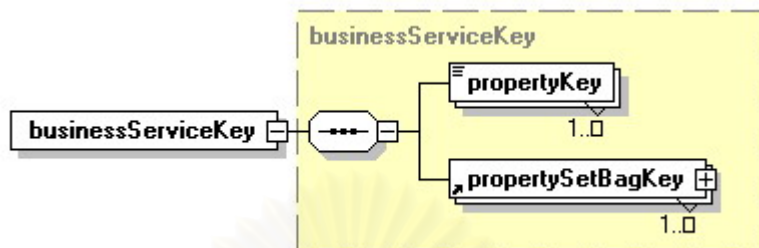


รูปที่ 3.45 โครงสร้างของ delete\_serviceProperty

- อิลิเมนต์
  - authInfo ข้อมูลของผู้ที่จะทำการลบข้อมูล ซึ่งจะระบุหรือไม่ก็ได้
  - businessServiceKey ระบุข้อมูลรหัสประจำตัวบริการ ตามโครงสร้างดังรูปที่ 3.46 ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

### 3.3.2.7 โครงสร้างของ businessServiceKey

เป็นส่วนที่ระบุบริการพร้อมคุณสมบัติ โดยใช้รหัสประจำตัวบริการและคุณสมบัติเป็นเงื่อนไข โดยมีส่วนประกอบดังรูปที่ 3.46



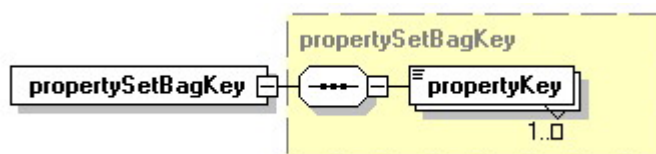
รูปที่ 3.46 โครงสร้างของ businessServiceKey

- แอตทริบิวต์
  - serviceKey ระบุรหัสประจำตัวบริการที่ต้องการใช้เป็นเงื่อนไขในการลบข้อมูลคุณสมบัติ
- อิลิเมนต์
  - propertyKey ระบุรหัสประจำตัวคุณสมบัติของบริการที่ต้องการลบข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า
  - propertySetBagKey ระบุข้อมูลรหัสประจำตัวเซตของคุณสมบัติบริการ ตามโครงสร้างดังรูปที่ 3.47 ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

### 3.3.2.8 โครงสร้างของ propertySetBagKey

เป็นส่วนที่ระบุเซตของคุณสมบัติบริการ โดยจะทำการลบข้อมูลเซตของคุณสมบัติหรือคุณสมบัติที่บรรจุในเซตของคุณสมบัติ โดยมีส่วนประกอบดังรูปที่ 3.47

- หากระบุเฉพาะรหัสประจำตัวเซตของคุณสมบัติ จะเป็นการลบเซตของคุณสมบัติที่ระบุออกจากบริการที่กำหนด
- หากระบุรหัสประจำตัวเซตของคุณสมบัติ และรหัสประจำตัวคุณสมบัติ จะเป็นการลบคุณสมบัติที่บรรจุในเซตของคุณสมบัติของบริการที่กำหนด



รูปที่ 3.47 โครงสร้างของ propertySetBagKey

- แอตทริบิวท์
  - propertySetKey ระบุรหัสประจำตัวเซตของคุณสมบัติที่ต้องการใช้เป็นเงื่อนไขในการลบข้อมูลคุณสมบัติ
- อีลิเมนต์
  - propertyKey ระบุรหัสประจำตัวคุณสมบัตินี้บรรจุในเซตของคุณสมบัติของบริการที่ต้องการลบข้อมูล ซึ่งสามารถทำการระบุได้มากกว่า 1 ค่า

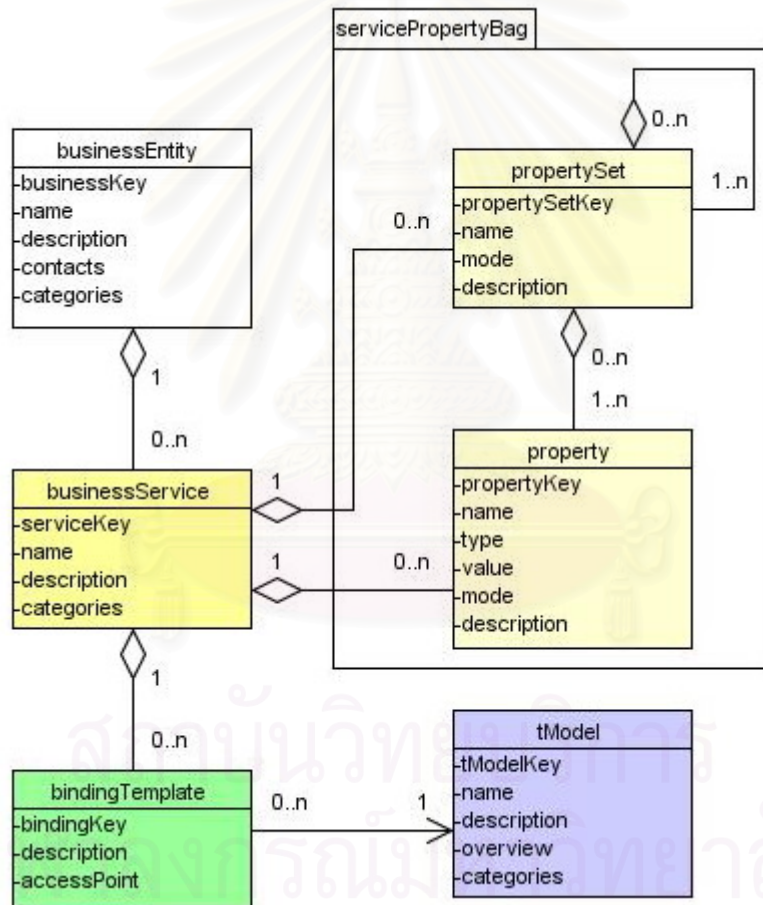


สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 4

### ต้นแบบส่วนขยายยูดีดีไอ

บทนี้จะกล่าวถึงการพัฒนาส่วนขยายยูดีดีไอ ให้สนับสนุนการโฆษณาคุณสมบัติของบริการ รวมทั้งการใช้คุณสมบัติของบริการที่เพิ่มเติมเป็นเงื่อนไขในการค้นหา ตามโครงสร้างข้อมูลและส่วนต่อประสานโปรแกรมที่กล่าวในบทที่ 3 โดยแนวคิดในวิทยานิพนธ์นี้คือ คุณสมบัติของบริการนั้นอาจจะถูกกำหนดเป็นคุณสมบัติเฉพาะของแต่ละบริการ หรือกำหนดเป็นชุดคุณสมบัติ โดยบรรจุอยู่ในเซตของคุณสมบัติ ดังนั้นบริการที่ใช้ส่วนขยายยูดีดีไออาจจะโฆษณาโดยใช้คุณสมบัติที่กำหนดเฉพาะตัว หรือใช้เซตของคุณสมบัติ หรือใช้ทั้งสองอย่างก็ได้ ดังรูปที่ 4.1

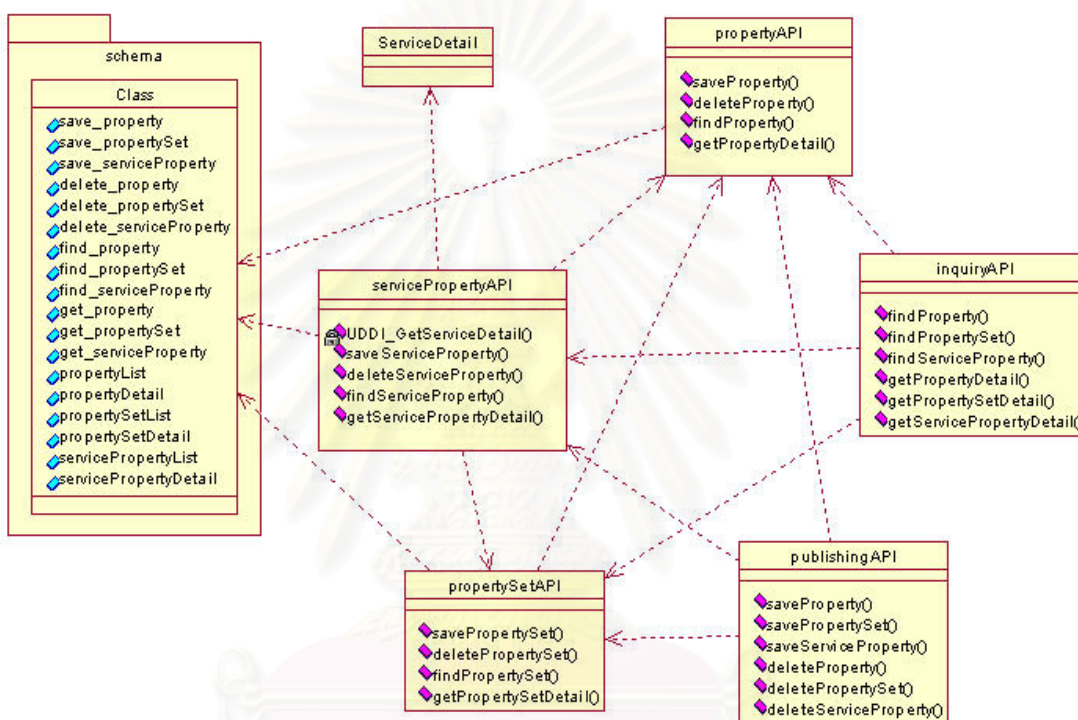


รูปที่ 4.1 โครงสร้างข้อมูลของบริการในยูดีดีไอที่แสดงคุณสมบัติและเซตของคุณสมบัติ

#### 4.1 คลาสหลักต้นแบบส่วนขยายยูดีดีไอ

ในบทนี้จะนำโครงสร้างจากรูปที่ 4.1 มาพัฒนาเป็นคลาสหลักต้นแบบของส่วนขยายยูดีดีไอ ดังนี้ (รูปที่ 4.2)

- schema
- propertyAPI
- propertySetAPI
- servicePropertyAPI
- inquiryAPI
- publishingAPI



รูปที่ 4.2 คลาสหลักต้นแบบส่วนขยายยูดีดีไอ

4.1.1 แพคเกจ schema เป็นคลาสต่างๆ ของโครงสร้างข้อมูลหลักของส่วนขยายยูดีดีไอและส่วนต่อประสานโปรแกรม ตามโครงสร้างที่อธิบายในบทที่ 3

4.1.2 คลาส propertyAPI เป็นส่วนต่อประสานโปรแกรมที่มีตัวกระทำเกี่ยวกับคุณสมบัติบริการ ซึ่งประกอบด้วย การโฆษณาคุณสมบัติและการค้นหาคุณสมบัติ

4.1.3 คลาส propertySetAPI เป็นส่วนต่อประสานโปรแกรมที่มีตัวกระทำเกี่ยวกับเซตของคุณสมบัติบริการ ซึ่งประกอบด้วย การโฆษณาเซตของคุณสมบัติและการค้นหาเซตของคุณสมบัติ

4.1.4 คลาส servicePropertyAPI เป็นส่วนต่อประสานโปรแกรมที่มีตัวกระทำเกี่ยวกับบริการที่ใช้ประโยชน์จากส่วนขยายยูดีดีไอ ซึ่งประกอบด้วย การโฆษณาบริการพร้อมคุณสมบัติ และการค้นหาบริการโดยใช้คุณสมบัติเป็นเงื่อนไขในการค้นหา

4.1.5 คลาส inquiryAPI เป็นส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา เพื่อให้สามารถเรียกใช้งานผ่านเว็บเซอร์วิสได้ ซึ่งส่วนนี้เป็นส่วนที่ติดต่อกับผู้ใช้งาน โดยผู้ใช้งานสามารถเขียนโปรแกรมเรียกใช้งานได้ตามข้อกำหนดของเว็บเซอร์วิส

4.1.6 คลาส publishingAPI เป็นส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณา เพื่อให้สามารถเรียกใช้งานผ่านเว็บเซอร์วิสได้ ซึ่งส่วนนี้เป็นส่วนที่ติดต่อกับผู้ใช้งาน โดยผู้ใช้งานสามารถเขียนโปรแกรมเรียกใช้งานได้ตามข้อกำหนดของเว็บเซอร์วิส

## 4.2 ส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา

ในที่นี้จะอธิบายถึงคลาส inquiryAPI ซึ่งเป็นส่วนต่อประสานโปรแกรมที่ให้ผู้ใช้งานสามารถทำการเขียนโปรแกรมเรียกใช้งานโดยผ่านเว็บเซอร์วิส เพื่อทำการค้นหาบริการโดยสามารถใช้คุณสมบัติบริการเป็นเงื่อนไขในการค้นหาข้อมูลได้ ประกอบด้วยตัวกระทำการดังนี้

### 4.2.1 ตัวกระทำการ findProperty

เป็นตัวกระทำการที่ทำหน้าที่ในการค้นหาคุณสมบัติบริการ โดยมีรายละเอียดดังนี้

```
public propertyList findProperty(find_property find_pD, out string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- find\_pD คือโครงสร้างข้อมูลแบบ find\_property ตามหัวข้อที่ 3.3.1.1 ซึ่งเป็นเงื่อนไขของคุณสมบัติที่ต้องการค้นหา
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำการ

ค่าคืนกลับของตัวกระทำการ ประกอบด้วย

- propertyList เป็นโครงสร้างรายการข้อมูลของคุณสมบัติ ตามหัวข้อที่ 3.3.1.1.2

### 4.2.2 ตัวกระทำการ findPropertySet

เป็นตัวกระทำการที่ทำหน้าที่ในการค้นหาเซตของคุณสมบัติบริการ โดยมีรายละเอียดดังนี้

```
public propertySetList findPropertySet(find_propertySet find_pT, out string
po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- find\_pT คือโครงสร้างข้อมูลแบบ find\_propertySet ตามหัวข้อที่ 3.3.1.2 ซึ่งเป็นเงื่อนไขของเซตของคุณสมบัติที่ต้องการค้นหา
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- propertySetList เป็นโครงสร้างรายการข้อมูลของเซตของคุณสมบัติตามหัวข้อที่ 3.3.1.1.2

#### 4.2.3 ตัวกระทำ findServiceProperty

เป็นตัวกระทำที่ทำหน้าที่ในการค้นหาบริการโดยระบุคุณสมบัติเป็นเงื่อนไขในการค้นหา โดยมีรายละเอียดดังนี้

```
public servicePropertyList findServiceProperty(find_serviceProperty find_sP, out
string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- find\_sP คือโครงสร้างข้อมูลแบบ find\_serviceProperty ตามหัวข้อที่ 3.3.1.3 ซึ่งเป็นเงื่อนไขของบริการพร้อมคุณสมบัติที่ต้องการค้นหา
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- servicePropertyList เป็นโครงสร้างรายการข้อมูลของบริการพร้อมคุณสมบัติ ตามหัวข้อที่ 3.3.1.3.3

#### 4.2.4 ตัวกระทำ getPropertyDetail

เป็นตัวกระทำที่ทำหน้าที่ในการเรียกดูรายละเอียดคุณสมบัติ โดยมีรายละเอียดดังนี้

```
public propertyDetail getPropertyDetail(get_propertyDetail get_pD, out string
po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- get\_pD คือโครงสร้างข้อมูลแบบ get\_propertyDetail ตามหัวข้อที่ 3.3.1.4 ซึ่งเป็นเงื่อนไขของคุณสมบัติที่ต้องการเรียกดู
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- propertyDetail เป็นโครงสร้างรายละเอียดข้อมูลของคุณสมบัติ ตามหัวข้อที่ 3.3.1.4.1

#### 4.2.5 ตัวกระทำ getPropertySetDetail

เป็นตัวกระทำที่ทำหน้าที่ในการเรียกดูรายละเอียดเซตของคุณสมบัติ โดยมีรายละเอียดดังนี้

```
public propertySetDetail getPropertySetDetail (get_propertySetDetail get_pT, out
string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- get\_pT คือโครงสร้างข้อมูลแบบ get\_propertySetDetail ตามหัวข้อที่ 3.3.1.5 ซึ่งเป็นเงื่อนไขของเซตของคุณสมบัติที่ต้องการเรียกดู
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- propertySetDetail เป็นโครงสร้างรายละเอียดข้อมูลของเซตของคุณสมบัติ ตามหัวข้อที่ 3.3.1.5.1

#### 4.2.6 ตัวกระทำ getServicePropertyDetail

เป็นตัวกระทำที่ทำหน้าที่ในการเรียกดูรายละเอียดของบริการพร้อมคุณสมบัติ โดยมีรายละเอียดดังนี้



```
public servicePropertyDetail getServicePropertyDetail (get_servicePropertyDetail
get_sP, out string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- get\_sP คือโครงสร้างข้อมูลแบบ get\_servicePropertyDetail ตามหัวข้อที่ 3.3.1.6 ซึ่งเป็นเงื่อนไขของบริการพร้อมคุณสมบัติต้องการเรียกดู
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- servicePropertyDetail เป็นโครงสร้างรายละเอียดข้อมูลของบริการพร้อมคุณสมบัติ ตามหัวข้อที่ 3.3.1.6.1

#### 4.3 ส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการโฆษณา

ในที่นี้จะอธิบายถึงคลาส publishingAPI ซึ่งเป็นส่วนต่อประสานโปรแกรมที่ให้ผู้ใช้งานสามารถทำการเขียนโปรแกรมเรียกใช้งานโดยผ่านเว็บเซอวิส เพื่อทำการโฆษณาบริการพร้อมคุณสมบัติได้ ประกอบด้วยตัวกระทำดังนี้

##### 4.3.1 ตัวกระทำการ saveProperty

เป็นตัวกระทำที่ทำหน้าที่ในการโฆษณาคุณสมบัติบริการ โดยมีรายละเอียดดังนี้

```
public save_property saveProperty(save_property save_sP, out string
po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- save\_sP คือโครงสร้างข้อมูลแบบ save\_property ตามหัวข้อที่ 3.3.2.1 ซึ่งเป็นโครงสร้างรายละเอียดข้อมูลคุณสมบัติ
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- `save_property` เป็นโครงสร้างข้อมูลของคุณสมบัติ ที่มีรหัสประจำตัว  
คุณสมบัติ

#### 4.3.2 ตัวกระทำการ `savePropertySet`

เป็นตัวกระทำการที่ทำหน้าที่ในการโฆษณาเซตของคุณสมบัติบริการ โดยมี  
รายละเอียดดังนี้

```
public save_propertySet savePropertySet(save_propertySet save_pT, out string  
po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- `save_pT` คือโครงสร้างข้อมูลแบบ `save_propertySet` ตามหัวข้อที่  
3.3.2.2 ซึ่งเป็นโครงสร้างรายละเอียดข้อมูลเซตของคุณสมบัติ
- `po_returnMsg` คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึง  
ความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำการ

ค่าคืนกลับของตัวกระทำการ ประกอบด้วย

- `save_propertySet` เป็นโครงสร้างข้อมูลของเซตของคุณสมบัติ ที่มีรหัส  
ประจำตัวเซตของคุณสมบัติ

#### 4.3.3 ตัวกระทำการ `saveServiceProperty`

เป็นตัวกระทำการที่ทำหน้าที่ในการโฆษณาบริการพร้อมคุณสมบัติบริการ โดยมี  
รายละเอียดดังนี้

```
public save_serviceProperty saveServiceProperty(save_serviceProperty save_sP,  
out string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- `save_sP` คือโครงสร้างข้อมูลแบบ `save_serviceProperty` ตามหัวข้อที่  
3.3.2.3 ซึ่งเป็นโครงสร้างรายละเอียดข้อมูลของบริการและคุณสมบัติ
- `po_returnMsg` คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึง  
ความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำการ

ค่าคืนกลับของตัวกระทำการ ประกอบด้วย

- `save_serviceProperty` เป็นโครงสร้างรายละเอียดข้อมูลของบริการและคุณสมบัติ ซึ่งในที่นี้คุณสมบัติและเซตของคุณสมบัติจะต้องแสดงรหัสประจำตัวด้วย

#### 4.3.4 ตัวกระทำการ `deleteProperty`

เป็นตัวกระทำที่ทำหน้าที่ในการลบคุณสมบัติบริการ โดยมีรายละเอียดดังนี้

```
public bool deleteProperty(delete_property delete_pD, out string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- `delete_pD` คือโครงสร้างข้อมูลแบบ `delete_property` ตามหัวข้อที่ 3.3.2.4 ซึ่งเป็นโครงสร้างเงื่อนไขการลบคุณสมบัติ
- `po_returnMsg` คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำการ

ค่าคืนกลับของตัวกระทำการ ประกอบด้วย

- `bool` เป็นค่าตรรกะ ถ้าตัวกระทำการทำงานสำเร็จจะคืนค่าจริง (`true`) ถ้าเกิดข้อผิดพลาดจะคืนค่าเท็จ (`false`)

#### 4.3.5 ตัวกระทำการ `deletePropertySet`

เป็นตัวกระทำที่ทำหน้าที่ในการลบเซตของคุณสมบัติบริการ โดยมีรายละเอียดดังนี้

```
public bool deletePropertySet(delete_propertySet delete_pT, out string po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- `delete_pT` คือโครงสร้างข้อมูลแบบ `delete_propertySet` ตามหัวข้อที่ 3.3.2.5 ซึ่งเป็นโครงสร้างเงื่อนไขการลบเซตของคุณสมบัติ
- `po_returnMsg` คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำการ

ค่าคืนกลับของตัวกระทำการ ประกอบด้วย

- bool เป็นค่าตรรกะ ถ้าตัวกระทำการทำงานสำเร็จจะเป็นค่าจริง (true) ถ้าเกิดข้อผิดพลาดจะเป็นค่าเท็จ (false)

#### 4.3.6 ตัวกระทำการ deleteServiceProperty

เป็นตัวกระทำที่ทำหน้าที่ในการลบคุณสมบัติของบริการที่ระบุ โดยมีรายละเอียดดังนี้

```
public bool deleteServiceProperty(delete_serviceProperty delete_sP, out string
po_returnMsg)
```

พารามิเตอร์ ประกอบด้วย

- delete\_sP คือโครงสร้างข้อมูลแบบ delete\_serviceProperty ตามหัวข้อที่ 3.3.2.6 ซึ่งเป็นโครงสร้างเงื่อนไขการลบคุณสมบัติของบริการที่โฆษณาให้กับบริการที่ระบุ
- po\_returnMsg คือข้อมูลที่คืนค่ากลับมา เป็นชุดตัวอักษรที่จะแสดงถึงความสำเร็จหรือข้อผิดพลาดของการทำงานของตัวกระทำ

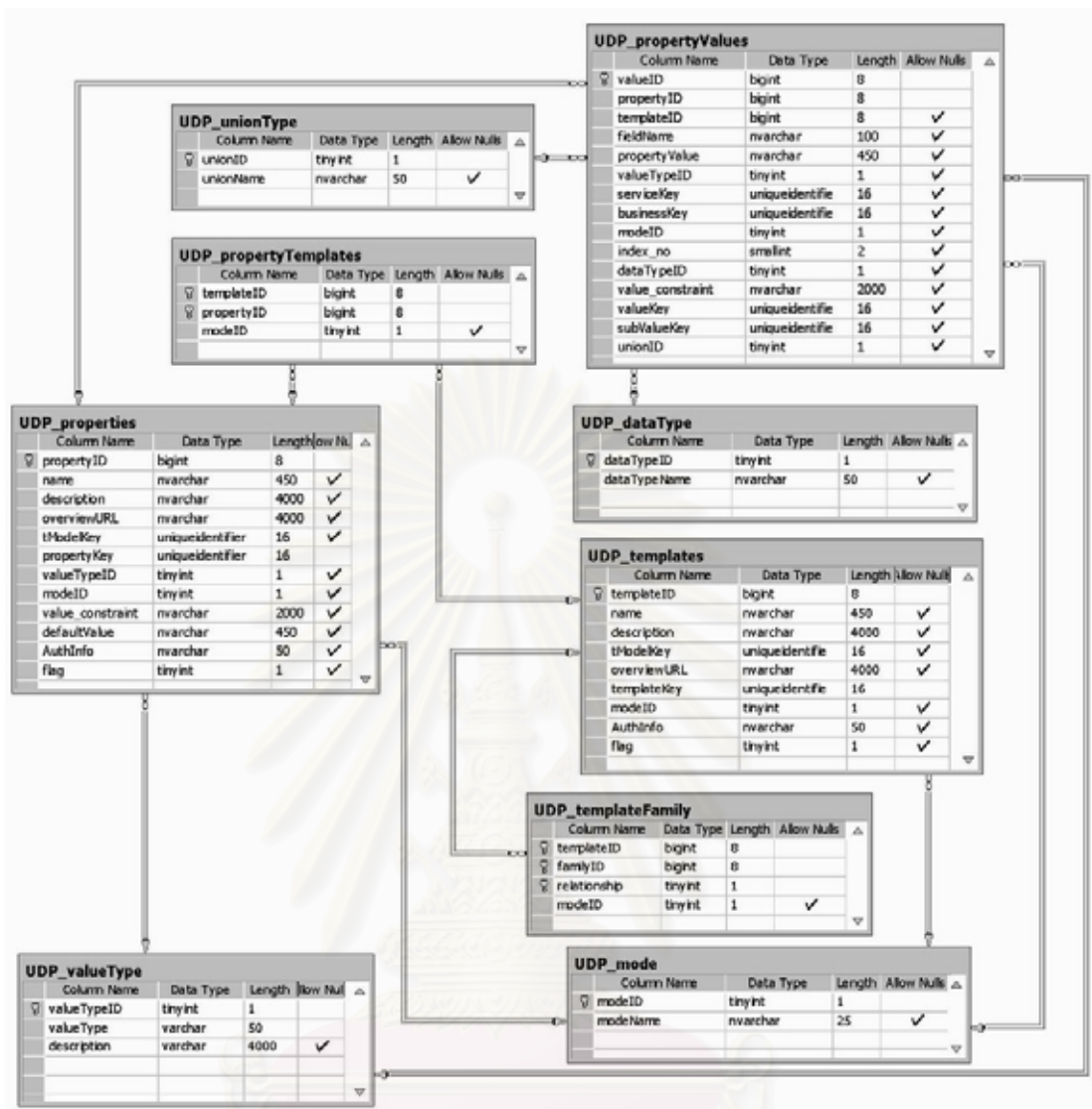
ค่าคืนกลับของตัวกระทำ ประกอบด้วย

- bool เป็นค่าตรรกะ ถ้าตัวกระทำการทำงานสำเร็จจะเป็นค่าจริง (true) ถ้าเกิดข้อผิดพลาดจะเป็นค่าเท็จ (false)

#### 4.4 ฐานข้อมูลส่วนขยายยูดีดีไอ

การพัฒนาฐานข้อมูลเพื่อสนับสนุนส่วนขยายยูดีดีไอนั้น ไม่ได้ทำการแก้ไขในส่วนของฐานข้อมูลยูดีดีไอ ริจิสตรีเดิม แต่ได้ทำการเพิ่มตารางที่ใช้สำหรับการเก็บข้อมูลส่วนขยายยูดีดีไอดังรูปที่ 4.3

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.3 ฐานข้อมูลส่วนขยายยูดีดีไอ

ส่วนขยายยูดีดีไอ ประกอบด้วยตารางต่างๆ สำหรับฐานข้อมูลยูดีดีไอ ริจิสตรี ดังนี้

- UDP\_properties เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายละเอียดคุณสมบัติ
- UDP\_propertyValues เป็นตารางที่เก็บข้อมูลเกี่ยวกับคุณสมบัติ พร้อมค่าข้อมูลคุณสมบัติของบริการต่างๆ
- UDP\_templates เป็นตารางที่เก็บข้อมูลเกี่ยวกับรายละเอียดเซตของคุณสมบัติ
- UDP\_propertyTemplates เป็นตารางที่เก็บข้อมูลเกี่ยวกับคุณสมบัติแต่ละตัวว่าบรรจุอยู่ในเซตของคุณสมบัติใด

- UDP\_templateFamily เป็นตารางที่เก็บข้อมูลเกี่ยวกับเซตของคุณสมบัติว่าเซตของคุณสมบัติใดมีความสัมพันธ์เป็นระดับพ่อหรือระดับลูกของเซตของคุณสมบัติใด
- UDP\_mode เป็นตารางที่เก็บข้อมูลเกี่ยวกับคุณลักษณะความจำเป็นของข้อมูล ตามตารางที่ 3.1
- UDP\_valueType เป็นตารางที่เก็บข้อมูลเกี่ยวกับลักษณะของข้อมูลตามตารางที่ 3.2
- UDP\_dataType เป็นตารางที่เก็บข้อมูลเกี่ยวกับชนิดของข้อมูล ตามตารางที่ 3.3

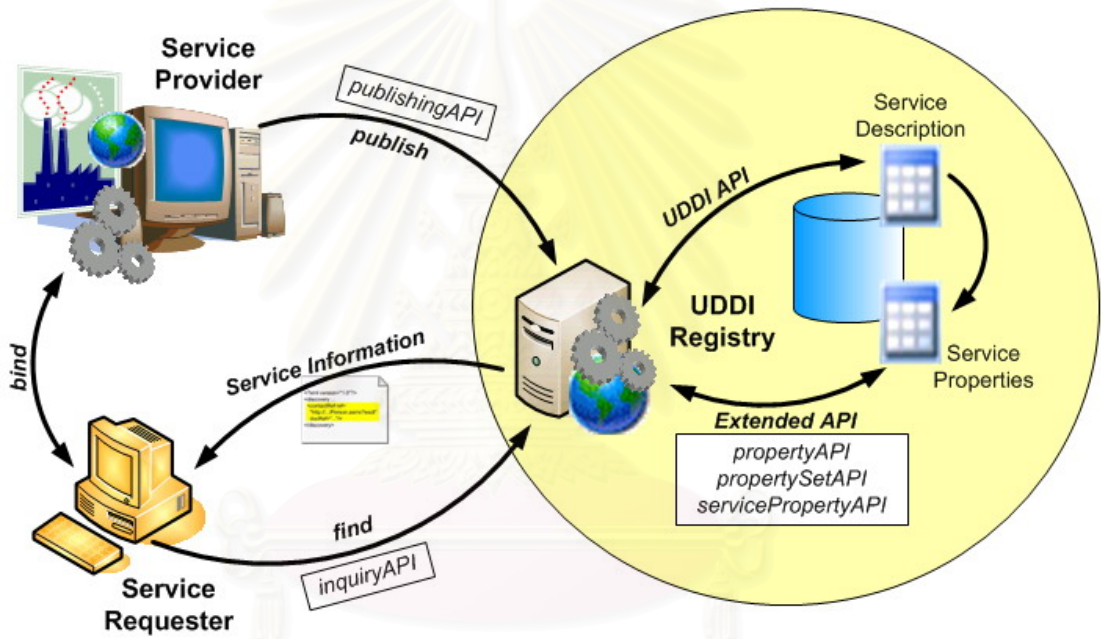
#### 4.5 เครื่องมือในการพัฒนาต้นแบบ

การพัฒนาต้นแบบของส่วนขยายยูดีดีไอในวิทยานิพนธ์นี้ได้พัฒนาโดยเครื่องมือต่อไปนี้

- ส่วนขยายยูดีดีไอ พัฒนาขึ้นภายใต้ข้อกำหนดของยูดีดีไอ รุ่นที่ 1.0 ตาม UDDI Open Draft Specification 30 September 2000 โดยใช้บริการ เอนเทอร์ไพรส์ ยูดีดีไอ เซอร์วิส (Enterprise UDDI Services) ของระบบปฏิบัติการไมโครซอฟท์ วินโดวส์ เซิร์ฟเวอร์ รุ่น 2003 ฉบับเอนเทอร์ไพรส์ (Microsoft Windows Server 2003 Enterprise Edition)
- ยูดีดีไอ รีจิสตรี ใช้ระบบฐานข้อมูลไมโครซอฟท์ ซีเคเวล เซิร์ฟเวอร์ รุ่น 2000 (Microsoft SQL Server 2000)
- เครื่องมือในการพัฒนาส่วนต่อประสาน ใช้ไมโครซอฟท์ วิซวล สตูดิโอ ดอทเน็ต รุ่น 2003 (Microsoft Visual Studio .NET 2003) ภาษาที่ใช้ในการเขียนโปรแกรมคือ ซีชาร์ป (C#)

การทดสอบต้นแบบส่วนขยายยูดีดีไอ

ส่วนขยายยูดีดีไอ เป็นการขยายมาตรฐานของข้อกำหนดยูดีดีไอ ให้สามารถอธิบายคุณสมบัติของบริการ เพื่อกำหนดลักษณะเฉพาะของบริการ ซึ่งจะทำให้สามารถทำการค้นหาบริการได้เฉพาะเจาะจงมากขึ้น โดยการใช้ประโยชน์จากคุณสมบัติของบริการเป็นเงื่อนไขในการค้นหา ซึ่งต้นแบบส่วนขยายยูดีดีไอนั้น จะมีส่วนต่อประสานโปรแกรมที่ติดต่อกับผู้ใช้งานและส่วนที่ทำงานเบื้องหลังเพื่อให้สามารถทำการโฆษณาและการค้นหาด้วยคุณสมบัติบริการตามที่ต้องการแบบไว้ ดังรูปที่ 5.1



รูปที่ 5.1 ส่วนต่อประสานโปรแกรมของส่วนขยายยูดีดีไอ

ลำดับขั้นตอนการทำงานเป็นดังนี้

1. การโฆษณาสอบการ (Publish) ผู้ให้บริการ (Service Provider) สามารถทำการโฆษณาสอบการของเว็บเซอร์วิสพร้อมคุณสมบัติของบริการ โดยเรียกใช้งานส่วนต่อประสานการโฆษณา (publishingAPI) มายังยูดีดีไอ รีจิสตรี (UDDI Registry)
2. ส่วนต่อประสานโปรแกรมส่วนขยาย (Extended API) ที่ทำหน้าที่เบื้องหลังเกี่ยวกับการโฆษณา ซึ่งประกอบไปด้วย propertyAPI propertySetAPI และ servicePropertyAPI จะทำการนำข้อมูลที่ผู้ให้บริการโฆษณาไปเก็บในคลังข้อมูลโฆษณาสอบการในส่วนขยายคุณสมบัติบริการ (Service Properties)

3. การค้นหาบริการ (Find) โดยใช้คุณสมบัติบริการเป็นเงื่อนไขในการค้นหา ผู้รับบริการ (Service Requester) ทำการค้นหาบริการโดยเรียกใช้ส่วนต่อประสานการค้นหา (inquiryAPI) ซึ่งสามารถทำการระบุคุณสมบัติบริการที่ต้องการเป็นเงื่อนไขในการค้นหา
4. ส่วนต่อประสานโปรแกรมส่วนขยาย (Extended API) ที่ทำหน้าที่เบื้องหลังเกี่ยวกับการค้นหา ซึ่งประกอบไปด้วย propertyAPI propertySetAPI และ servicePropertyAPI จะทำการค้นหาข้อมูลบริการที่ผู้ให้บริการโฆษณาไว้พร้อมกับคุณสมบัติบริการจากคลังข้อมูลโฆษณาบริการในส่วนขยายคุณสมบัติบริการ (Service Properties)
5. ผู้รับบริการจะได้ข้อมูลของบริการ ซึ่งตรงกับเงื่อนไขคุณสมบัติบริการที่ระบุไว้คืนกลับมาในรูปแบบของเอกสารเอ็กซ์เอ็มแอล (Service Information)
6. ผู้รับบริการสามารถสร้างโปรแกรมเพื่อทำการผูกมัด (Bind) กับบริการของผู้ให้บริการตามข้อตกลงที่ระบุไว้ในเอกสารข้อมูลที่ได้ในข้อ 5 เพื่อทำการขอรับบริการต่อไป

### 5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

1. เครื่องเซิร์ฟเวอร์ AMD Athlon™ 1.33 กิกะเฮิร์ตซ์ หน่วยความจำขนาด 512 เมกะไบต์ ระบบปฏิบัติการวินโดวส์เซิร์ฟเวอร์ รุ่น 2003 ฉบับเอ็นเทอร์ไพรส์ (Microsoft Windows Server 2003 Enterprise Edition) โดยเครื่องเซิร์ฟเวอร์นี้มีชื่อคอมพิวเตอร์คือ dotnet และจำลองเป็น ยูดีดีไอ รีจิสตรี โดยเรียกใช้บริการต่างๆ ดังนี้
  - บริการเอ็นเทอร์ไพรส์ยูดีดีไอเซอร์วิส (Enterprise UDDI Services) รุ่น 5.2.3663.0
  - บริการเว็บเซิร์ฟเวอร์ (IIS : Internet Information Services) รุ่น 6.0
  - ไมโครซอฟท์ดอตเน็ตเฟรมเวิร์ค (Microsoft.Net Framework) รุ่น 1.0.3705.21
  - ระบบจัดการฐานข้อมูลไมโครซอฟท์ซีเคเวลเซิร์ฟเวอร์ รุ่น 2000 (Microsoft SQL Server 2000)
  - อินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer) รุ่น 6.0 ทำหน้าที่เป็นเว็บเบราว์เซอร์
  - ส่วนต่อประสานโปรแกรมของส่วนขยายยูดีดีไอ (Extended API InquiryAPI และ PublishingAPI)



2. เครื่องไคลเอนต์ Intel® Pentium® M 1.3 กิกะเฮิร์ตซ์ หน่วยความจำขนาด 256 เมกะไบต์ ระบบปฏิบัติการวินโดวส์เอ็กซ์พี รุ่น 2002 ฉบับโฮม (Microsoft Windows XP Home Edition Version 2002) เพื่อจำลองเป็นผู้ให้บริการที่ทำการโฆษณาบริการและจำลองเป็นผู้รับบริการที่ทำการค้นหาบริการ โดยเรียกใช้บริการ ดังนี้

- อินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer) รุ่น 6.0 ทำหน้าที่เป็นเว็บเบราว์เซอร์
- โปรแกรมต้นแบบที่ทำหน้าที่โฆษณาบริการและค้นหาบริการ

## 5.2 การทดสอบการโฆษณาบริการพร้อมคุณสมบัติบริการ

การทดสอบส่วนขยายยูดีดีไอ ในส่วนที่ทำหน้าที่เกี่ยวกับการโฆษณาบริการพร้อมคุณสมบัติ ซึ่งเป็นการเพิ่มเติมข้อกำหนดของยูดีดีไอ โดยที่แต่เดิมนั้นยังไม่มีข้อกำหนดในการเพิ่มข้อมูลเกี่ยวกับคุณลักษณะเฉพาะของบริการ ตัวอย่างการทดสอบในที่นี้จะเกี่ยวกับธุรกิจโรงแรมซึ่งมีบริการการจองห้องพัก โดยการทดสอบจะทำกับข้อมูลของธุรกิจโรงแรมจำนวน 4 แห่งและธุรกิจตัวแทนท่องเที่ยว 1 แห่ง โดยทั้ง 5 แห่งนั้นมีบริการที่เหมือนกันคือ Hotel Reservation ซึ่งเป็นการจองห้องพักของโรงแรม โดยธุรกิจทั้ง 5 แห่ง คือ

1. US Travel Agency เป็นธุรกิจตัวแทนท่องเที่ยว
2. Holiday Inn เป็นธุรกิจการโรงแรม
3. Marriott เป็นธุรกิจการโรงแรม
4. Hyatt เป็นธุรกิจการโรงแรม
5. Sheraton เป็นธุรกิจการโรงแรม

การทดสอบเริ่มจากการป้อนข้อมูลที่บริการยูดีดีไอที่ <http://dotnet/uddi/> ในส่วนหน้าจอของการโฆษณาบริการ ข้อมูลเหล่านี้เป็นข้อมูลของธุรกิจและบริการที่เรียกใช้บริการยูดีดีไอเดิม เพื่อเป็นข้อมูลตั้งต้นในการทดสอบ ตัวอย่างหน้าจอการเพิ่มข้อมูลของโรงแรม Hyatt เป็นดังรูปที่ 5.2 และข้อมูลในรูปแบบเชิงโครงสร้างเอ็กซ์เอ็มแอลแสดงในรูปที่ 5.3

My UDDI | Hyatt  
**Hotel reservation**

Name, describe, and categorize each Web service, making it easy to locate. Publish one or more access points for a service by adding a binding, and then add instance information for each interface implemented at that access point.

Details Bindings Categories

Name and briefly describe this service in one or more languages. The service key is unique and is intended for use in programmatic queries only.

**Service Key:**  
e5544aba-dd13-4af1-b29f-9ff915023fb5

Name	Actions
(English) Hotel reservation	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
1 record(s) found.	<input type="button" value="Add Name"/>

Descriptions	Actions
(English) online booking, discount 5%	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
1 record(s) found.	<input type="button" value="Add Description"/>

รูปที่ 5.2 การเพิ่มข้อมูลบริการตามข้อกำหนดยูดีดีไอ

```
<?xml version="1.0" encoding="UTF-8" ?>
- <businessEntity>
  <name>Hyatt</name>
  <description>Hotel</description>
  + <contacts>
  + <categoryBag>
  - <businessServices>
    - <businessService>
      <name>Hotel Reservation</name>
      <description>online booking, discount 5%</description>
    + <bindingTemplates>
    + <categoryBag>
    </businessService>
  </businessServices>
</businessEntity>
```

รูปที่ 5.3 ข้อมูลบริการเชิงโครงสร้างตามข้อกำหนดยูดีดีไอ

จากตัวอย่างข้างต้น การโฆษณาบริการสามารถโฆษณาได้เพียงข้อมูลบริการที่ไม่ละเอียดนัก แม้จะมีการจัดหมวดหมู่ด้วยลักษณะการจำแนกประเภทบริการ แต่รายละเอียดคุณสมบัติหลายๆ อย่างของบริการ หากต้องการอธิบายอาจจะจัดเก็บในหมวดคำอธิบาย (Description) ได้ (ในที่นี้คือ online booking และ discount 5%) จึงทำให้ไม่เป็นระเบียบแบบแผนนัก อีกทั้งทำให้ลำบากในการค้นหาตามคุณสมบัติเหล่านั้น และเมื่อธุรกิจประเภทเดียวกันทำการระบุหมวดคำอธิบายแตกต่างกันไป จะทำให้เกิดความยุ่งยากในการเปรียบเทียบบริการจำนวนมากเพื่อตัดสินใจเลือกใช้บริการ

เพื่อลดปัญหาดังกล่าว ต้นแบบได้ทำการขยายข้อกำหนดของยูดีดีไอเพื่อให้สามารถโฆษณาบริการพร้อมคุณสมบัติเฉพาะต่างๆ หรือเซตของคุณสมบัติได้ ตัวอย่างการเรียกใช้งานส่วนขยายยูดีดีไอที่เกี่ยวข้องกับการโฆษณา (publishingAPI) เพื่อเพิ่มคุณสมบัติบริการให้กับบริการ Hotel Reservation ของโรงแรม Hyatt ดังนี้

คุณสมบัติของบริการที่เพิ่มเติม คือ

- discount (%) หมายถึง ส่วนลดค่าบริการ
- credit card หมายถึง บัตรเครดิตที่สามารถชำระค่าบริการได้
- online booking หมายถึง สามารถทำรายการจองห้องพักได้เอง

เซตของคุณสมบัติของบริการที่เพิ่มเติม คือ

- room price หมายถึง เซตของคุณสมบัติที่แสดงราคาห้องพัก ในหน่วยดอลลาร์สหรัฐ ประกอบด้วยคุณสมบัติ ดังนี้
  - single-bed room หมายถึง ราคาห้องพักเตียงเดี่ยว
  - double-bed room หมายถึง ราคาห้องพักเตียงคู่
  - extra-bed หมายถึง ราคาเตียงเสริม

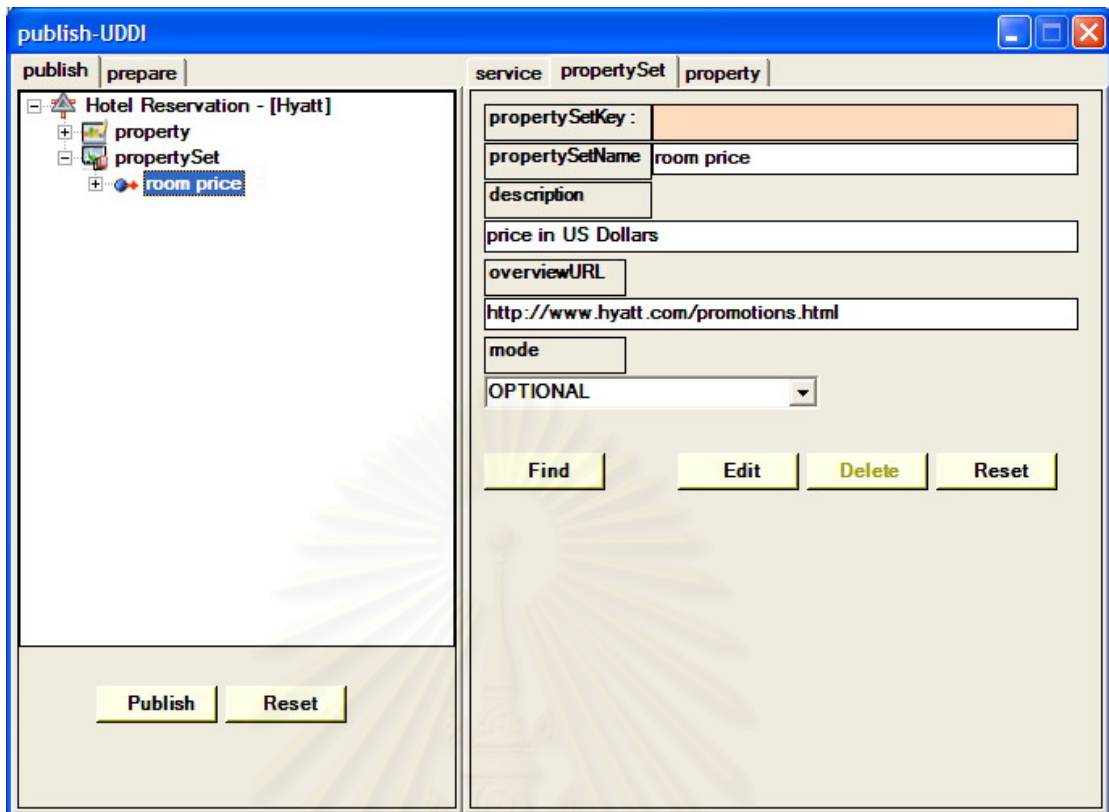
ตัวอย่างหน้าจอของการเพิ่มคุณสมบัติชื่อ discount (%) ที่มีชนิดข้อมูลเป็น integer มีค่าเท่ากับ 5 แสดงในรูปที่ 5.4 และตัวอย่างหน้าจอการเพิ่มเซตของคุณสมบัติชื่อ roomPrice และหน้าจอการเพิ่มคุณสมบัติเกี่ยวกับราคาห้องพักแบบต่างๆ ลงในเซตของคุณสมบัติแสดงในรูปที่ 5.5 และ รูปที่ 5.6 ตามลำดับ

The screenshot shows the 'publish-UDDI' application window. The left pane displays a tree view of a service named 'Hotel Reservation - [Hyatt]'. Under the 'property' folder, the 'discount (%)' property is selected. The right pane shows the configuration details for this property:

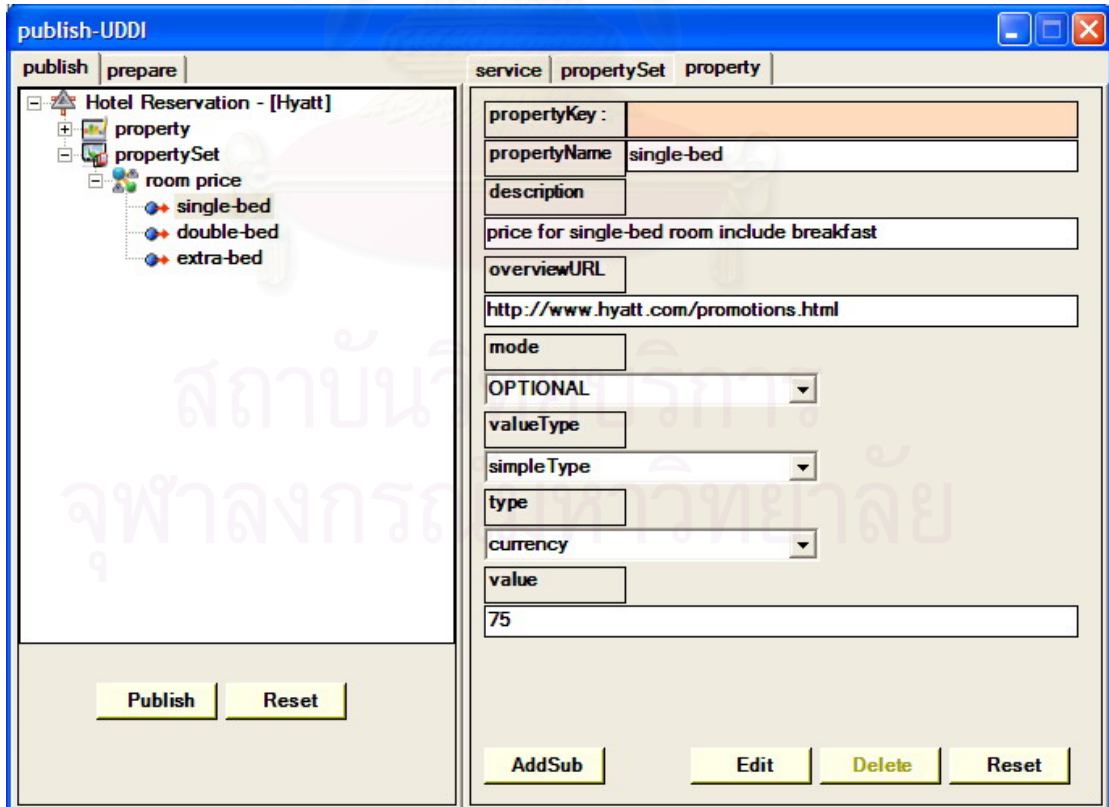
- propertyKey :** (empty)
- propertyName:** discount (%)
- description:** discount in percentage of price for online booking
- overviewURL:** http://www.hyatt.com/promotions.html
- mode:** OPTIONAL
- valueType:** (empty)
- simpleType:** integer
- type:** integer
- value:** 5

Buttons at the bottom include 'Publish', 'Reset', 'AddSub', 'Edit', 'Delete', and 'Reset'.

รูปที่ 5.4 การเพิ่มข้อมูลคุณสมบัติบริการตามส่วนขยายยูดีดีไอ



รูปที่ 5.5 การเพิ่มเซตของคุณสมบัติ room price



รูปที่ 5.6 การเพิ่มคุณสมบัติเป็นสมาชิกเซต room price

รายละเอียดของบริการข้างต้นพร้อมด้วยคุณสมบัติต่างๆ ในรูปแบบเชิงโครงสร้าง เพื่อให้  
โฆษณาแก่ส่วนขยายยูทิลิตี้เป็นดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<save_serviceProperty>
<businessService serviceKey="e5544aba-dd13-4af1-b29f-9ff915023fb5"
businessKey="b14de731-7f2e-4a64-ae4a-af0e1eeabcf">
  <businessName>Hyatt</businessName>
  <name>Hotel Reservation</name>
  <description>room reservation service</description>
  <servicePropertyBag>
    <propertyBag>
      <propertyReference mode="OPTIONAL">
        <name>discount (%)</name>
        <description>discount in percentage of price for on line-
booking</description>
        <valueReference valueTypeRef="simpleType">
          <simpleType>
            <simpleValue type="integer">5</simpleValue>
          </simpleType>
        </valueReference>
      </propertyReference>
      <propertyReference mode="OPTIONAL">
        <name>credit card</name>
        <description>accepted credit cards</description>
        <valueReference valueTypeRef="arrayType">
          <arrayType type="string" no_of_element="3">
            <arrayValue index_no="0">Visa</arrayValue>
            <arrayValue index_no="1">Master</arrayValue>
            <arrayValue index_no="2">American Express</arrayValue>
          </arrayType>
        </valueReference>
      </propertyReference>
      <propertyReference mode="OPTIONAL">
        <name>online booking</name>
        <description>available for online-booking</description>
```

```

    <valueReference valueTypeRef="simpleType">
      <simpleType>
        <simpleValue type="boolean">>true</simpleValue>
      </simpleType>
    </valueReference>
  </propertyReference>
</propertyBag>
<propertySet mode="OPTIONAL">
  <name>room price</name>
  <description>price in US Dollars</description>
  <propertyBag>
    <propertyReference mode="OPTIONAL">
      <name>single-bed room</name>
      <description>price for single-bed room with
breakfast</description>
      <valueReference valueTypeRef="simpleType">
        <simpleType>
          <simpleValue type="currency">75</simpleValue>
        </simpleType>
      </valueReference>
    </propertyReference>
    <propertyReference mode="OPTIONAL">
      <name>double-bed room</name>
      <description>price for double-bed room with
breakfast</description>
      <valueReference valueTypeRef="simpleType">
        <simpleType>
          <simpleValue type="currency">99</simpleValue>
        </simpleType>
      </valueReference>
    </propertyReference>
    <propertyReference mode="OPTIONAL">
      <name>extra-bed</name>
      <description>price for extra-bed with breakfast</description>
      <valueReference valueTypeRef="simpleType">

```

```

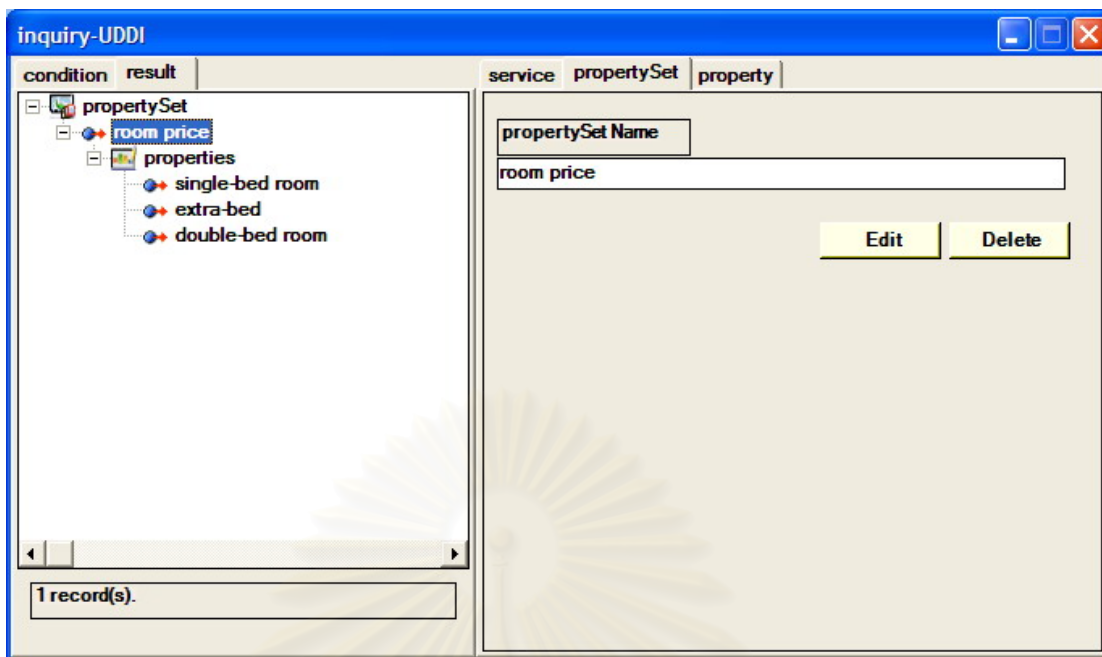
<simpleType>
  <simpleValue type="currency">40</simpleValue>
</simpleType>
</valueReference>
</propertyReference>
</propertyBag>
</propertySet>
</servicePropertyBag>
</businessService>
</save_serviceProperty>

```

การบันทึกข้อมูลที่มีเงื่อนไขคุณสมบัติบริการทำโดยการเรียกใช้งานตัวกระทำการ saveServiceProperty ของส่วนต่อประสาน publishingAPI เพื่อทำการบันทึกข้อมูลคุณสมบัติบริการไปที่ยูทิลิตี้โอริจิสตรี โดยข้อมูลในส่วนนี้จะเป็นการเพิ่มรายละเอียดคุณสมบัติและเซตของ คุณสมบัติให้กับบริการ ทำให้บริการ Hotel Reservation ของ Hyatt มีรายละเอียดเฉพาะมากขึ้น ผลลัพธ์ที่ได้ของตัวกระทำการผ่านพารามิเตอร์ที่คืนค่าในที่นี้ได้ค่าชุดตัวอักษรเป็น Success หมายถึง การบันทึกข้อมูลทำได้สำเร็จเรียบร้อยแล้ว ซึ่งหากมีข้อผิดพลาดเกิดขึ้นจะทำการคืนค่า กลับออกมาเป็นข้อความแสดงการทำงานผิดพลาดของระบบ

ตัวกระทำการ saveServiceProperty นี้ ต้องการพารามิเตอร์ 2 ตัวคือ พารามิเตอร์ข้อมูล เข้าที่มีชนิดโครงสร้างข้อมูลเป็น save\_serviceProperty ซึ่งระบุค่าข้อมูลคุณสมบัติของบริการ และพารามิเตอร์ po\_returnMsg ซึ่งเป็นชนิดชุดตัวอักษรที่เป็นข้อมูลออกที่รับข้อความแสดงผล การทำงานดังข้างต้น ส่วนการคืนค่ากลับของตัวกระทำการนี้ จะคืนค่ากลับมาในรูปของโครงสร้าง save\_serviceProperty เช่นเดียวกับพารามิเตอร์ข้อมูลนำเข้าแต่มีการเพิ่มหมายเลขประจำตัว คุณสมบัติเข้าไปในข้อมูลด้วย

เซตของคุณสมบัติที่มีการโฆษณาไปแล้วนั้น สามารถนำมาใช้ซ้ำได้โดยบริการที่ต้องการ โฆษณาคุณสมบัติที่คล้ายกัน เพื่อให้บริการที่เหมือนกันมีคุณสมบัติที่คล้ายคลึงกัน เช่น Holiday Inn มีบริการห้องพักโรงแรมเช่นเดียวกัน หากต้องการโฆษณาด้วยเซตของคุณสมบัติแสดง ราคาห้องพัก room price เหมือนกัน ก็สามารถทำการค้นหาและใช้ซ้ำเซตของคุณสมบัติที่มีการ กำหนดไว้แล้วได้ ดังแสดงรูปการค้นหาเซตของคุณสมบัติในรูปที่ 5.7

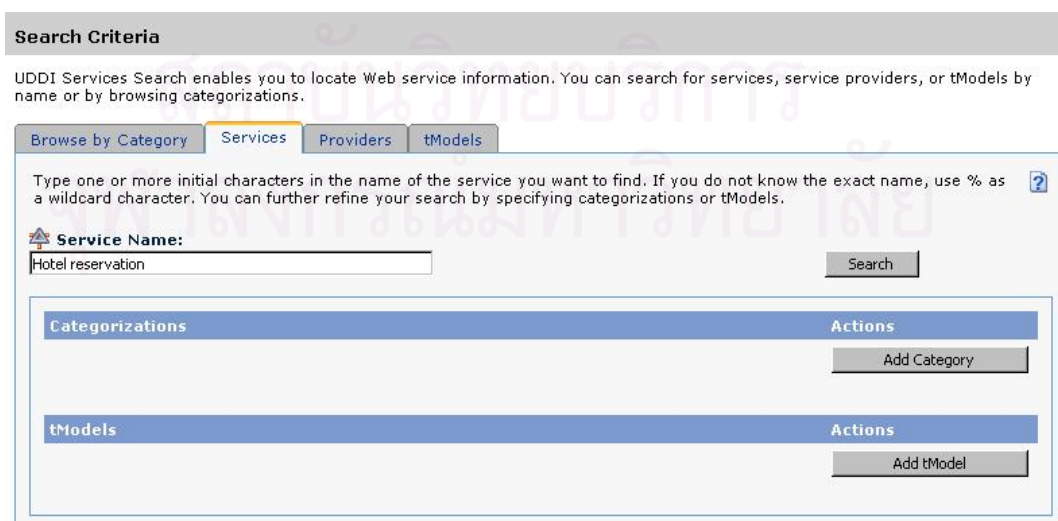


รูปที่ 5.7 การค้นหาเซตของคุณสมบัติ room price เพื่อใช้ซ้ำ

### 5.3 การทดสอบการค้นหาบริการด้วยคุณสมบัติบริการ

ตัวอย่างการค้นหาบริการในที่นี้คือ ผู้รับบริการต้องการค้นหาโรงแรมที่มีบริการจองห้องพัก และมีส่วนลดในการบริการ 5 เปอร์เซ็นต์

การทดสอบจะเป็นการค้นหาบริการโดยใช้คุณสมบัติบริการเป็นเงื่อนไขในการค้นหาข้อมูล เปรียบเทียบกับการค้นหาข้อมูลด้วยบริการของยูดีดีไอเดิม จากข้อมูลตัวอย่างที่กล่าวไว้แล้วใน หัวข้อที่ 5.2 ซึ่งการค้นหาข้อมูลด้วยบริการของยูดีดีไอเดิมจะเป็นการค้นหาข้อมูลเกี่ยวกับบริการ โดยใช้ชื่อของบริการ Hotel Reservation เป็นเงื่อนไขในการค้นหา ดังรูปที่ 5.8



รูปที่ 5.8 การค้นหาข้อมูลบริการตามข้อกำหนดยูดีดีไอเดิม



ผลลัพธ์ที่ได้คือธุรกิจทั้งหลายที่มีบริการที่ชื่อว่า Hotel Reservation ซึ่งในการทดสอบนี้จะได้ผลลัพธ์เป็น 5 รายการ ตามข้อมูลตัวอย่างที่กล่าวถึงในหัวข้อที่ข้อ 5.2 แสดงได้ดังรูปที่ 5.9



รูปที่ 5.9 รายการบริการที่มีชื่อว่า Hotel Reservation

จากผลลัพธ์ที่ได้จากการค้นหาด้วยบริการของยูดีดีไอเดิมนั้น เป็นผลจากการค้นหาด้วยเงื่อนไขที่ยังไม่ละเอียดนัก ทำให้ผู้รับบริการต้องทำการสืบค้นข้อมูลต่อไปอีกด้วยวิธีการอื่น เพื่อหาว่าโรงแรมใดบ้างที่มีส่วนลดค่าบริการ 5 เปอร์เซ็นต์ เช่น ต้องดูผลการค้นหาที่ได้มาเองว่าบริการใดบ้างที่ระบุการให้ส่วนลดไว้ในข้อมูลหมวดคำอธิบาย หรือทำการสืบค้นไปยังเว็บเพจของโรงแรม หรือทำการสอบถามข้อมูลเพิ่มเติมทางโทรศัพท์ เป็นต้น

ในการทดสอบการค้นหาข้อมูลบริการในส่วนของยูดีดีไอซึ่งสามารถทำการค้นหาบริการโดยใช้คุณสมบัติของบริการเป็นเงื่อนไขในการค้นหานั้น ส่วนต่อประสานโปรแกรมที่ทำหน้าที่เกี่ยวกับการค้นหา (inquiryAPI) จะถูกเรียกผ่านตัวกระทำการ findServiceProperty เพื่อทำการค้นหาบริการด้วยการระบุคุณสมบัติเป็นเงื่อนไข โดยมีพารามิเตอร์ 2 ตัวคือ ข้อมูลที่ใช้ในการค้นหาบริการ Hotel Reservation พร้อมระบุคุณสมบัติ discount 5% ซึ่งอยู่ในรูปของโครงสร้างชนิด find\_serviceProperty (รูปที่ 5.10) และพารามิเตอร์อีกหนึ่งตัวคือ po\_returnMsg ซึ่งเป็นชนิดชุดตัวอักษรที่จะรับข้อความการทำงานจากระบบ ถ้าหากสามารถทำงานได้สำเร็จจะได้ค่าเป็น Success ถ้าเกิดการผิดพลาดในการทำงานจะได้ค่าข้อความแสดงการทำงานผิดพลาดของระบบ ส่วนผลลัพธ์ที่ได้จากการเรียกใช้งานตัวกระทำการ findServiceProperty จะอยู่ในรูปของโครงสร้าง servicePropertyList ซึ่งเป็นรายการบริการพร้อมคุณสมบัติ

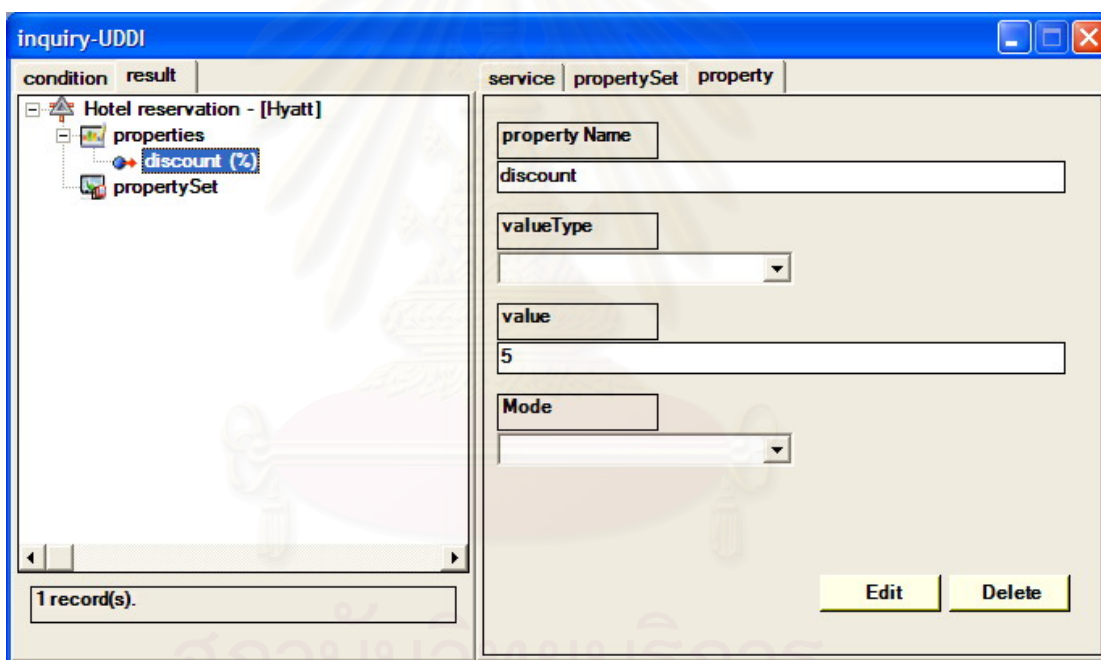
```

<?xml version="1.0" encoding="UTF-8" ?>
- <find_serviceProperty>
  <name>Hotel Reservation</name>
  - <propertyValueBag>
    - <propertyValueInfo>
      <name>discount (%)</name>
      <propertyValue>5</propertyValue>
    </propertyValueInfo>
  </propertyValueBag>
</find_serviceProperty>

```

รูปที่ 5.10 โครงสร้างข้อมูลของการค้นหาตามส่วนขยายยูดีดีไอ (find\_serviceProperty)

ค่าที่คืนกลับมาในรูปแบบโครงสร้าง servicePropertyList นั้นจะแสดงเพียงบริการที่ตรงตามเงื่อนไข คือเป็นบริการ Hotel Reservation ที่มีคุณสมบัติ discount (%) มีค่าเท่ากับ 5 เท่านั้น ดังนั้นค่าที่คืนกลับมาจึงมีเพียง 1 รายการของโรงแรม Hyatt (รูปที่ 5.11)



รูปที่ 5.11 ข้อมูลที่ได้จากการค้นหาข้อมูลด้วยส่วนขยายยูดีดีไอโดยระบุเงื่อนไข

รายละเอียดของผลลัพธ์ servicePropertyList ที่คืนกลับมาจากการค้นหาดังข้างต้น แสดงในรูปแบบเชิงโครงสร้างได้ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<servicePropertyList xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```

<servicePropertyInfos>
  <servicePropertyInfo serviceKey="e5544aba-dd13-4af1-b29f-
9ff915023fb5" businessKey="b14de731-7f2e-4a64-ae4a-af0e1eeabcfe">
    <businessName>Hyatt</businessName>
    <name>Hotel reservation</name>
    <propertyChildren>
      <propertyChild propertyKey="d804981a-24c4-466c-
873a-ac87c854b32b" mode="OPTIONAL">
        <name>discount (%)</name>
        <description>discount in percentage of price for
online-booking</description>
      </propertyChild>
    </propertyChildren>
  </servicePropertyInfo>
</servicePropertyInfos>
</servicePropertyList>

```

สังเกตได้ว่าผลการค้นหาข้างต้นจะแสดงข้อมูลที่มีลักษณะเป็นโครงสร้างซึ่งระบุว่าบริการประกอบด้วยคุณสมบัติอะไรบ้าง มีคำอธิบายอย่างไร ตามเงื่อนไขที่ระบุในการค้นหา แต่หากต้องการทราบรายละเอียดของคุณสมบัติต่างๆ เช่น ค่าของคุณสมบัติของบริการ จะต้องทำการเรียกใช้งานตัวกระทำ `getServicePropertyDetail` อีกทีหนึ่ง โดยระบุรหัสประจำตัวบริการที่ต้องการดูรายละเอียด เป็นเงื่อนไข

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 6

### สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลสรุปของงานวิจัย ปัญหาและข้อจำกัดต่างๆ รวมทั้งข้อเสนอแนะที่สามารถนำไปปรับปรุงและพัฒนาได้ต่อไปในอนาคต โดยมีรายละเอียดดังต่อไปนี้

#### 6.1 สรุปผลการวิจัย

งานวิจัยชิ้นนี้ ได้ทำการออกแบบและพัฒนาส่วนขยายยูดีดีไอเพื่อสนับสนุนการโฆษณา และการค้นหาบริการโดยใช้คุณสมบัติของบริการ โดยมีพื้นฐานแนวคิดมาจากบริการเทรดเดอร์ ที่มุ่งเน้นในการเพิ่มโครงสร้างข้อมูลที่จะสามารถทำการเพิ่มคุณสมบัติให้กับบริการ เพื่อให้สามารถกำหนดข้อมูลให้กับบริการได้เฉพาะเจาะจงมากขึ้น และสามารถทำการค้นหาข้อมูลบริการโดยใช้คุณสมบัติที่เพิ่มขึ้นเหล่านั้นเป็นเงื่อนไขในการค้นหาได้ ผู้ให้บริการสามารถกำหนดคุณสมบัติหลายๆ ตัวให้กับบริการ หรือกำหนดในรูปเซตของคุณสมบัติที่เกี่ยวข้องกัน ซึ่งการอ้างอิงถึงเซตของบริการ จะเป็นการกำหนดคุณสมบัติทั้งหมดที่ระบุอยู่ในเซตนั้น อันเป็นการสนับสนุนให้เกิดลักษณะของการนำนิยามของคุณสมบัติตามเซตของคุณสมบัติกลับมาใช้ใหม่ร่วมกันระหว่างบริการต่างๆ ต้นแบบส่วนขยายยูดีดีไอนี้เป็นตัวอย่างการเสริมความสามารถในการทำงานของข้อกำหนดยูดีดีไอเดิมโดยไม่ได้ปรับแก้ จึงทำให้สามารถรองรับการโฆษณาและค้นหาบริการตามมาตรฐานเดิมอยู่

จากผลการทดสอบการใช้งานต้นแบบ แสดงให้เห็นถึงส่วนขยายยูดีดีไอที่สามารถระบุคุณสมบัติของบริการและใช้เป็นเงื่อนไขในการค้นหาเพื่อคัดกรองข้อมูลให้มีความละเอียดขึ้นได้ ซึ่งทำให้ปริมาณข้อมูลบริการที่ได้จากการค้นหาลดลงและตรงความต้องการตามเงื่อนไขที่ระบุ

งานวิจัยนี้มุ่งที่การพัฒนาโครงสร้างเพื่อรองรับการประกาศคุณสมบัติเพิ่มให้กับบริการเท่านั้น โดยการกำหนดว่าบริการควรประกาศคุณสมบัติใดเพิ่มเติมเพื่อโฆษณาบ้างนั้นเป็นส่วนที่นอกเหนือขอบเขตงานวิจัย เนื่องจากเป็นเรื่องเฉพาะของโดเมนงานของบริการซึ่งผู้เชี่ยวชาญของโดเมนควรเป็นผู้กำหนดเอง อย่างไรก็ตาม ผู้วิจัยเห็นว่าการโฆษณาบริการด้วยคุณสมบัติต่างๆ นั้นควรโฆษณาคุณสมบัติที่บ่งบอกถึงลักษณะที่สำคัญของบริการ ซึ่งเป็นคุณสมบัติที่จะให้ข้อมูลสำคัญที่ดึงดูดใจผู้รับบริการหรือเป็นคุณสมบัติที่ทำให้บริการนั้นดูแตกต่างจากบริการอื่นๆ ทั้งนี้โดยไม่จำเป็นต้องทำการโฆษณารายละเอียดทั้งหมด ยกตัวอย่างเช่น บริการร้านขายหนังสือซึ่งต้องการโฆษณาบริการสั่งซื้อหนังสือออนไลน์ ควรจะทำการโฆษณาบริการพร้อมระบุคุณสมบัติชื่อ "AvailableCategories" ซึ่งระบุประเภทของหนังสือที่ร้านมีขาย มากกว่าที่จะทำการโฆษณาด้วย

คุณสมบัติชื่อ “AvailableBookTitles” ซึ่งระบุรายชื่อหนังสือทั้งหมดที่มีขาย ซึ่งจะส่งผลให้ส่วนขยายยูดีไอต้องเก็บข้อมูลรายชื่อหนังสือเป็นจำนวนมากเกินความจำเป็นและไม่เกิดประโยชน์สมค่านัก ในกรณีนี้ผู้วิจัยเห็นว่าผู้รับบริการสามารถทราบถึงรายชื่อหนังสือทั้งหมดที่มีขายที่ร้านได้ โดยการเข้าถึงเว็บไซต์ของร้านซึ่งมีที่อยู่บนอินเทอร์เน็ตระบุไว้ในคุณสมบัติตัวหนึ่งของบริการแทน น่าจะเหมาะสมกว่า กล่าวคือคุณสมบัติที่จะโฆษณาควรเป็นเมตาดาตาของบริการมากกว่าตัวข้อมูลของบริการเอง

## 6.2 ปัญหาและข้อจำกัดที่ได้พบจากการวิจัย

1. คุณสมบัติต่างๆ ที่นำมาโฆษณาให้กับบริการนั้น โดยนัยแล้วอาจเป็นคุณสมบัติเดียวกัน แต่ผู้ให้บริการอาจจะใช้ชื่อคนละชื่อ จึงอาจจะทำให้ผู้รับบริการค้นหาข้อมูลเหล่านี้ไม่พบก็ได้
2. คุณสมบัติและเซตของคุณสมบัติที่จะนำมาโฆษณา เพื่อเป็นคุณสมบัติและเซตของคุณสมบัติกลางนั้น ควรจะมีองค์กรที่มาจัดหมวดหมู่ของคุณสมบัติให้เป็นเซตคุณสมบัติที่เป็นมาตรฐาน เพื่อให้เข้าใจตรงกันและสามารถนำไปใช้ได้ในความหมายเดียวกัน เพราะการเปิดโอกาสให้ผู้ให้บริการสามารถที่จะทำการกำหนดข้อมูลของตนขึ้นมาได้เองนั้น ก็จะทำให้ข้อมูลไม่มีความเป็นหนึ่งเดียว
3. ในงานวิจัยนี้ไม่ได้ออกแบบในเรื่องของความปลอดภัยของข้อมูล เพราะต้องการแสดงให้เห็นถึงการออกแบบการโฆษณาและค้นหาคุณสมบัติเท่านั้น โดยยังมีข้อกำหนดอื่นๆ ที่เกี่ยวข้องกับเรื่องความปลอดภัยของข้อมูลอยู่แล้ว
4. งานวิจัยนี้ ใช้พื้นฐานจากข้อกำหนดของยูดีไอ รุ่นที่ 1.0 ซึ่งปัจจุบันข้อกำหนดยูดีไอได้ถูกพัฒนาไปเป็น รุ่นที่ 3.0 แล้ว แต่ข้อกำหนดหรือโครงสร้างหลักยังไม่ได้มีการเปลี่ยนแปลงมากนัก

## 6.3 ข้อเสนอแนะ

1. โครงสร้างของคุณสมบัติและเซตของคุณสมบัตินั้นสามารถนำไปพัฒนาเพิ่มเติมให้กับข้อมูลส่วนอื่นในยูดีไอ นอกเหนือไปจากบิสเนสเซอร์วิสได้ ตัวอย่างเช่น หากพัฒนาเพิ่มให้กับบิสเนสเอนทิตี ก็จะทำให้สามารถโฆษณาคุณสมบัติอื่นๆ ของตัวผู้ให้บริการได้
2. ในการค้นหาตามข้อกำหนดของยูดีไอนั้น สามารถที่จะระบุเงื่อนไขในการค้นหาได้หลายรูปแบบ โดยการค้นหาเป็นการเปรียบเทียบค่าข้อมูลที่โฆษณาไว้กับข้อมูลที่ระบุเมื่อค้นหา โดยจะค้นหาพบเมื่อค่าข้อมูลตรงกัน อย่างไรก็ตามการค้นหาอาจมีประสิทธิผลมากขึ้นเมื่ออาศัยตัวดำเนินการอื่นๆ ร่วมด้วย นอกเหนือไปจากการเทียบค่าตรงกันเพียงอย่าง

เดียว ตัวอย่างเช่น ตัวดำเนินการทางคณิตศาสตร์ เช่น มากกว่า หรือ น้อยกว่า หรือตัวดำเนินการทางตรรกะ เช่น และ หรือ เป็นต้น จะทำให้การกำหนดเงื่อนไขในการค้นหาทำได้หลากหลายและยืดหยุ่นขึ้น ในงานวิจัยนี้ไม่ได้ดำเนินการในเรื่องนี้ แต่การกำหนดโครงสร้างการนิยามคุณสมบัติให้มีชนิดข้อมูลแบบต่างๆ ถือเป็นจุดเริ่มต้นที่จะช่วยสนับสนุนการพัฒนาให้สามารถทำการค้นหาด้วยตัวดำเนินการต่างๆ ได้ต่อไป



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## รายการอ้างอิง

- [1] Sleeper, B., and Robins, B. "Defining Web Services". Available from: [http://www.stencilgroup.com/ideas\\_scope\\_200106wsdefined.html](http://www.stencilgroup.com/ideas_scope_200106wsdefined.html), [2001, Jun].
- [2] Bray, T., Paoli, J., and McQueen, C. M. S. "Extensible Markup Language (XML) 1.0". Available from: <http://www.w3.org/TR/1998/REC-xml-19980210.pdf>, [1998, Feb 10].
- [3] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., and Winer, D. "Simple Object Access Protocol (SOAP) 1.1". Available from: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, [2000, May 8].
- [4] Ehnebuske, D., Rogers, D., and Riegen, C. V. "UDDI Version 2.0 Data Structure Reference". Available from: <http://www.uddi.org/pubs/DataStructure-V2.00-Open-20010608.pdf>, [2001, Jun 8].
- [5] Mckee, B., Ehnebuske, D., and Rogers, D. "UDDI Version 2.0 API Specification". Available from: <http://www.uddi.org/pubs/ProgrammerAPI-V2.00-Open-20010608.pdf>, [2001, Jun 8].
- [6] OMG. "Trading Object Service Specification version 1.0". Available from: <ftp://ftp.omg.org/pub/docs/formal/00-06-27.pdf>, [2000, May].
- [7] Vasudevan, V., and Bannon, T. "WebTrader: Discovery and Programmed Access to Web-Based Services". Available from: <http://www.icc3.com/ec/objects/webtrader.html>, [1999].
- [8] Rungsawang, S. "A Design and Development of an Extension to Jini Lookup Service to Support Non-Exact Match Search". Master's Thesis Proposal, Department of Computer Engineering, Chulalongkorn University, January, 2002.
- [9] Christensen, E., Cuebera, F., Meredith, G., and Weerawarana, S. "Web Services Description Language (WSDL) 1.1". Available from: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, [2001, Mar 15].
- [10] Curbera, F., Ehnebuske, D., and Rogers, D. "Using WSDL in a UDDI Registry 1.05". Available from: <http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf>, [2001, Jun 25].



ภาคผนวก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## ภาคผนวก ก

### ข้อกำหนดเชิงโครงสร้างของส่วนขยายยูติตี้ไอ

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:simpleType name="type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="char"/>
      <xs:enumeration value="string"/>
      <xs:enumeration value="byte"/>
      <xs:enumeration value="integer"/>
      <xs:enumeration value="double"/>
      <xs:enumeration value="float"/>
      <xs:enumeration value="currency"/>
      <xs:enumeration value="date"/>
      <xs:enumeration value="time"/>
      <xs:enumeration value="datetime"/>
      <xs:enumeration value="boolean"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="simpleKey">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="fieldKey">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="structKey">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="arrayKey">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="enumKey">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>

```

```
</xs:simpleType>
<xs:simpleType name="unionKey">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="propertyKey" final="restriction">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="propertySetKey">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="serviceKey">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="businessKey">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:simpleType name="valueType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="simpleType"/>
    <xs:enumeration value="structType"/>
    <xs:enumeration value="arrayType"/>
    <xs:enumeration value="enumType"/>
    <xs:enumeration value="unionType"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="mode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OPTIONAL"/>
    <xs:enumeration value="READONLY"/>
    <xs:enumeration value="MANDATORY"/>
    <xs:enumeration value="MANDATORY_READONLY"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="findQualifier">
  <xs:restriction base="xs:string">
```

```

<xs:enumeration value="exactNameMatch"/>
<xs:enumeration value="caseSentiveMatch"/>
<xs:enumeration value="sortByNameAsc"/>
<xs:enumeration value="sortByNameDesc"/>
<xs:enumeration value="sortByDateAsc"/>
<xs:enumeration value="sortByDateDesc"/>
</xs:restriction>
</xs:simpleType>
<xs:element name="authInfo" type="xs:string"/>
<xs:element name="bindingTemplates"/>
<xs:element name="categoryBag"/>
<xs:complexType name="simpleValue">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="type" type="type" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="simpleValue" type="simpleValue"/>
<xs:complexType name="fieldValue">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="type" type="type" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="fieldValue" type="fieldValue"/>
<xs:complexType name="arrayValue">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="index_no" type="xs:integer" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="arrayValue" type="arrayValue"/>

```

```

<xs:complexType name="enumValue">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="valueRef" type="xs:integer" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="enumValue" type="enumValue"/>
<xs:complexType name="unionValue">
  <xs:complexContent>
    <xs:extension base="valueReference">
      <xs:attribute name="unionValueKey" type="unionKey" use="optional"/>
      <xs:attribute name="enumValueRef" type="xs:integer" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="unionValue" type="unionValue"/>
<xs:complexType name="defaultUnion">
  <xs:complexContent>
    <xs:extension base="valueReference">
      <xs:attribute name="defaultUnionKey" type="unionKey" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="defaultUnion" type="defaultUnion"/>
<xs:complexType name="fieldName">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="fieldName" type="fieldName"/>
<xs:complexType name="fieldStruct">
  <xs:sequence>
    <xs:element ref="fieldName"/>
    <xs:element ref="fieldValue"/>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:sequence>
    <xs:attribute name="fieldKey" type="fieldKey" use="required"/>
</xs:complexType>
<xs:element name="fieldStruct" type="fieldStruct"/>
<xs:complexType name="simpleType">
  <xs:sequence>
    <xs:element ref="simpleValue"/>
  </xs:sequence>
  <xs:attribute name="simpleKey" type="simpleKey" use="required"/>
</xs:complexType>
<xs:element name="simpleType" type="simpleType"/>
<xs:complexType name="structType">
  <xs:sequence>
    <xs:element ref="fieldStruct" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="structKey" type="structKey" use="required"/>
</xs:complexType>
<xs:element name="structType" type="structType"/>
<xs:complexType name="arrayType">
  <xs:sequence>
    <xs:element ref="arrayValue" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="arrayKey" type="arrayKey" use="required"/>
  <xs:attribute name="type" type="type" use="required"/>
  <xs:attribute name="no_of_element" type="xs:string" use="optional"/>
</xs:complexType>
<xs:element name="arrayType" type="arrayType"/>
<xs:complexType name="enumType">
  <xs:sequence>
    <xs:element ref="enumValue" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="enumKey" type="enumKey" use="required"/>
</xs:complexType>
<xs:element name="enumType" type="enumType"/>
<xs:complexType name="unionType">

```

```

<xs:sequence>
  <xs:element ref="enumType"/>
  <xs:element ref="unionValue" minOccurs="unbounded"/>
  <xs:element ref="defaultUnion"/>
</xs:sequence>
<xs:attribute name="unionKey" type="unionKey" use="required"/>
</xs:complexType>
<xs:element name="unionType" type="unionType"/>
<xs:complexType name="valueReference">
  <xs:choice minOccurs="0">
    <xs:element ref="simpleType"/>
    <xs:element ref="structType"/>
    <xs:element ref="arrayType"/>
    <xs:element ref="enumType"/>
    <xs:element ref="unionType"/>
  </xs:choice>
  <xs:attribute name="valueTypeRef" type="valueType" use="required"/>
</xs:complexType>
<xs:element name="valueReference" type="valueReference"/>
<xs:element name="constraint" type="xs:string"/>
<xs:element name="overviewURL" type="xs:string"/>
<xs:element name="description" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="businessName" type="xs:string"/>
<xs:complexType name="propertyReference">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="overviewURL" minOccurs="0"/>
    <xs:element ref="constraint" minOccurs="0"/>
    <xs:element ref="valueReference"/>
  </xs:sequence>
  <xs:attribute name="propertyKey" type="propertyKey" use="optional"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>

```

```

<xs:element name="propertyReference" type="propertyReference"/>
<xs:complexType name="propertyDetail">
  <xs:sequence>
    <xs:element ref="propertyReference" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyDetail" type="propertyDetail">
  <xs:annotation>
    <xs:documentation>Data Structure : Result of get_propertyDetail</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="propertyInfo">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="propertySetParents" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="propertyKey" type="propertyKey" use="required"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertyInfo" type="propertyInfo"/>
<xs:complexType name="propertyInfos">
  <xs:sequence>
    <xs:element ref="propertyInfo" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyInfos" type="propertyInfos"/>
<xs:complexType name="propertyList">
  <xs:sequence>
    <xs:element ref="propertyInfos"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyList" type="propertyList">
  <xs:annotation>
    <xs:documentation>Data Structure : Result of find_property</xs:documentation>
  </xs:annotation>

```

```

    </xs:annotation>
</xs:element>
<xs:complexType name="propertySetParent">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="optional"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertySetParent" type="propertySetParent"/>
<xs:complexType name="propertySetParents">
  <xs:sequence>
    <xs:element ref="propertySetParent" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetParents" type="propertySetParents"/>
<xs:complexType name="propertySetInfo">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="propertyChildren" minOccurs="0"/>
    <xs:element ref="propertySetBag" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="required"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertySetInfo" type="propertySetInfo"/>
<xs:complexType name="propertySetInfos">
  <xs:sequence>
    <xs:element ref="propertySetInfo" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetInfos" type="propertySetInfos"/>
<xs:complexType name="propertyBag">

```



```

    <xs:sequence>
      <xs:element ref="propertyReference" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
<xs:element name="propertyBag" type="propertyBag"/>
<xs:complexType name="propertySetBag">
  <xs:sequence>
    <xs:element ref="propertySetFamily" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetBag" type="propertySetBag"/>
<xs:complexType name="propertySet">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="overviewURL" minOccurs="0"/>
    <xs:element ref="constraint" minOccurs="0"/>
    <xs:element ref="propertyBag" minOccurs="0"/>
    <xs:element ref="propertySetBag" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="optional"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertySet" type="propertySet"/>
<xs:complexType name="propertySetChild">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="optional"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertySetChild" type="propertySetChild"/>
<xs:complexType name="propertySetChildren">
  <xs:sequence>

```

```

        <xs:element ref="propertySetChild" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="propertySetChildren" type="propertySetChildren"/>
<xs:complexType name="propertySetFamily">
    <xs:sequence>
        <xs:element ref="propertySetParent" minOccurs="0"/>
        <xs:element ref="propertySetChild" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="propertySetFamily" type="propertySetFamily"/>
<xs:complexType name="propertySetDetail">
    <xs:sequence>
        <xs:element ref="propertySet" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="propertySetDetail" type="propertySetDetail">
    <xs:annotation>
        <xs:documentation>Data Structure : Result of get_propertySetDetail</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="propertyChild">
    <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="description" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="propertyKey" type="propertyKey" use="required"/>
    <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="propertyChild" type="propertyChild"/>
<xs:complexType name="propertyChildren">
    <xs:sequence>
        <xs:element ref="propertyChild" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

```

```

<xs:element name="propertyChildren" type="propertyChildren"/>
<xs:complexType name="propertySetList">
  <xs:sequence>
    <xs:element ref="propertySetInfos"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetList" type="propertySetList">
  <xs:annotation>
    <xs:documentation>Data Structure : Result of find_propertySet</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="servicePropertyBag">
  <xs:sequence>
    <xs:element ref="propertyBag" minOccurs="0"/>
    <xs:element ref="propertySet" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="servicePropertyBag" type="servicePropertyBag"/>
<xs:complexType name="businessService">
  <xs:sequence>
    <xs:element ref="businessName" minOccurs="0"/>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element ref="description" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="bindingTemplates" minOccurs="0"/>
    <xs:element ref="categoryBag" minOccurs="0"/>
    <xs:element ref="servicePropertyBag" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="serviceKey" type="serviceKey" use="optional"/>
  <xs:attribute name="businessKey" type="businessKey" use="optional"/>
</xs:complexType>
<xs:element name="businessService" type="businessService"/>
<xs:complexType name="businessServiceKey">
  <xs:sequence>
    <xs:element name="propertyKey" type="propertyKey" maxOccurs="unbounded"/>
    <xs:element ref="propertySetBagKey" maxOccurs="unbounded"/>
  </xs:sequence>

```

```

</xs:sequence>
<xs:attribute name="serviceKey" type="serviceKey" use="optional"/>
<xs:attribute name="businessKey" type="businessKey" use="optional"/>
</xs:complexType>
<xs:element name="businessServiceKey" type="businessServiceKey"/>
<xs:complexType name="propertySetBagKey">
  <xs:sequence>
    <xs:element name="propertyKey" type="propertyKey" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="optional"/>
</xs:complexType>
<xs:element name="propertySetBagKey" type="propertySetBagKey"/>
<xs:complexType name="servicePropertyDetail">
  <xs:sequence>
    <xs:element ref="businessService" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="servicePropertyDetail" type="servicePropertyDetail">
  <xs:annotation>
    <xs:documentation>Data Structure : Result of
get_servicePropertyDetail</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="servicePropertyInfo">
  <xs:sequence>
    <xs:element ref="businessName" minOccurs="0"/>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="propertyChildren" minOccurs="0"/>
    <xs:element ref="servicePropertySets" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="serviceKey" type="serviceKey" use="required"/>
  <xs:attribute name="businessKey" type="businessKey" use="required"/>
</xs:complexType>
<xs:element name="servicePropertyInfo" type="servicePropertyInfo"/>

```

```

<xs:complexType name="servicePropertyInfos">
  <xs:sequence>
    <xs:element ref="servicePropertyInfo" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="servicePropertyInfos" type="servicePropertyInfos"/>
<xs:complexType name="servicePropertySet">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="description" minOccurs="0"/>
    <xs:element ref="propertyChildren" minOccurs="0"/>
    <xs:element ref="propertySetParent" minOccurs="0"/>
    <xs:element ref="propertySetChild" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="propertySetKey" type="propertySetKey" use="optional"/>
  <xs:attribute name="mode" type="mode" use="optional"/>
</xs:complexType>
<xs:element name="servicePropertySet" type="servicePropertySet"/>
<xs:complexType name="servicePropertySets">
  <xs:sequence>
    <xs:element ref="servicepropertySet" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="servicePropertySets" type="servicePropertySets"/>
<xs:complexType name="servicePropertyList">
  <xs:sequence>
    <xs:element ref="servicePropertyInfos"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="servicePropertyList" type="servicePropertyList">
  <xs:annotation>
    <xs:documentation>Data Structure : Result of find_serviceProperty</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="save_property">

```

```

    <xs:sequence>
      <xs:element ref="authInfo" minOccurs="0"/>
      <xs:element ref="propertyReference" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="save_property" type="save_property">
    <xs:annotation>
      <xs:documentation>API (Publishing) : save_property</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="save_propertySet">
    <xs:sequence>
      <xs:element ref="authInfo" minOccurs="0"/>
      <xs:element ref="propertySet" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="save_propertySet" type="save_propertySet">
    <xs:annotation>
      <xs:documentation>API (Publishing) : save_propertySet</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="save_serviceProperty">
    <xs:sequence>
      <xs:element ref="authInfo" minOccurs="0"/>
      <xs:element ref="businessService"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="save_serviceProperty" type="save_serviceProperty">
    <xs:annotation>
      <xs:documentation>API (Publishing) : save_serviceProperty</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="delete_property">
    <xs:sequence>
      <xs:element ref="authInfo" minOccurs="0"/>

```

```

        <xs:element name="propertyKey" type="propertyKey" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="delete_property" type="delete_property">
    <xs:annotation>
        <xs:documentation>API (Publishing) : delete_property</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="delete_propertySet">
    <xs:sequence>
        <xs:element ref="authInfo" minOccurs="0"/>
        <xs:element name="propertySetKey" type="propertySetKey" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="delete_propertySet" type="delete_propertySet">
    <xs:annotation>
        <xs:documentation>API (Publishing) : delete_propertySet</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="delete_serviceProperty">
    <xs:sequence>
        <xs:element ref="authInfo" minOccurs="0"/>
        <xs:element name="businessServiceKey" type="businessServiceKey"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="delete_serviceProperty" type="delete_serviceProperty">
    <xs:annotation>
        <xs:documentation>API (Publishing) : delete_serviceProperty</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="get_propertyDetail">
    <xs:sequence>
        <xs:element ref="authInfo" minOccurs="0"/>
        <xs:element name="propertyKey" type="propertyKey" maxOccurs="unbounded"/>

```

```

    </xs:sequence>
</xs:complexType>
<xs:element name="get_propertyDetail" type="get_propertyDetail">
    <xs:annotation>
        <xs:documentation>API (Inquiry) : get_propertyDetail</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="get_propertySetDetail">
    <xs:sequence>
        <xs:element ref="authInfo" minOccurs="0"/>
        <xs:element name="propertySetKey" type="propertySetKey" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="get_propertySetDetail" type="get_propertySetDetail">
    <xs:annotation>
        <xs:documentation>API (Inquiry) : get_propertySetDetail</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name="get_servicePropertyDetail">
    <xs:sequence>
        <xs:element ref="authInfo" minOccurs="0"/>
        <xs:element name="serviceKey" type="serviceKey" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="get_servicePropertyDetail" type="get_servicePropertyDetail">
    <xs:annotation>
        <xs:documentation>API (Inquiry) : get_servicePropertyDetail</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="findQualifier">
    <xs:simpleType>
        <xs:restriction base="findQualifier"/>
    </xs:simpleType>
</xs:element>
<xs:complexType name="findQualifiers">

```



```

    <xs:sequence>
      <xs:element ref="findQualifier" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
<xs:element name="findQualifiers" type="findQualifiers"/>
<xs:complexType name="find_property">
  <xs:sequence>
    <xs:element ref="authInfo" minOccurs="0"/>
    <xs:element ref="findQualifiers" minOccurs="0"/>
    <xs:element ref="propertyCollection" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="find_property" type="find_property">
  <xs:annotation>
    <xs:documentation>API (Inquiry) : find_property</xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="valueNode" type="xs:string"/>
<xs:complexType name="propertyCollection">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element name="valueTypeNode" type="valueType" minOccurs="0"/>
    <xs:element ref="valueNode" minOccurs="0"/>
    <xs:element name="mode" type="mode" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyCollection" type="propertyCollection"/>
<xs:complexType name="propertyCollections">
  <xs:sequence>
    <xs:element ref="findQualifiers" minOccurs="0"/>
    <xs:element ref="propertyCollection" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyCollections" type="propertyCollections"/>
<xs:complexType name="propertySetCollections">

```

```

<xs:sequence>
  <xs:element ref="findQualifiers" minOccurs="0"/>
  <xs:element ref="propertySetCollectionParent" minOccurs="0"/>
  <xs:element ref="propertySetCollectionChild" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:element name="propertySetCollections" type="propertySetCollections"/>
<xs:complexType name="propertySetCollectionParent">
  <xs:sequence>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element name="mode" type="mode" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetCollectionParent" type="propertySetCollectionParent"/>
<xs:complexType name="propertySetCollectionChild">
  <xs:sequence>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element name="mode" type="mode" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetCollectionChild" type="propertySetCollectionChild"/>
<xs:complexType name="find_propertySet">
  <xs:sequence>
    <xs:element ref="authInfo" minOccurs="0"/>
    <xs:element ref="findQualifiers" minOccurs="0"/>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element ref="propertyCollections" minOccurs="0"/>
    <xs:element ref="propertySetCollections" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="serviceKey" type="serviceKey" use="optional"/>
</xs:complexType>
<xs:element name="find_propertySet" type="find_propertySet">
  <xs:annotation>
    <xs:documentation>API (Inquiry) : find_propertySet</xs:documentation>
  </xs:annotation>

```

```

</xs:element>
<xs:complexType name="propertyValueInfo">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element name="valueTypeNode" type="valueType" minOccurs="0"/>
    <xs:element ref="typeNode" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="propertyValue" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="mode" type="mode" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyValueInfo" type="propertyValueInfo"/>
<xs:element name="typeNode" type="type"/>
<xs:complexType name="propertyValueBag">
  <xs:sequence>
    <xs:element ref="propertyValueInfo" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertyValueBag" type="propertyValueBag"/>
<xs:complexType name="propertySetValue">
  <xs:sequence>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element name="mode" type="mode" minOccurs="0"/>
    <xs:element ref="propertySetCollections" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="propertySetValue" type="propertySetValue"/>
<xs:complexType name="find_serviceProperty">
  <xs:sequence>
    <xs:element ref="authInfo" minOccurs="0"/>
    <xs:element ref="findQualifiers" minOccurs="0"/>
    <xs:element ref="name" maxOccurs="unbounded"/>
    <xs:element ref="propertyValueBag" minOccurs="0"/>
    <xs:element ref="propertySetValue" minOccurs="0"/>
  </xs:sequence>

```

```
<xs:attribute name="businessKey" type="businessKey" use="optional"/>
</xs:complexType>
<xs:element name="find_serviceProperty" type="find_serviceProperty">
  <xs:annotation>
    <xs:documentation>API (Inquiry) : find_serviceProperty</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:schema>
```



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียนวิทยานิพนธ์

นายนิปกรณ์ชัย ทুমประสิทธิ์ เกิดเมื่อวันที่ 7 กุมภาพันธ์ พ.ศ. 2514 สำเร็จ การศึกษาหลักสูตรศิลปศาสตรบัณฑิต สาขาวิชารัฐศาสตร์ คณะรัฐศาสตร์ มหาวิทยาลัย รามคำแหง ปีการศึกษา 2536 และหลักสูตรวิทยาศาสตรบัณฑิต โปรแกรมวิชาวิทยาการ คอมพิวเตอร์ สาขาวิทยาศาสตร์ สถาบันราชภัฏสวนสุนันทา ปีการศึกษา 2538-2539 ได้เข้า ทำงานที่ธนาคารแห่งประเทศไทย ตำแหน่งผู้วิเคราะห์ ในปี พ.ศ. 2540 และเข้าศึกษาต่อใน หลักสูตรวิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2543



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย