

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยต่างๆที่เกี่ยวข้อง ซึ่งเป็นพื้นฐานในการออกแบบและพัฒนา กลไกการแทนที่กันในระดับอินสแตนซ์ของบริการที่มีความสัมพันธ์แบบเท่าเทียมกันในงานวิจัยนี้

#### 2.1 แนวคิดทฤษฎี

หัวข้อต่างๆที่จะกล่าวถึงมีดังต่อไปนี้

- การเรียกใช้บริการของผู้รับบริการในคอร์บา (Service Invocation in CORBA)
- คลังจัดเก็บส่วนต่อประสานของบริการ (Interface Repository)
- บริการเทรดเดอร์ (Trading Service)

##### 2.1.1 การเรียกใช้บริการของผู้รับบริการในคอร์บา (Service Invocation in CORBA)

ก่อนที่โปรแกรมของผู้รับบริการใดๆในสถาปัตยกรรมระบบกระจายเชิงวัตถุคอร์บาจะสามารถเรียก ใช้บริการหนึ่งของผู้ให้บริการได้นั้น โปรแกรมของผู้รับบริการจะต้องกระทำขั้นตอนที่เรียกว่า Bootstrapping เพื่อให้ได้รับข้อมูลอ้างอิงของบริการ (Object Reference)<sup>3</sup> นั้นเสียก่อน (ตัวอย่างของข้อมูลอ้างอิงของบริการอยู่ในรูปที่ 2.1) โดยข้อกำหนดของคอร์บานุญาตให้การทำงานในขั้นตอนนี้สามารถ กระทำได้หลายวิธีอาทิเช่นโดยการเรียกใช้บริการเทรดเดอร์ ซึ่งเป็นวัตถุชนิดหนึ่งในคอร์บาทำหน้าที่ในการ จับคู่ระหว่างความต้องการใช้งานของผู้รับบริการ (Service Requirement) กับบริการใด ๆ ที่มีผู้มาลง ทะเบียนประกาศการให้บริการไว้ในบริการเทรดเดอร์ [8,10]

นอกจากการเรียกใช้บริการเทรดเดอร์แล้ว ผู้ผลิตซอฟต์แวร์คอร์บา (Vendor) แต่ละรายมักจะมีวิธีการเฉพาะของตนที่ได้พัฒนาขึ้นเพื่อให้โปรแกรมของผู้รับบริการสามารถค้นหาข้อมูลอ้างอิงของบริการที่ ต้องการใช้งานได้ง่ายขึ้น อาทิเช่นซอฟต์แวร์วิสิโบริคเกอร์จะใช้การค้นหาจากบริการไดเรกทอรี (Directory Service)<sup>4</sup> [5,7] ที่ถูกสร้างขึ้นโดยเฉพาะสำหรับใช้ในการค้นหาข้อมูลอ้างอิงของบริการใดๆ เมื่อโปรแกรม ของผู้รับบริการเรียกใช้คำสั่ง bind เป็นต้น การที่ผู้ผลิตซอฟต์แวร์แต่ละรายสามารถเลือกกำหนดวิธีการ

---

<sup>3</sup> Object Reference เป็นข้อมูลเฉพาะอินสแตนซ์ของบริการใดๆ (Unique identifier) ข้อกำหนดของคอร์บากำหนดให้ ข้อมูลนี้มีรูปแบบที่เป็นมาตรฐานและไม่ขึ้นกับผู้ผลิตซอฟต์แวร์คอร์บาแต่ละราย ข้อมูลนี้บางครั้งถูกเรียกว่า IOR หรือ Interoperable Object Reference

<sup>4</sup> Directory Service นี้มีชื่อเรียกว่า สมาร์ทเอเจนต์ (Smart agent)

เฉพาะของตนขึ้นใช้งานได้นั้นเป็นเพราะข้อกำหนดคอร์บายังไม่ได้กำหนดให้ใช้วิธีการใดวิธีการหนึ่งโดยเฉพาะนั่นเอง

```
// ตัวอย่างข้อมูลอ้างอิงของบริการ (IOR) ตามข้อกำหนดของคอร์บาย
IOR:00000000000000002249444c3a6f6d672e6f72672f436f7354726164696e672f4c6f6f6b7570
3a312e300000000000000010000000000000370001000000000018636f7262617372762e616
5726f746861692e6f722e74680013880000000000f54726164696e675365727669636500

// ความหมายจากการถอดรหัสข้อมูลอ้างอิงของบริการ
Interoperable Object Reference
Type ID: IDL:Bank/AccountManager:1.0
Contains 1 profile.
Profile 0-IIOP Profile:
Version: 1.0
Host: 207.92.87.52
Port: 1067
Object Key: PersistentId [repld=IDL:Bank/AccountManager:1.0,objectName=dallas]
```

รูปที่ 2.1 ตัวอย่างข้อมูลอ้างอิงของบริการตามข้อกำหนดของคอร์บาย

คำสั่ง bind จะถูกกำหนดอยู่ในแฟ้มข้อมูลตัวช่วย (Helper file) ของซอฟต์แวร์วิสิโบริคเกอร์ ซึ่งแฟ้มข้อมูลตัวช่วยนี้จะถูกสร้างขึ้นจากการคอมไพล์แฟ้มข้อมูลไอดีแอล (IDL File<sup>5</sup>) ซึ่งเป็นแฟ้มข้อมูลที่ใช้จัดเก็บส่วนต่อประสานของบริการที่มีผู้เสนอให้บริการในเครือข่าย รูปแบบการเรียกใช้งานคำสั่ง bind มีทั้งสิ้น 3 รูปแบบ [5] (รายละเอียดอยู่ในรูปที่ 2.2)

เมื่อโปรแกรมของผู้รับบริการเรียกใช้คำสั่ง bind จะส่งผลให้ซอฟต์แวร์ระดับล่างเกิดการดำเนินงานตามขั้นตอนต่างๆ ดังต่อไปนี้

1. ออร์บีในฝั่งของโปรแกรมผู้รับบริการจะพยายามติดต่อกับสมาร์ทเอเจนท์โดยใช้การส่งข้อมูลแบบกระจาย (Broadcast) ไปบนเครือข่ายด้วยโพรโตคอลแบบยูดีพี (UDP) เพื่อค้นหาตำแหน่งของสมาร์ทเอ

<sup>5</sup> IDL (Interface Definition Language) File เป็นแฟ้มข้อมูลที่ใช้จัดเก็บนิยามส่วนต่อประสานของบริการหนึ่งในคอร์บาย โดยแฟ้มข้อมูลนี้จะถูกคอมไพล์เพื่อใช้สร้างโปรแกรมในส่วนของสตับ (Stub) และสเคเลตัน (Skeleton) สำหรับใช้งานร่วมกับโปรแกรมของผู้ให้บริการและผู้รับบริการ

เจนท์และเมื่อได้รับข้อมูลจากสมาร์ทเอเจนท์ตอบกลับมา ออร์บในฝั่งของโปรแกรมผู้รับบริการจะสร้างการเชื่อมต่อไปยังสมาร์ทเอเจนท์ที่พบในรูปของการสื่อสารแบบจุดต่อจุดด้วยโพรโตคอลแบบยูดีพี (UDP Point-to-Point Communication)

2. โปรแกรมของผู้รับบริการจะจัดส่งค่าพารามิเตอร์ต่างๆที่ระบุมาในคำสั่ง bind ไปทำการค้นหาอินสแตนซ์ของบริการต่างๆที่มาลงทะเบียนไว้ในสมาร์ทเอเจนท์ที่ติดต่อด้วย เมื่อพบบริการที่ต้องการใช้งานลงทะเบียนอยู่ในสมาร์ทเอเจนท์ ออร์บของโปรแกรมผู้รับบริการจะทำการสร้างวัตถุพร็อกซีท้องถิ่น (Local Proxy Object) ขึ้นตามข้อมูลอ้างอิงของบริการที่ได้รับจากสมาร์ทเอเจนท์ ทั้งนี้วัตถุพร็อกซีท้องถิ่นจะถูกใช้ในการรับส่งข้อมูล (Data Marshaling) ระหว่างโปรแกรมของผู้รับบริการกับอินสแตนซ์จริงของบริการซึ่งทำงานอยู่ภายในโปรแกรมของผู้ให้บริการ หลังจากนั้นโปรแกรมของผู้รับบริการจึงจะได้รับข้อมูลอ้างอิงของบริการจากออร์บเพื่อนำไปใช้งานในขั้นตอนอื่นต่อไป แต่ถ้าภายในสมาร์ทเอเจนท์ที่ติดต่อด้วยไม่มีบริการที่ต้องการ สมาร์ทเอเจนท์จะนำค่าพารามิเตอร์ต่างๆตามคำสั่ง bind ไปทำการค้นหาข้อมูลอ้างอิงของบริการยังสมาร์ทเอเจนท์อื่นๆบนเครือข่ายโดยอัตโนมัติ ในท้ายที่สุดถ้าไม่สามารถค้นหาบริการที่ต้องการใช้งานได้ ออร์บของคอร์บาจะทำให้โปรแกรมของผู้รับบริการเกิดเอ็็กซ์เซพชันแบบ org.omg.CORBA.NO\_IMPLEMENTATION ขึ้น

1. Object Reference <Interface name>Helper.bind(org.omg.CORBA.ORB orb)
2. Object Reference <Interface name>Helper.bind(org.omg.CORBA.ORB orb, String Name)
3. Object Reference <Interface name>Helper.bind(org.omg.CORBA.ORB orb, String Name,String Host, org.omg.CORBA.BindOptions Options)

#### คำอธิบาย

*Name* หมายถึง ชื่ออินสแตนซ์ของบริการใดๆ

*Host* หมายถึง ชื่อโฮสต์ซึ่งมีอินสแตนซ์ของบริการที่ต้องการใช้งานทำงานอยู่

*Options* หมายถึง ทางเลือกที่ใช้ระบุลักษณะการเชื่อมต่อระหว่างโปรแกรมของผู้รับบริการและผู้ให้บริการ เช่น คุณลักษณะ *defer\_bind* ในทางเลือกนี้จะถูกใช้ในการกำหนดให้ออร์บสร้างการเชื่อมต่อไปยังอินสแตนซ์ของบริการ เมื่อโปรแกรมผู้รับบริการเรียกใช้เมทอดของบริการเป็นครั้งแรก เป็นต้น

รูปที่ 2.2 รูปแบบของคำสั่ง bind ทั้ง 3 แบบในแฟ้มข้อมูลตัวช่วยของซอฟต์แวร์วิสิโรรคเกอร์

ในการระบุค่าพารามิเตอร์ตามคำสั่ง bind รูปแบบที่ 1 จะส่งผลให้เกิดการค้นหาอินสแตนซ์ใดๆของบริการที่พัฒนาขึ้นตามส่วนต่อประสานที่ระบุ ซึ่งถ้าหาไม่พบออร์บจะทำการค้นหาอินสแตนซ์อื่นที่มีส่วนต่อประสานเป็นซับไทป์ของส่วนต่อประสานที่ระบุเพื่อใช้งานแทนที่ตามหลักการแทนที่กันในระดับชนิดของบริการ แต่ถ้าโปรแกรมผู้รับบริการเรียกใช้คำสั่ง bind รูปแบบที่ 2 โดยทำการระบุชื่อของอินสแตนซ์ที่ต้องการใช้งานมาในคำสั่งด้วย ออร์บจะส่งกลับข้อมูลอ้างอิงของบริการที่พัฒนาขึ้นตามส่วนต่อประสานและมีชื่อของอินสแตนซ์ตามที่ระบุมาเท่านั้น และถ้าเลือกใช้คำสั่ง bind รูปแบบที่ 3 ซึ่งมีการระบุค่าพารามิเตอร์ที่แสดงถึงชื่อโฮสต์ (Host) ที่บริการนั้นทำงานอยู่ด้วย ออร์บจะทำการค้นหาเฉพาะอินสแตนซ์ของบริการที่ทำงานในโฮสต์ที่ระบุมาเท่านั้น ดังนั้นจึงสรุปได้ว่าเมื่อโปรแกรมของผู้รับบริการใช้คำสั่ง bind รูปแบบที่ 2 และ 3 จะไม่ก่อให้เกิดกลไกการทำงานแทนที่กันของบริการทั้งในระดับชนิดและในระดับอินสแตนซ์ของบริการ [5,7]

ภายหลังจากที่โปรแกรมของผู้รับบริการได้รับข้อมูลอ้างอิงของบริการที่ต้องการใช้งานแล้ว โปรแกรมของผู้รับบริการจะสามารถเรียกใช้งานอินสแตนซ์ของบริการนั้นๆได้ตามเมทอดที่กำหนดไว้ในส่วนต่อประสานโดยใช้การส่งผ่านค่าพารามิเตอร์ต่างๆไปประมวลผลยังอินสแตนซ์ของบริการนั้นๆ ตามตัวอย่างการเขียนโปรแกรมในรูปที่ 2.3

```
// โปรแกรมของผู้รับบริการกำหนดค่าโดยปริยายให้ออร์บและทำการเรียกใช้คำสั่ง bind จากเพิ่ม
ข้อมูลตัวช่วยเพื่อค้นหาข้อมูลอ้างอิงของบริการที่มีส่วนต่อประสานเป็น 'Bank.AccountManager'
และมีชื่ออินสแตนซ์ของบริการเป็น 'dallas' จากสมาร์ทเอเจนท์
org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(orgs,null);
Bank.AccountManager manager = Bank.AccountManagerHelper.bind(orb,"dallas");

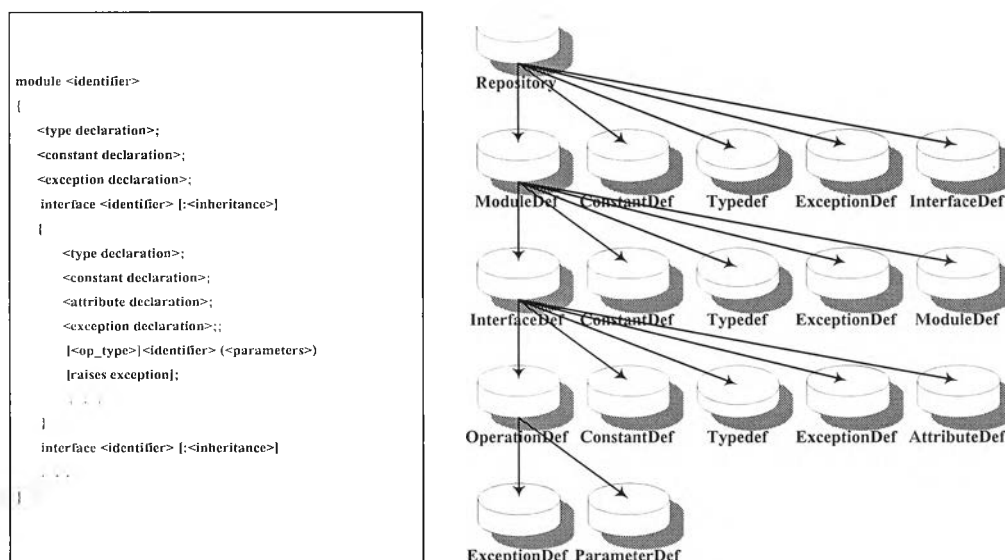
// โปรแกรมของผู้รับบริการได้รับข้อมูลอ้างอิงของบริการ 'dallas' และจัดเก็บลงในตัวแปร manager
หลังจากนั้นจึงเรียกใช้เมทอด open ของบริการ 'dallas' โดยให้การส่งผ่านข้อมูลแบบสายอักขระเป็น
พารามิเตอร์อินพุตตามรูปแบบในส่วนต่อประสาน Bank.AccountManager ไปยังอินสแตนซ์ของ
บริการ 'dallas'
// โปรแกรมของผู้รับบริการได้รับผลการทำงานของบริการ 'dallas' และจัดเก็บลงในตัวแปร account
Bank.Account account = manager.open("john");
```

รูปที่ 2.3 ตัวอย่างการเรียกใช้บริการของโปรแกรมผู้รับบริการในคอร์บา

### 2.1.2 คลังจัดเก็บส่วนต่อประสานของบริการ (Interface Repository)

คลังจัดเก็บส่วนต่อประสานหรือไออาร์ (IR – Interface Repository) [1,2] เป็นวัตถุชนิดหนึ่งในคอร์บาที่ทำหน้าที่เป็นเสมือนคลังที่ใช้ในการจัดเก็บนิยามส่วนต่อประสาน (Interface Definition) ของบริการต่างๆ โดยในการจัดเก็บข้อมูลนิยามส่วนต่อประสานนั้น คอร์บากำหนดให้คลังใช้การจัดเก็บแบบลำดับชั้นของการบรรจุ (Containment Hierarchy) ตัวอย่างเช่น สมมติให้ภายในโมดูล (Module) หนึ่งๆ ของนิยามส่วนต่อประสานที่ผู้พัฒนาโปรแกรมกำหนดขึ้นอาจจะประกอบไปด้วยนิยามของส่วนต่อประสาน (Interface) หลายๆ ส่วนต่อประสาน โดยที่ภายในส่วนต่อประสานแต่ละส่วนต่อประสานก็อาจจะประกอบไปด้วยนิยามของการดำเนินงาน (Operation) หลายๆ การดำเนินงานและภายในแต่ละการดำเนินงานก็อาจจะประกอบไปด้วยนิยามของชนิดพารามิเตอร์และเอ็กซ์เซพชันต่างๆ ที่อาจเกิดขึ้นจากการใช้งาน เป็นต้น

ดังนั้นภายในวัตถุที่ทำหน้าที่เป็นคลังจัดเก็บส่วนต่อประสานของคอร์บาจึงถูกใช้จัดเก็บวัตถุโมดูลเดฟ (ModuleDef) เพื่อใช้แทนข้อมูลนิยามในส่วนของโมดูลและภายในวัตถุโมดูลเดฟนี้ก็จะถูกใช้จัดเก็บวัตถุอินเตอร์เฟซเดฟ (InterfaceDef) เพื่อใช้แทนข้อมูลนิยามในส่วนของส่วนต่อประสาน นอกจากนี้ภายในวัตถุอินเตอร์เฟซเดฟก็ยังคงถูกใช้จัดเก็บข้อมูลวัตถุโอเปอเรชันเดฟ (OperationDef) เพื่อใช้แทนข้อมูลนิยามของการดำเนินงานเป็นลำดับชั้นอย่างนี้เรื่อยไป ตามรูปที่ 2.4



รูปที่ 2.4 ลำดับชั้นของการบรรจุในคลังจัดเก็บส่วนต่อประสานของคอร์บา

โปรแกรมของผู้รับบริการจะสามารถใช้งานคลังจัดเก็บส่วนต่อประสานของคอร์บาเพื่อทำการร้องขอข้อมูลเกี่ยวกับนิยามของส่วนต่อประสานที่ถูกจัดเก็บอยู่ภายในคลังได้ผ่านเมทอดต่างๆเช่น

- Describe เป็นเมทอดที่ใช้ในการแสดงรายละเอียด (Description) ของวัตถุที่ถูกบรรจุ (Contained Object) ภายในวัตถุบรรจุ (Container Object)
- Lookup\_name เป็นเมทอดที่ใช้ในการค้นหาวัตถุภายในคลังจากชื่อ
- Describe\_interface เป็นเมทอดที่ใช้ในการแสดงรายละเอียดทางโครงสร้าง (Structure) ของส่วนต่อประสานในวัตถุอินเตอร์เฟสเดฟ เป็นต้น

นอกเหนือจากการจัดเก็บข้อมูลนิยามส่วนต่อประสานของบริการเป็นลำดับชั้นตามที่กล่าวถึงข้างต้นแล้ว คลังจัดเก็บส่วนต่อประสานของคอร์บายังมีข้อมูลที่ใช้แสดงความสัมพันธ์แบบซับซ้อนระหว่างบริการสองบริการใดๆอีกด้วย ส่งผลให้คอร์บาสามารถรองรับการทำงานแทนที่กันในระดับชนิดของบริการได้

อย่างไรก็ตาม งานวิจัย [4] มีแนวความคิดที่จะทำการเพิ่มส่วนขยายให้กับคลังจัดเก็บส่วนต่อประสานในคอร์บาสำหรับรองรับข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างบริการ ซึ่งจะส่งผลให้คอร์บามีข้อมูลเพียงพอสำหรับรองรับการทำงานแทนที่กันระหว่างอินสแตนซ์ของสองบริการใดๆที่มีความสามารถในการทำงานเท่าเทียมกันได้ รายละเอียดส่วนขยายของคลังจัดเก็บส่วนต่อประสานตามงานวิจัย [4] อยู่ในหัวข้อที่ 2.2.2

### 2.1.3 บริการเทรดเดอร์ (Trading Service)

คอร์บาใช้แนวความคิดของบริการเทรดเดอร์เพื่อสนับสนุนการเรียกใช้งานบริการใดๆในลักษณะพลวัต (Dynamic) กล่าวคือเทรดเดอร์จะถูกใช้เป็นตัวกลางที่เป็นเสมือนตัวกลางเชื่อมโยงการรับและให้บริการระหว่างผู้รับบริการและผู้ให้บริการใดๆในสถาปัตยกรรมระบบแบบกระจายเชิงวัตถุ

กระบวนการทำงานของเทรดเดอร์ [8,10] จะเริ่มจากการที่เทรดเดอร์ได้รับข้อเสนอบริการ (Service Offers) จากเอ็กซ์พอร์ตเตอร์ (Exporter) หรืออีกนัยหนึ่งคือผู้ให้บริการ เพื่อประกาศว่าผู้ให้บริการมีความพร้อมที่จะเสนอบริการรูปแบบใดให้กับผู้ที่ต้องการ เอ็กซ์พอร์ตเตอร์จะทำการเอ็กซ์พอร์ตบริการไปยังเทรดเดอร์โดยเรียกใช้เมทอด Export ตามรูปแบบที่กำหนดไว้ในส่วนต่อประสาน Register ของเทรดเดอร์

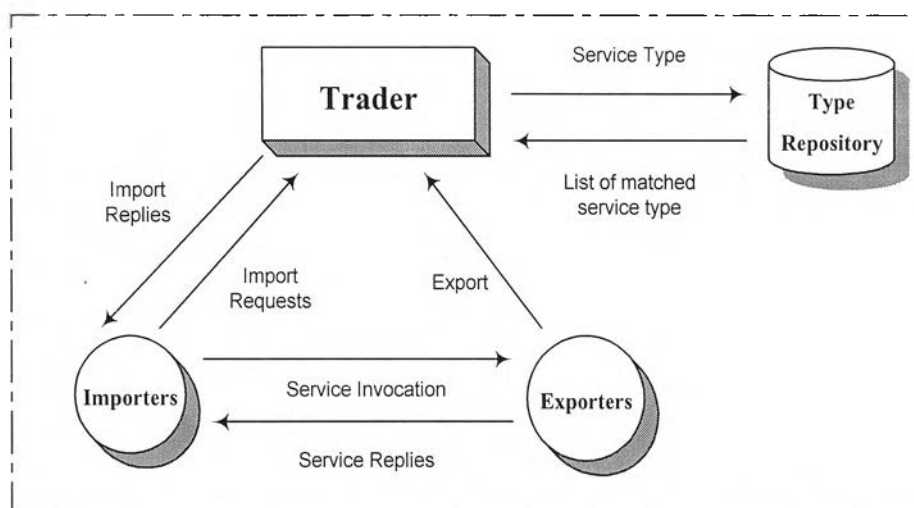
ข้อเสนอบริการที่เทรดเดอร์ได้รับมานี้จะประกอบไปด้วยข้อมูลชนิดของบริการ (Service Type) ข้อมูลส่วนต่อประสานของบริการ (Interface) ตลอดจนรายละเอียดเกี่ยวกับคุณสมบัติของบริการ (Service Properties) ที่พร้อมจะทำงาน ภายหลังจากที่เทรดเดอร์ได้รับข้อเสนอบริการนั้นแล้ว เเทรดเดอร์จะทำการตรวจสอบความถูกต้องของชนิดบริการในข้อเสนอกับคลังชนิดของบริการ (Type Repository) ภายในเทรดเดอร์ก่อนที่จะทำการจัดเก็บข้อเสนอบริการนี้ลงในฐานข้อมูลภายในเทรดเดอร์ (Trader Database) ต่อไป

กระบวนการคัดเลือกข้อเสนอบริการของเทรดเดอร์ (Service Selection) เป็นดังรูปที่ 2.5 โดยจะเริ่มจากการที่เทรดเดอร์ได้รับคำร้องขอใช้บริการจากอิมพอร์ตเตอร์ (Importer) หรืออีกนัยหนึ่งคือผู้ขอรับบริการ ซึ่งคำร้องขอใช้บริการ (Service Request) หนึ่งๆจะบรรยายละเอียดเกี่ยวกับข้อมูลชนิดของบริการ (Service Type) และข้อมูลคุณสมบัติของบริการที่อิมพอร์ตเตอร์ต้องการใช้งาน (Constraint) นอกจากนี้แล้วอิมพอร์ตเตอร์ยังอาจจะระบุข้อมูลลำดับความต้องการ (Preference) ตลอดจนข้อมูลการกำหนดขอบเขตนโยบายในการค้นหาของเทรดเดอร์ (Searching Policies) ได้อีกด้วย อนึ่งอิมพอร์ตเตอร์จะทำการอิมพอร์ตคำร้องขอใช้บริการใดๆมายังเทรดเดอร์ผ่านทางเมทอด query ในส่วนต่อประสาน Lookup ของเทรดเดอร์

เทรดเดอร์จะใช้ข้อมูลภายในคลังชนิดของบริการเพื่อตรวจสอบว่าเทรดเดอร์รู้จักชนิดของบริการที่อิมพอร์ตเตอร์ต้องการหรือไม่ ตลอดจนทำการตรวจสอบคุณสมบัติของบริการที่ระบุว่าจะมีความถูกต้องหรือไม่ ก่อนที่เทรดเดอร์จะเริ่มกระบวนการค้นหาและจับคู่ (Matchmaking) ระหว่างข้อมูลที่ได้รับจากทั้งเอ็กซ์พอร์ตเตอร์และอิมพอร์ตเตอร์เพื่อคัดเลือกกลุ่มของข้อเสนอบริการที่เหมาะสมจากข้อมูลภายในฐานข้อมูล

ในขั้นตอนการค้นหาข้อมูลจากฐานข้อมูลนั้น เเทรดเดอร์จะใช้ข้อมูลความสัมพันธ์แบบซับซ้อนระหว่างชนิดของบริการจากคลังชนิดของบริการมาประกอบการตัดสินใจเลือกด้วย ทั้งนี้เพื่อให้การเรียกใช้งานบริการเทรดเดอร์สามารถรองรับการค้นหาไปยังข้อเสนอของบริการอื่นๆซึ่งมีชนิดของบริการเป็นซับซ้อนของชนิดบริการที่อิมพอร์ตเตอร์ต้องการได้ อย่างไรก็ตาม อิมพอร์ตเตอร์จะสามารถระบุให้บริการเทรดเดอร์คัดเลือกเฉพาะกลุ่มข้อเสนอของบริการที่มีชนิดตามชนิดบริการที่ระบุมาเท่านั้นก็ได้ (คือไม่ต้องการชนิดของบริการที่เป็นซับซ้อน) โดยใช้การระบุค่า exact\_type\_match ให้เป็นจริง (True) ในช่วงของการกำหนดนโยบายในการอิมพอร์ตบริการ [10]

หลังจากที่อิมพอร์ตเตอร์ได้รับกลุ่มของข้อเสนอบริการที่ผ่านการคัดเลือกแล้ว อิมพอร์ตเตอร์จะสามารถติดต่อและเรียกใช้งานบริการเหล่านั้นได้โดยใช้ข้อมูลอ้างอิงของบริการภายในข้อเสนอบริการที่ได้รับมาในการสร้างการเชื่อมต่อ (Binding) เข้ากับอินสแตนซ์ของบริการนั้นๆเพื่อเรียกใช้งานต่อไป



รูปที่ 2.5 องค์ประกอบและกระบวนการทำงานของบริการเทรดเดอร์

ปัจจุบันนี้การคัดเลือกบริการอื่นที่สามารถทำงานแทนที่บริการที่อิมพอร์ตเตอร์ระบุมาของเทรดเดอร์ จะรองรับเพียงการคัดเลือกบริการที่มีชนิดของบริการเป็นแบบซัพไทป์ปิงของบริการที่ระบุมาเท่านั้น

การทำงานของเทรดเดอร์และคลังชนิดของบริการยังมีข้อมูลที่ไม่เพียงพอสำหรับรองรับการคัดเลือกบริการใดๆที่มีความสัมพันธ์แบบเท่าเทียมกันมาใช้งานทดแทนกัน [10]

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 งานวิจัย Evolution Transparency for Distributed Service Types [9]

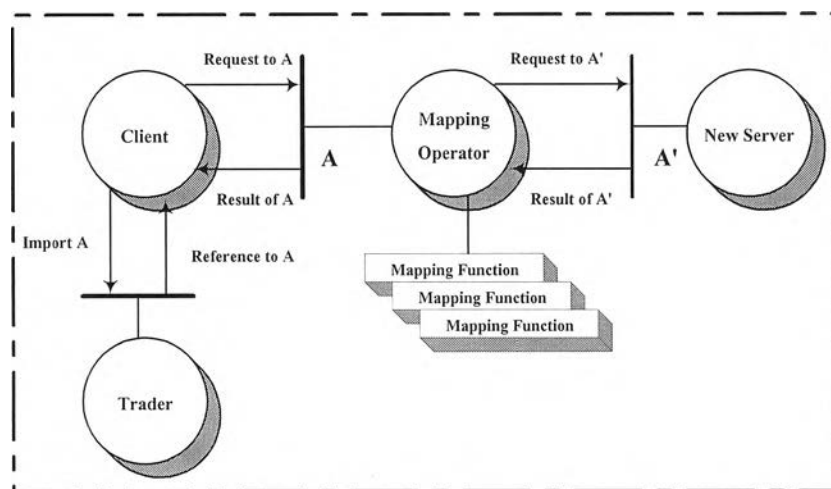
งานวิจัย [9] นี้ได้นำเสนอแบบจำลองเพื่อสร้างความโปร่งใสในการเปลี่ยนรุ่นของบริการ โดยบริการรุ่นใหม่ไม่ถูกจำกัดให้อยู่ในรูปแบบที่เป็นซัพไทป์ปิงของบริการรุ่นเดิมเท่านั้น ตามแบบจำลองของงานวิจัย [9] นี้ (ดังรูปที่ 2.6) จะใช้การสร้างตัวดำเนินการแปลง (Mapping Operator) ขึ้นมาแบบกึ่งอัตโนมัติเมื่อเกิดการเปลี่ยนรุ่นของบริการ โดยผู้ทำการเปลี่ยนรุ่นของบริการจะต้องเป็นผู้รับผิดชอบในการกำหนดข้อมูลฟังก์ชันการแปลง (Mapping Function) ข้อมูลฟังก์ชันการแปลงจะไม่สามารถถูกกำหนดขึ้นโดยอัตโนมัติได้ทั้งนี้เนื่องจากบริการรุ่นใหม่สามารถถูกทำให้เปลี่ยนแปลงไปจากบริการรุ่นเดิมได้อย่างอิสระ<sup>6</sup>

ตัวดำเนินการแปลงที่ถูกสร้างขึ้นจะทำการแปลงคำร้องขอใช้บริการจากผู้รับบริการตามส่วนต่อประสานของบริการรุ่นเก่าให้สอดคล้องกับส่วนต่อประสานของบริการรุ่นใหม่ ซึ่งจะส่งผลให้โปรแกรมของ

<sup>6</sup> การเปลี่ยนรุ่นของบริการอย่างอิสระ หมายถึงการเปลี่ยนรุ่นของบริการที่สามารถเปลี่ยนแปลงได้ทั้งความหมาย (Semantics) และรูปแบบการเรียกใช้งาน (Syntax)



ผู้รับบริการสามารถทำงานต่อไปได้โดยผู้รับบริการไม่จำเป็นต้องทำการเปลี่ยนแปลงโปรแกรมที่ใช้งานอยู่ในทันทีที่ส่วนต่อประสานของบริการถูกเปลี่ยนรูปไปจากเดิม ตัวดำเนินการแปลงจะถูกสร้างเข้าแทนที่บริการรุ่นเดิมโดยถูกพัฒนาให้มีชื่อของส่วนต่อประสานและชื่อของอินสแตนซ์ที่เหมือนกับอินสแตนซ์ของบริการรุ่นเก่าทุกประการ ทั้งนี้เพื่อใช้คำร้องขอใช้บริการจากโปรแกรมผู้รับบริการรุ่นเดิมได้โดยอัตโนมัติ หนึ่งงานวิจัย [9] นี้ได้ตั้งข้อสมมติฐานที่ว่าอินสแตนซ์ของบริการรุ่นเก่าจะถูกยกเลิกไม่ใช้งานอีกเมื่อผู้ให้บริการนำบริการรุ่นใหม่มาใช้แทนที่ ดังนั้นตัวดำเนินการแปลงจึงสามารถถูกสร้างให้มีชื่อส่วนต่อประสานและอินสแตนซ์เหมือนกับบริการรุ่นเก่าได้



รูปที่ 2.6 การทำงานของตัวดำเนินการแปลงและฟังก์ชันการแปลงตามงานวิจัย [9]

การทำงานแทนที่กันของบริการในงานวิจัย [9] ได้ออกแบบให้อินสแตนซ์หนึ่งของบริการรุ่นเก่าถูกแทนที่ด้วยอีกอินสแตนซ์หนึ่งของบริการรุ่นใหม่ได้เท่านั้น ดังนั้นผู้ทำการเปลี่ยนรุ่นจึงจะต้องทำการกำหนดคู่ของอินสแตนซ์ที่จะทำงานแทนที่กัน โดยอินสแตนซ์ใดๆของบริการรุ่นเก่าอาจจะไม่สามารถถูกใช้แทนที่ด้วยอินสแตนซ์ใดๆก็ได้ของบริการรุ่นใหม่ อย่างไรก็ตาม งานวิจัย [9] นี้ยังไม่ได้คำนึงถึงการจัดเก็บข้อมูล ฟังก์ชันการแปลงให้เป็นระบบ ข้อมูลส่วนนี้ยังถูกจัดเก็บไว้ในแฟ้มข้อมูลต่างๆอย่างกระจัดกระจายในระบบแฟ้มข้อมูลของระบบทั้งๆที่ควรจะถูกจัดเก็บรวบรวมไว้เป็นส่วนหนึ่งของข้อมูลที่เกี่ยวข้องกับบริการภายในคลังจัดเก็บส่วนต่อประสาน

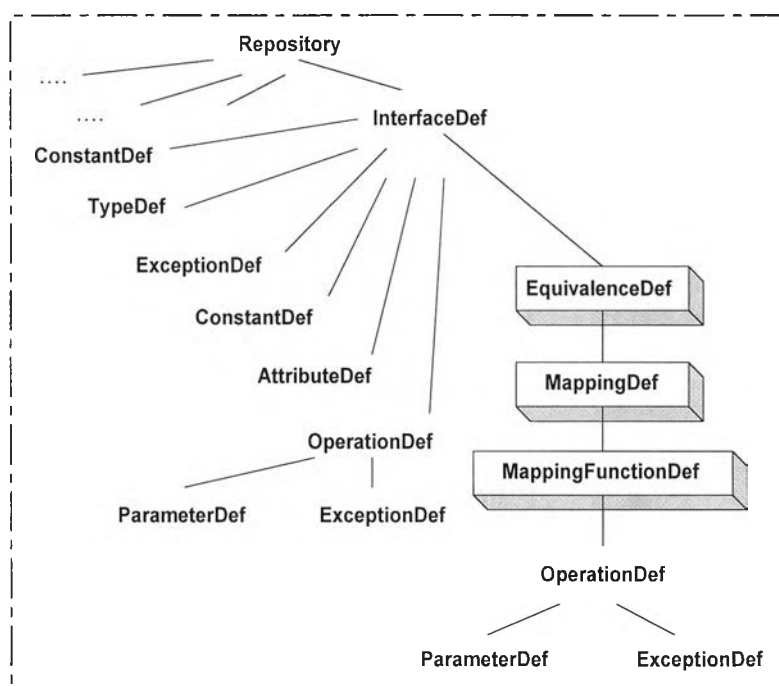
งานวิจัยนี้มีแนวความคิดที่จะขยายขอบเขตการใช้งานกลไกของงานวิจัย [9] ให้ครอบคลุมไปถึงการแทนที่ของสองบริการใดๆที่มีความสัมพันธ์แบบเท่าเทียมกันโดยไม่จำเป็นต้องเป็นรุ่นของกันและกันได้ด้วย อย่างไรก็ตามงานวิจัยนี้จะไม่สามารถใช้งานกลไกของงานวิจัย [9] ในรูปแบบของการแทนที่ดังกล่าวนี้ได้โดยทันที ผู้รับบริการจะประสบปัญหาเกี่ยวกับการค้นหาและเรียกใช้งานตัวดำเนินการแปลงทั้งนี้เนื่องจากผู้ให้บริการแทนที่จะไม่สามารถปลอมตัวดำเนินการแปลงให้มีชื่อของส่วนต่อประสาน

และชื่อของอินสแตนซ์ที่เหมือนกับบริการที่จะถูกแทนที่ได้ (บริการที่จะถูกแทนที่ยังคงทำงานอยู่ไม่ได้ถูกยกเลิกไป) ด้วยข้อจำกัดของการใช้งานกลไกในสถานการณ์ดังกล่าวทำให้งานวิจัยนี้มีความจำเป็นที่จะต้องเพิ่มกลไกการทำงานซึ่งจะช่วยทำให้ผู้รับบริการสามารถเรียกใช้งานตัวดำเนินการแปลงเพื่อทำหน้าที่แปลงและส่งผ่านคำร้องไปยังอินสแตนซ์ของบริการที่จะถูกเรียกใช้งานแทนที่ได้ต่อไป

## 2.2.2 งานวิจัยการออกแบบและพัฒนาคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน [4]

งานวิจัยนี้เป็นงานวิจัยที่สืบเนื่องจากงานวิจัย [9] โดยมุ่งประเด็นไปในเรื่องของการจัดสร้างส่วนขยายให้กับคลังจัดเก็บส่วนต่อประสานในคอร์บาเพื่อให้คลังนี้สามารถรองรับการจัดเก็บข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างบริการได้

ข้อมูลนิยามส่วนต่อประสานของบริการภายในคลังจัดเก็บส่วนต่อประสานจะถูกเพิ่มเติมให้มีรายละเอียดของวัตถุคิควาเลนซ์เดฟ (EquivalenceDef) แมปปีงเดฟ (MappingDef) และแมปปีงฟังก์ชันเดฟ (MappingFunctionDef) ตามรูปที่ 2.7



รูปที่ 2.7 ส่วนขยายของคลังจัดเก็บส่วนต่อประสานตามงานวิจัย [4]

จากรูปงานวิจัย [4] จะใช้วัตถุคิควาเลนซ์เดฟในการแสดงรายละเอียดของข้อมูลแสดงความเท่าเทียมกันของบริการ (Equivalence Information) เพื่อระบุว่าวัตถุอินเทอร์เฟซเดฟหนึ่งจะมีรูปแบบ

ความสัมพันธ์ที่เท่าเทียมกันกับอีกวัตถุอินเทอร์เฟซเดฟหนึ่งที่ถูกกำหนดขึ้นได้อย่างไร ส่วนขยายนี้จะใช้วัตถุแมปปีงเดฟทำหน้าที่จัดเก็บข้อมูลกลุ่มของฟังก์ชันการแปลงซึ่งเป็นฟังก์ชันที่ใช้ในการแปลงคำร้องขอใช้บริการของตัวดำเนินการแปลง นอกจากนี้แล้วในแต่ละฟังก์ชันการแปลงก็ยังคงมีการจัดเก็บรายละเอียดด้วยการใช้วัตถุแมปปีงฟังก์ชันเดฟ ซึ่งภายในวัตถุนี้จะประกอบด้วยวัตถุโอเปอเรชันเดฟสำหรับแสดงรูปแบบของการดำเนินงาน วัตถุพารามิเตอร์เดฟและเอ็กซ์เซพชันเดฟสำหรับแสดงพารามิเตอร์ต่างๆ สำหรับการเรียกใช้บริการและเอ็กซ์เซพชันต่างๆที่อาจเกิดขึ้นในการทำงานเป็นลำดับขั้น

งานวิจัยนี้จะใช้ข้อมูลความสัมพันธ์แบบเท่าเทียมกันของบริการ ซึ่งถูกจัดเก็บอยู่ในคลังจัดเก็บส่วนต่อประสานของงานวิจัย [4] เพื่อช่วยในการค้นหาบริการที่ผู้รับบริการสามารถเรียกใช้งานแทนที่บริการที่ไม่สามารถให้บริการได้โดยข้อมูลนี้จะถูกเรียกใช้โดยอัตโนมัติและจะส่งผลให้เกิดความโปร่งใสต่อการแทนที่กันของบริการ ทั้งนี้เนื่องจากผู้รับบริการไม่มีความจำเป็นที่จะต้องทราบล่วงหน้าว่าบริการแทนที่คือบริการใด การทำงานแทนที่กันของบริการจะเกิดขึ้นเองในขณะรันไทม์โดยผู้ให้บริการแทนที่จะเป็นผู้รับผิดชอบทำหน้าที่ในการกำหนดและจัดเก็บข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างบริการลงในคลังจัดเก็บส่วนต่อประสานเพื่อพร้อมให้เรียกใช้งานได้ต่อไป