

บทที่ 4

การออกแบบและพัฒนาส่วนแก้ไขของบริการเทรดเดอร์ในคอร์บา ให้รองรับการเรียกใช้งานบริการที่เท่าเทียมกัน

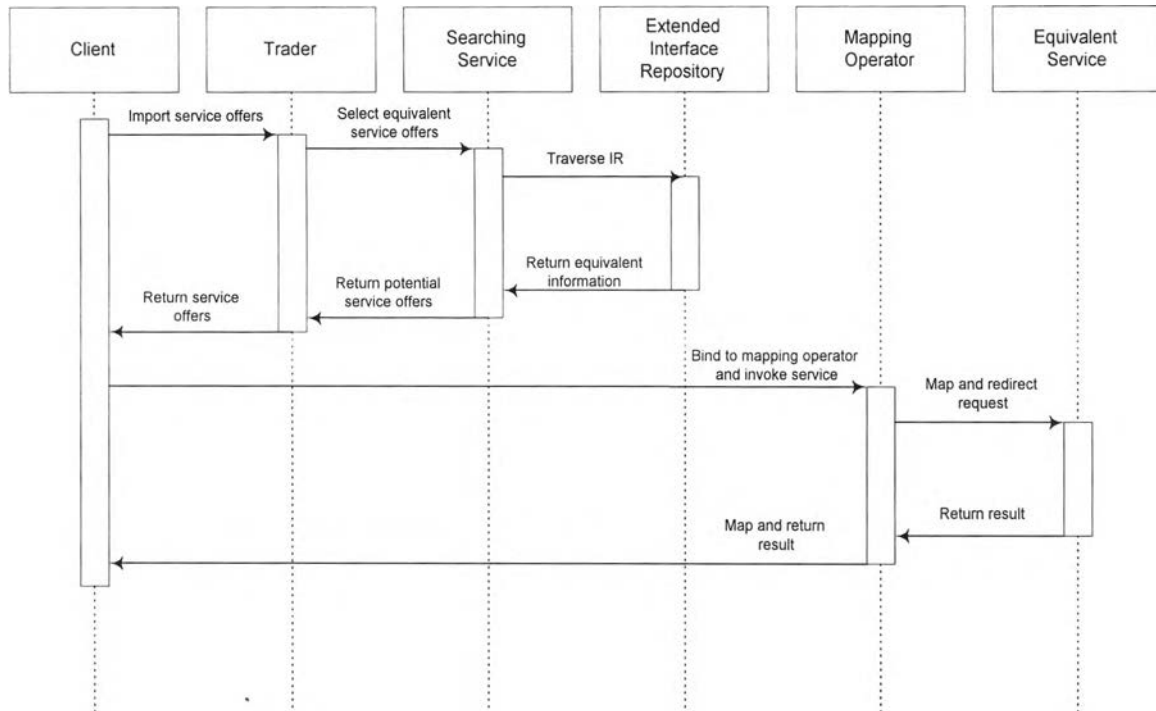
ในบทนี้จะขอก้าวถึงรายละเอียดของงานวิจัยนี้ในส่วนที่เกี่ยวข้องกับการแก้ไขบริการเทรดเดอร์ในคอร์บาให้รองรับการเรียกใช้งานข้อมูลความสัมพันธ์แบบเท่าเทียมกันของบริการภายในคลังจัดเก็บส่วนต่อประสานของงานวิจัย [4]

4.1 บริการเทรดเดอร์ (Trading Service)

การพัฒนาโปรแกรมพีโรเซสเซอร์เพื่อใช้แทรกคำสั่งการค้นหาบริการแทนที่จากคลังจัดเก็บส่วนต่อประสานลงในโปรแกรมของผู้รับบริการจัดเป็นกลไกการทำงานหนึ่งของงานวิจัยนี้ที่สามารถแก้ไขปัญหาการแทนที่ของบริการได้ในกรณีที่โปรแกรมของผู้รับบริการทำการค้นหาข้อมูลอ้างอิงของบริการใดๆผ่านทางออร์บด้วยการเรียกใช้คำสั่ง bind เท่านั้น โปรแกรมพีโรเซสเซอร์นี้จะไม่สามารถใช้แก้ไขปัญหาได้หากโปรแกรมของผู้รับบริการไม่ได้เรียกใช้คำสั่ง bind แต่ทำการค้นหาอินสแตนซ์ของบริการโดยเรียกใช้บริการจากเทรดเดอร์ งานวิจัยนี้จึงมีเป้าหมายที่จะแก้ไขโปรแกรมของบริการเทรดเดอร์ให้มีความสามารถในการนำข้อมูลความสัมพันธ์แบบเท่าเทียมกันของบริการจากคลังจัดเก็บส่วนต่อประสานที่ได้รับการเพิ่มขยายมาใช้ประกอบการคัดเลือกข้อเสนอบริการ (Service Offer) ที่จะถูกส่งกลับไปยังโปรแกรมผู้รับบริการเมื่อบริการเทรดเดอร์ไม่พบข้อเสนอบริการใดเลยภายในฐานข้อมูลที่สอดคล้องตามเงื่อนไขการค้นหา (Constraint) ที่ผู้รับบริการระบุมาในเมทอด query

บริการเทรดเดอร์ที่ผ่านการแก้ไขโปรแกรมแล้วจะทำงานตามกลไกในซีควนซ์ไดอะแกรมในรูปที่ 4.1 การทำงานตามซีควนซ์ไดอะแกรมนี้จะเริ่มจากการที่อิมพอร์ตเตอร์หรืออ็อบเจกต์หนึ่งคือโปรแกรมผู้รับบริการทำการค้นหาอินสแตนซ์ของบริการใดๆที่ต้องการใช้งานผ่านทางบริการเทรดเดอร์โดยใช้การเรียกไปยังเมทอด query¹¹ ของบริการเทรดเดอร์ตามรูปแบบที่กำหนดไว้ในข้อกำหนดการใช้งานบริการเทรดเดอร์ของโอเอ็มจี (Common Object Service (COS) Specification – Trading Specification) [10] (รูปที่ 4.2)

¹¹ ข้อกำหนดของโอเอ็มจีกำหนดให้ผู้รับบริการทำการอิมพอร์ตข้อเสนอบริการ (Import Service Offer) โดยเรียกใช้เมทอด query ในส่วนต่อประสาน Lookup ของบริการเทรดเดอร์



รูปที่ 4.1 ซีควอนซ์ไดอะแกรมการทำงานของบริการเทรดเดอร์ที่ผ่านการแก้ไขโปรแกรมแล้ว

```

typedef string Istring;
typedef Istring ServiceTypeName, Constraint, Preference;
enum HowManyProps { none, some, all };
union SpecifiedProps switch ( HowManyProps ) {
    case some: PropertyNameSeq prop_names;
};
void query(
    in ServiceTypeName type, in Constraint constr, in Preference pref,
    in PolicySeq policies, in SpecifiedProps desired_props, in unsigned long how_many,
    out OfferSeq offers, out OfferIterator offer_itr, out PolicyNameSeq limits_applied
) raises (
    IllegalServiceType, UnknownServiceType, IllegalConstraint, IllegalPreference,
    IllegalPolicyName, PolicyTypeMismatch, InvalidPolicyValue, IllegalPropertyName,
    DuplicatePropertyName, DuplicatePolicyName
);
  
```

รูปที่ 4.2 เมทอด query ของบริการเทรดเดอร์ตามข้อกำหนดของไอเอ็มจี

เมื่ออิมพอร์ตเตอร์เรียกใช้เมทอด query เพื่อร้องขอข้อเสนอบริการใดๆที่ต้องการใช้งาน เทรตเตอร์ จะทำการตรวจสอบชนิดของบริการ (Service Type Name) ที่อิมพอร์ตเตอร์ระบุในเมทอด query กับ คลังชนิดของบริการ (Type Repository) ก่อนที่จะทำการค้นหาข้อเสนอบริการที่มีเอ็กซ์พอร์ตเตอร์ (Exporter) มาลงทะเบียน (Register) ประกาศการให้บริการไว้ในฐานข้อมูลของเทรตเตอร์ ถ้าอิมพอร์ตเตอร์ระบุชนิดของบริการที่ไม่ถูกต้องหรือไม่มีข้อมูลอยู่ในคลังชนิดของบริการ บริการเทรตเตอร์จะยุติการค้นหาข้อเสนอบริการและทำการส่งเอ็กซ์เซพชันชนิด IllegalServiceType หรือ UnknownServiceType ไปยังออร์บของอิมพอร์ตเตอร์ อย่างไรก็ตามถ้าเทรตเตอร์พบชนิดของบริการที่อิมพอร์ตเตอร์ระบุในคลังชนิดของบริการ เทรตเตอร์จะนำข้อมูลชนิดของบริการนั้นไปใช้ในการตรวจสอบความถูกต้องของเงื่อนไขการค้นหา (Constraint) ตลอดจนลำดับความต้องการ (Preference) ที่อิมพอร์ตเตอร์ระบุในเมทอด query ต่อไป [10]

การค้นหาข้อเสนอบริการของเทรตเตอร์จะอยู่ในรูปของการพิจารณาชนิดตลอดจนคุณสมบัติของ ข้อเสนอบริการใดๆภายในฐานข้อมูลให้สอดคล้องกับเงื่อนไขการค้นหาที่ระบุโดยใช้ภาษาอธิบายเงื่อนไข (Constraint Language) อาทิเช่นถ้าผู้รับบริการต้องการค้นหาข้อเสนอบริการใดๆที่มีชนิดของบริการ เป็น ::service_by_id::account_by_id (สมมติให้บริการนี้ทำหน้าที่ค้นหายอดเงินคงเหลือในบัญชีจาก เลขที่บัญชี) แต่เนื่องจากอาจจะมีบริการที่มีชนิดดังกล่าวนี้มากกว่า 1 บริการ ดังนั้นผู้รับบริการจึงทำการ กำหนดเงื่อนไขการค้นหาเพื่อเลือกข้อเสนอบริการที่มีคุณสมบัติเฉพาะดังต่อไปนี้

- เป็นบริการที่ใช้เลขที่บัญชีเป็นคีย์ในการค้นหาข้อมูลจากฐานข้อมูลของธนาคาร
- เป็นบริการที่มีพื้นที่ให้บริการครอบคลุมพื้นที่ภาคกลางและภาคใต้ของประเทศ
- เป็นบริการที่คิดค่าใช้จ่ายในการค้นหาไม่เกิน 10 บาท/ครั้ง
- เป็นบริการที่ใช้ระยะเวลาในการประมวลผลไม่เกิน 100 ms/ครั้ง

ผู้รับบริการต้องการให้บริการเทรตเตอร์ส่งกลับข้อเสนอบริการใดๆที่พบเรียงตามลำดับค่าใช้จ่ายในการใช้บริการจากน้อยไปหามาก ดังนั้นจึงต้องระบุพารามิเตอร์ในเมทอด query ดังนี้

ค่าชนิดของบริการ (Service Type Name) เป็น ::service_by_id::account_by_id

ค่าเงื่อนไขการค้นหา (Constraint) ตามภาษาอธิบายเงื่อนไขเป็น (search_by == 'id')and ('C' ~ area)and('S' ~ area)and(cost <= 10)and(process_time <=100)

ค่าลำดับความต้องการ (Preference) เป็น min(cost) เป็นต้น

บริการเทรตเตอร์จะทำการค้นหาและส่งกลับกลุ่มของข้อเสนอบริการที่พบกลับไปยังอิมพอร์ตเตอร์ผ่านทางค่าพารามิเตอร์เอาต์พุตชนิด OfferSeq และ OfferIterator ของเมทอด query ตามรูปที่ 4.2 ชนิดของค่าพารามิเตอร์แบบ OfferSeq และ OfferIterator ถูกประกาศค่าไว้ในส่วนต่อประสาน CosTrading ตามข้อกำหนดของไอเอ็มจีดังนี้

```

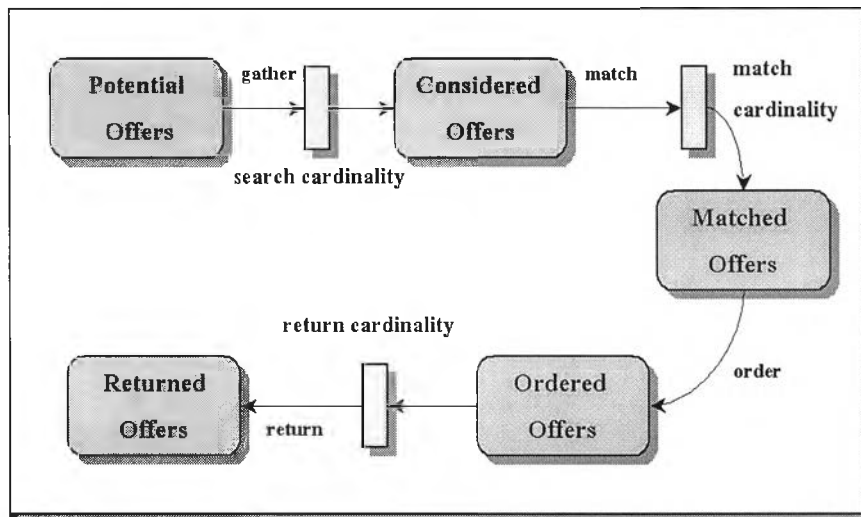
typedef Istring PropertyName;
typedef sequence<PropertyName> PropertyNameSeq;
typedef any PropertyValue;
struct Property{
    PropertyName name; PropertyValue value;
};
typedef sequence<Property> PropertySeq;
struct Offer {
    Object reference;
    PropertySeq properties;
};
typedef sequence<Offer> OfferSeq;
interface OfferIterator {
    unsigned long max_left () raises (UnknownMaxLeft);
    boolean next_n (in unsigned long n,out OfferSeq offers);
    void destroy ();
};

```

รูปที่ 4.3 การประกาศชนิด OfferSeq และ OfferIterator ในบริการเทรดเดอร์

จากรูปที่ 4.3 จะแสดงถึงรายละเอียดในการประกาศชนิด OfferSeq ซึ่งเป็นข้อมูลกลุ่ม (Sequence) ของ Offer โดยกำหนดให้ Offer แต่ละเอลิเมนต์ประกอบไปด้วยข้อมูลอ้างอิงของบริการ (Object Reference) ซึ่งจะมีชนิดข้อมูลเป็น Object ตลอดจนคุณสมบัติของข้อเสนอบริการนั้น (Properties) ซึ่งมีชนิดของข้อมูลเป็น PropertySeq บริการเทรดเดอร์จะส่งกลับข้อเสนอบริการที่พบผ่านทางพารามิเตอร์ OfferSeq ตามจำนวนที่อิมพอร์ตเตอร์ระบุมาในพารามิเตอร์ how_many ของเมทอด query ส่วนข้อเสนอบริการอื่นๆที่พบนอกเหนือจากที่ส่งกลับมาแล้วใน OfferSeq จะถูกส่งให้โปรแกรมของผู้รับบริการผ่านทางพารามิเตอร์เอาต์พุต OfferIterator เพื่อรอให้โปรแกรมของผู้รับบริการพิจารณาเรียกใช้งานต่อไป จำนวนข้อเสนอบริการทั้งหมดที่จะถูกส่งกลับไปยังอิมพอร์ตเตอร์จะมีจำนวนไม่เกินค่า return_card ของนโยบายการอิมพอร์ต (Import Policy)

ขั้นตอนตลอดจนสถานะของข้อเสนอบริการในกระบวนการค้นหาของเมทอด query ในบริการเทรดเดอร์จะเป็นไปตามรูปที่ 4.4 [10]



รูปที่ 4.4 สถานะของข้อเสนอบริการในกระบวนการอิมพอร์ตของบริการเทรดเดอร์

การคัดเลือกข้อเสนอบริการเพื่อส่งกลับไปยังโปรแกรมผู้รับบริการจะเริ่มจากการคัดเลือกกลุ่มของข้อเสนอบริการที่จะถูกตรวจสอบคุณสมบัติจากข้อเสนอบริการทั้งหมดภายในฐานข้อมูลของเทรดเดอร์ โดยกำหนดให้มีจำนวนไม่เกินค่า `search_card` ของนโยบายการอิมพอร์ต (ตารางที่ 4.1) ข้อเสนอบริการในขั้นตอนนี้จะเรียกว่า `Considered Offer` จะถูกส่งไปคัดเลือกเฉพาะข้อเสนอบริการที่สอดคล้องกับเงื่อนไขการค้นหาที่ระบุมาในเมทอด `query` เท่านั้น ข้อเสนอบริการที่ผ่านการคัดเลือกคุณสมบัติแล้วจะถูกเรียกว่า `Matched Offer` โดยบริการเทรดเดอร์จะกำหนดให้ `Matched Offer` มีจำนวนไม่เกินค่า `match_card` ของนโยบายการอิมพอร์ต (ตารางที่ 4.1) และก่อนที่บริการเทรดเดอร์จะส่งกลับข้อเสนอบริการที่ผ่านการคัดเลือกคุณสมบัติแล้วกลับไปยังอิมพอร์ตเตอร์ บริการเทรดเดอร์จะนำ `Matched Offer` ที่พบไปจัดเรียงลำดับตามข้อมูลลำดับความต้องการ (ใช้รูปแบบตามตารางที่ 4.2) ซึ่งข้อเสนอบริการที่ผ่านกระบวนการจัดลำดับแล้วจะถูกเรียกว่า `Ordered Offer` ในขั้นตอนนี้สุดท้ายของกระบวนการอิมพอร์ตบริการเทรดเดอร์จะส่ง `Ordered Offer` ที่ได้นั้นกลับไปยังอิมพอร์ตเตอร์เป็นจำนวนที่ไม่เกินกว่าค่าที่ระบุไว้ใน `return_card` ของนโยบายการอิมพอร์ต (ตารางที่ 4.1) ข้อเสนอบริการที่ถูกส่งกลับไปยังอิมพอร์ตเตอร์นี้จะถูกเรียกว่า `Returned Offer`

ตารางที่ 4.1 ค่าขอบเขตของนโยบายการอิมพอร์ตในบริการเทรดเดอร์ [10]

ค่าขอบเขต	นโยบาย	ชนิดข้อมูล ไอดีแอล	คำอธิบาย
Search_card	อิมพอร์ต	unsigned long	ค่าขอบเขตสูงสุดของจำนวนข้อเสนอบริการในฐานข้อมูลที่จะถูกค้นหา
Match_card	อิมพอร์ต	unsigned long	ค่าขอบเขตสูงสุดของจำนวนข้อเสนอบริการที่จะถูกเรียงลำดับ
Return_card	อิมพอร์ต	unsigned long	ค่าขอบเขตสูงสุดของจำนวนข้อเสนอบริการที่จะถูกส่งกลับไปยังอิมพอร์ตเตอร์

ตารางที่ 4.2 ลำดับความต้องการในบริการเทรดเดอร์ [10]

ค่าลำดับความต้องการ	คำอธิบาย
max <expression>	Matched Offer จะถูกส่งกลับไปยังอิมพอร์ตเตอร์โดยใช้การเรียงลำดับตาม <expression> จากมากไปน้อย
min <expression>	Matched Offer จะถูกส่งกลับไปยังอิมพอร์ตเตอร์โดยใช้การเรียงลำดับตาม <expression> จากน้อยไปมาก
with <expression>	Matched Offer จะถูกส่งกลับไปยังอิมพอร์ตเตอร์โดยใช้การเรียงลำดับตาม <expression> ที่เป็นจริงไว้ก่อนหน้าค่าที่เป็นเท็จ
Random	Matched Offer จะถูกส่งกลับไปยังอิมพอร์ตเตอร์โดยใช้การสุ่มเลือก
first ¹²	Matched Offer จะถูกส่งกลับไปยังอิมพอร์ตเตอร์เรียงตามลำดับการค้นหาพบจากฐานข้อมูล

ถ้าบริการเทรดเดอร์ไม่พบข้อเสนอบริการใดเลยในฐานข้อมูลที่สอดคล้องกับเงื่อนไขการค้นหาในเมทอด query บริการเทรดเดอร์โดยทั่วไปที่ไม่ผ่านการแก้ไขโปรแกรมจะส่งกลับ OfferSeq ที่มีจำนวนเอलिเมนต์เป็น 0 ซึ่งจะส่งผลให้โปรแกรมของผู้รับบริการไม่สามารถเรียกใช้งานบริการใดๆได้เลย [6,10] งานวิจัยนี้ได้ทำการแก้ไขโปรแกรมของบริการเทรดเดอร์ให้เรียกไปยังเมทอด search_equ_trader ของบริการค้นหาเพื่อให้บริการค้นหาทำการค้นหาและส่งกลับข้อเสนอบริการใดๆที่สามารถทำงานได้เท่าเทียมกันกับข้อเสนอบริการที่มีชนิดของบริการตามที่คุณใช้งานต้องการ (รายละเอียดของเมทอด search_equ_trader

¹² บริการเทรดเดอร์จะใช้ค่า first เป็นลำดับความต้องการโดยปริยาย (Default Preference) ถ้าหากว่าอิมพอร์ตเตอร์ไม่ได้ทำการระบุค่าใดๆมาในเมทอด query

อยู่ในหัวข้อที่ 3.2) บริการเทรดเดอร์ใหม่ที่ผ่านการแก้ไขโปรแกรมแล้ว จะทำการคัดเลือกข้อมูลที่ได้รับจากบริการค้นหาให้เหลือเฉพาะข้อเสนอบริการที่มีคุณสมบัติสอดคล้องกับเงื่อนไขการค้นหาเพื่อส่งกลับไปทำงานยังอิมพอร์ตเตอร์ต่อไป งานวิจัยนี้ใช้การกำหนดค่าข้อมูลอ้างอิงของตัวดำเนินการแปลงลงในส่วน reference ของข้อเสนอบริการที่จะถูกส่งกลับไปยังอิมพอร์ตเตอร์ การเรียกใช้งานบริการแทนที่ของโปรแกรมผู้รับบริการจึงมีรูปแบบที่เป็นไปตามซีควนซ์ไดอะแกรมในรูปที่ 4.1 กล่าวคือโปรแกรมผู้รับบริการจะต้องทำการเรียกใช้งานบริการแทนที่ผ่านทางตัวดำเนินการแปลงโดยใช้การสกัดข้อมูลอ้างอิงของตัวดำเนินการแปลงจากส่วน reference ของข้อเสนอบริการที่ได้รับมาจากบริการเทรดเดอร์ก่อน

การแก้ไขบริการเทรดเดอร์ให้ทำงานในลักษณะดังกล่าวนี้จะช่วยเพิ่มโอกาสในการค้นพบข้อเสนอบริการที่มีคุณสมบัติตรงตามความต้องการของผู้รับบริการมากยิ่งขึ้น ภายหลังจากที่ผู้รับบริการได้รับข้อเสนอบริการที่เท่าเทียมกันจากบริการเทรดเดอร์แล้ว โปรแกรมผู้รับบริการจะสามารถเรียกใช้งานตัวดำเนินการแปลงได้ตามตัวอย่างโปรแกรมดังต่อไปนี้

```
(1) org.omg.CosTrading.Policy[] policies = new org.omg.CosTrading.Policy[0];
(2) org.omg.CosTrading.LookupPackage.SpecifiedProps desiredprops = new
    org.omg.CosTrading.LookupPackage.SpecifiedProps();
(3) desiredprops._default();
(4) org.omg.CosTrading.OfferSeqHolder offers = new org.omg.CosTrading.OfferSeqHolder();
(5) org.omg.CosTrading.OfferIteratorHolder iter = new org.omg.CosTrading.OfferIteratorHolder();
(6) org.omg.CosTrading.PolicyNameSeqHolder limits = new
    org.omg.CosTrading.PolicyNameSeqHolder();
(7) String servicetypeName = "::service_by_id::account_by_id";
(8) String constraint = "(searchby == 'id')and('C' ~ area)and('S' ~ area)and(cost <= 10)
    and(responsetime < 100)";
(9) String pref="min(cost)";
(10) int howmany =5;
(11) trader.query(servicetypeName,constraint,pref,policies,desiredprops,howmany,offers,iter,limits);
(12) org.omg.CosTrading.Offer[] offercon=offers.value;
(13) if((offercon==null)||((offercon.length==0)){
(14)     System.out.println("No Matched Service Offer with this constraint");
(15)     return;
(16) }
```

รูปที่ 4.5 ตัวอย่างการอิมพอร์ตข้อเสนอบริการจากบริการเทรดเดอร์

```

(17) org.omg.CORBA.Object obj = offercon[0].reference;
(18) org.omg.CosTrading.Property[] p = offercon[0].properties;
(19) if(obj!=null){
(20)     service_by_id.account_by_id ser1 = service_by_id.account_by_idHelper.narrow(obj);
(21)     String ret=String.valueOf(ser1.getbalance(1));
(22)     System.out.println("Invoke Service -> ID : 1 have saving balance = "+ret);
(23) }

```

รูปที่ 4.5 ตัวอย่างการอิมพอร์ตข้อเสนอบริการจากบริการเทรดเดอร์ (ต่อ)

บรรทัดที่ 1-10 เป็นคำสั่งส่วนการเตรียมค่าพารามิเตอร์ต่างๆที่อิมพอร์ตเตอร์จะส่งผ่านเมทอด query ไปยังบริการเทรดเดอร์

บรรทัดที่ 11 เป็นคำสั่งส่วนการเรียกใช้เมทอด query ในส่วนต่อประสาน Lookup ของบริการเทรดเดอร์

บรรทัดที่ 12-16 เป็นคำสั่งส่วนการพิจารณาข้อเสนอบริการที่ได้รับจากบริการเทรดเดอร์ทางพารามิเตอร์ในเมทอด query ที่มีชนิดของข้อมูลเป็น org.omg.CosTrading.Offer[]

บรรทัดที่ 17-18 เป็นคำสั่งส่วนการอ่านข้อมูลอ้างอิงของบริการ (Object Reference) และคุณสมบัติของบริการจากข้อเสนอบริการที่ได้รับจากบริการเทรดเดอร์เป็นเอลิเมนต์แรก

บรรทัดที่ 19-23 เป็นคำสั่งส่วนการเรียกใช้บริการโดยโปรแกรมของผู้รับบริการผ่านทางข้อมูลอ้างอิงของบริการที่ได้รับจากบริการเทรดเดอร์

4.2 การออกแบบและแก้ไขบริการเทรดเดอร์

งานวิจัยนี้ใช้โปรแกรมออร์บาคัสเทรดเดอร์รุ่น 3.3.1 สำหรับภาษาจาวา (ORBacus Trader For Java – version 3.3.1¹³) ซึ่งเป็นโปรแกรมประเภทฟรีแวร์ (Freeware) ที่สามารถดาวน์โหลด (Download) โปรแกรมต้นฉบับ (Source Code) มาทำการแก้ไขปรับปรุงการทำงานในเมทอด query ของเทรดเดอร์ได้ เพื่อให้ทำงานตามรายละเอียดที่กล่าวถึงไปแล้วในหัวข้อที่ 4.1

จากการศึกษาพบว่าโปรแกรมออร์บาคัสเทรดเดอร์ใช้หลักการโปรแกรมเชิงวัตถุในการพัฒนาคลาสต่างๆด้วยภาษาจาวาภายในแพ็คเกจ com.ooc.CosTrading เพื่อทำหน้าที่ให้บริการงานด้านต่างๆกับผู้รับบริการตามส่วนต่อประสานที่กำหนดไว้ในข้อกำหนดเทรดเดอร์ของโอเอ็มจี [13] อาทิเช่นคลาส

¹³ ซอฟต์แวร์ออร์บาคัสเทรดเดอร์สามารถดาวน์โหลดได้จาก <http://www.orbacus.com/trader/download.html>

RegisterImpl ถูกใช้สำหรับทำหน้าที่ให้บริการตามส่วนต่อประสาน Register หรือคลาส LookupImpl สำหรับทำหน้าที่ให้บริการตามส่วนต่อประสาน Lookup ของเทรดเดอร์เป็นต้น

งานวิจัยนี้ใช้การแก้ไขโปรแกรมออร์บาคัสเทรดเดอร์ภายในเพิ่มข้อมูลต่างๆดังรายละเอียดต่อไปนี้

4.2.1 เพิ่มข้อมูล Server.java

เป็นเพิ่มข้อมูลที่ใช้จัดเก็บคลาส Server ซึ่งถูกใช้งานเป็นคลาสหลักของโปรแกรมออร์บาคัสเทรดเดอร์ คลาส Server มีหน้าที่ในการรับค่าตัวเลือกต่างๆจากผู้ใช้งานเพื่อใช้ในการกำหนดรูปแบบการทำงานของบริการเทรดเดอร์ อาทิเช่นผู้ใช้งานสามารถเรียกใช้ตัวเลือก -i เพื่อกำหนดให้บริการเทรดเดอร์แสดงข้อมูลอ้างอิงของบริการบนหน้าจอตลอดจนทำการจัดเก็บข้อมูลนี้ลงในเพิ่มข้อมูลชื่อ ob_ts เป็นต้น

นอกจากนั้นแล้วคลาส Server ยังถูกใช้ในการสร้างและกำหนดค่าเริ่มต้นให้กับอินสแตนซ์ของออร์บ ออบเจ็คต์อะแดปเตอร์ ฐานข้อมูลท้องถิ่น (Local Database) ตลอดจนบริการย่อยต่างๆ (Trader Components) ตามส่วนต่อประสานที่ระบุไว้ในข้อกำหนดของบริการเทรดเดอร์ อาทิเช่นบริการ Lookup, Register หรือ Admin เป็นต้น งานวิจัยนี้ได้ทำการเพิ่มเติมโปรแกรมออร์บาคัสเทรดเดอร์ให้มีตัวเลือก -eq เพื่อให้ผู้ใช้งานโปรแกรมเทรดเดอร์สามารถกำหนดรูปแบบการค้นหาข้อเสนอบริการในเมทอด query ว่าจะต้องการให้เทรดเดอร์ทำการค้นหาข้อมูลความสัมพันธ์แบบเท่าเทียมกันของบริการจากคลังจัดเก็บส่วนต่อประสานที่ได้รับการขยาย เมื่อเทรดเดอร์ไม่พบข้อเสนอบริการในฐานข้อมูลที่สอดคล้องกับเงื่อนไขที่ผู้ใช้งานระบุมาหรือไม่

4.2.2 เพิ่มข้อมูล LookupImpl.java

เป็นเพิ่มข้อมูลที่ใช้จัดเก็บคลาส LookupImpl ซึ่งถูกใช้งานเป็นบริการย่อย Lookup ของโปรแกรมออร์บาคัสเทรดเดอร์ คลาส LookupImpl จะสืบทอดคุณสมบัติต่างๆจากคลาสไลบรารี org.omg.CosTrading._LookupImplBase โดยมีหน้าที่ในการตรวจสอบนโยบายการค้นหาข้อเสนอบริการตลอดจนทำการสร้างและเรียกใช้วัตถุ Query เพื่อใช้งานในกระบวนการอิมพอร์ตข้อเสนอบริการ

งานวิจัยนี้ได้ทำการเพิ่มเติมคุณสมบัติ useeq และเมทอด setuseeq เพื่อใช้ผ่านค่าคุณสมบัติการค้นหาบริการแทนที่มีชนิดตัวแปรแบบ boolean จากคลาส Server ไปยังคลาส Query

4.2.3 เพิ่มข้อมูล Query.java

เป็นเพิ่มข้อมูลที่ใช้จัดเก็บคลาส Query ซึ่งถูกใช้งานเป็นคลาสหลักในการทำกระบวนการอิมพอร์ตข้อเสนอบริการไปยังโปรแกรมของผู้รับบริการของโปรแกรมออร์บาคัสเทรดเดอร์

งานวิจัยนี้ได้ทำการเพิ่มเติมคุณสมบัติ useeq และเมทอดต่างๆได้แก่เมทอด setuseeq และ search_eq_service รวมทั้งได้ทำการแก้ไขเมทอด query ของคลาส Query กลไกการทำงานที่ถูกเพิ่มเติมในบริการเทรดเดอร์ใหม่จะเป็นไปตามแอ็คทิวิตีไดอะแกรมในรูปที่ 4.5

การทำงานจะเริ่มจากการตรวจสอบผลการค้นหาข้อเสนอบริการในฐานข้อมูลของเทรดเดอร์ตามเงื่อนไขการค้นหาที่อิมพอร์ตเตอร์ระบุมาในเมทอด query โดยใช้คำสั่งการตรวจสอบดังนี้

`(matchingOffers_.size()==0)&&(useeq==true)`

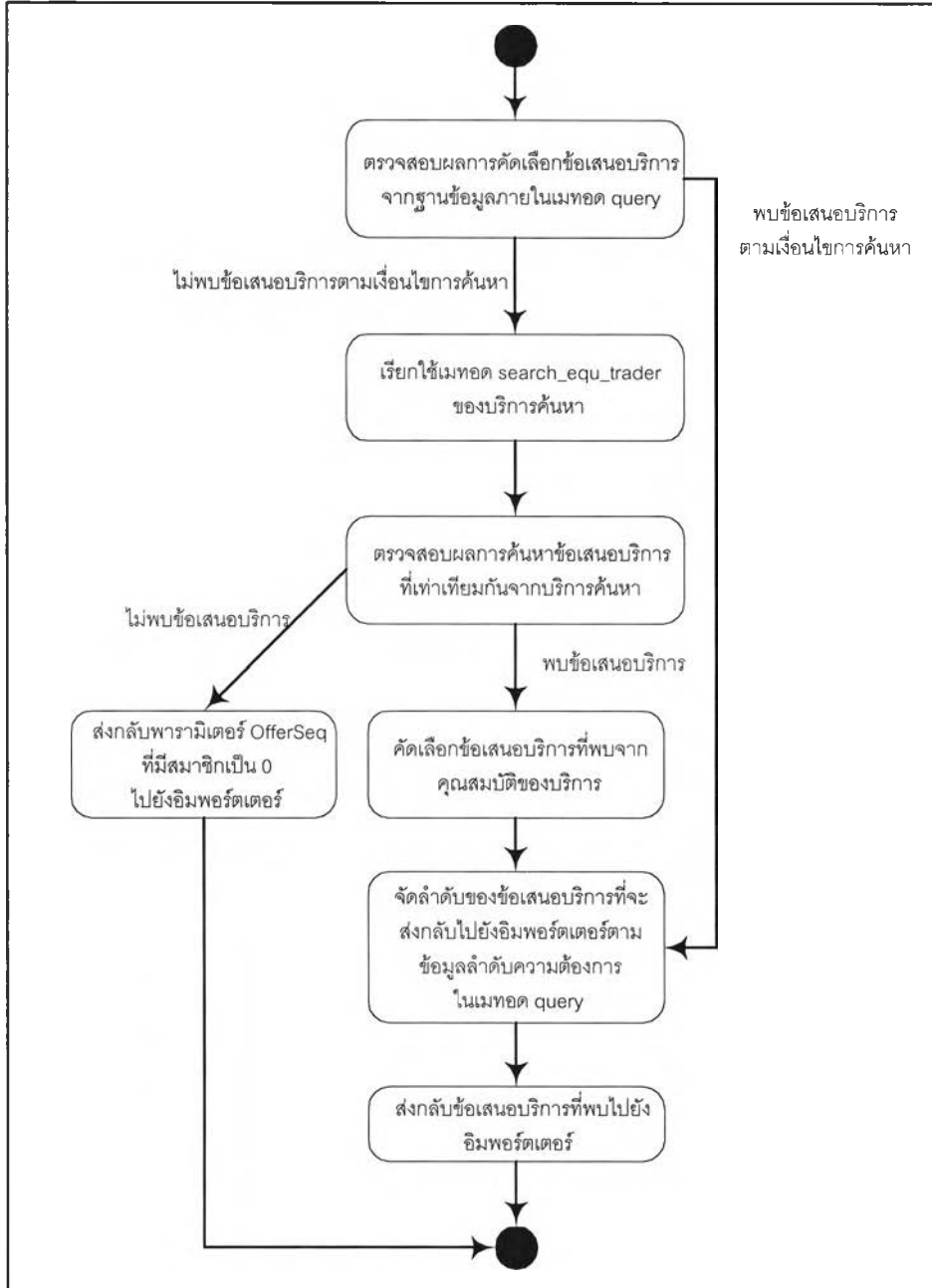
เมทอด query จะทำการเรียกเมทอด search_eq_service ซึ่งจะทำงานตามขั้นตอนต่างๆดังนี้

- สร้างการเชื่อมต่อไปยังบริการค้นหาเพื่อเรียกใช้เมทอด search_eq_trader ซึ่งจะส่งกลับกลุ่มของข้อเสนอบริการที่สามารถทำงานได้เท่าเทียม โดยจะมีชนิดของข้อมูลเป็น org.omg.CosTrading.Offer[] กลับมาที่บริการเทรดเดอร์

- คัดเลือกข้อเสนอบริการที่ได้รับจากบริการค้นหาตามจำนวนที่ระบุในค่า search_card ของนโยบายการอิมพอร์ต โดยเรียกใช้เมทอด ConsiderOffer ผลที่ได้รับจากการคัดเลือกจะถูกจัดเก็บลงในตัวแปร potentialEq เพื่อถูกใช้เป็นข้อเสนอบริการที่จะถูกคัดเลือกตามค่าคุณสมบัติของบริการต่อไป (Considered Service Offer)

- คัดเลือกข้อเสนอบริการตามคุณสมบัติของบริการนั้นให้สอดคล้องกับเงื่อนไขการค้นหา โดยใช้การสร้างอินสแตนซ์ของคลาส OfferEvaluator และเมทอด query จะทำการเรียกใช้เมทอด getResult ของคลาส OfferEvaluator เพื่อทำการจัดเก็บข้อเสนอบริการที่ตรงกับเงื่อนไขลงในตัวแปร matchingOffers_ พร้อมทั้งจัดเก็บจำนวนของข้อเสนอบริการที่ผ่านการคัดเลือกลงในตัวแปร localMatchCount_

ในลำดับสุดท้ายคลาส Query ของบริการเทรดเดอร์จะทำการจัดลำดับข้อเสนอบริการที่จะส่งกลับไปยังอิมพอร์ตเตอร์ตามข้อมูลลำดับความต้องการ (Preference) ที่ผู้รับบริการระบุมาในเมทอด query การจัดลำดับข้อเสนอบริการนี้จะกระทำผ่านการเรียกใช้เมทอด sort ของคลาส Preference



รูปที่ 4.6 แอ็คทิวิตี้ไดอะแกรมการทำงานของโปรแกรมที่เพิ่มลงในบริการเทรดเดอร์

4.3 การติดตั้งบริการเทรดเดอร์ที่ผ่านการแก้ไขแล้ว

ผู้ใช้งานเทรดเดอร์จะสามารถติดตั้งโปรแกรมออร์บาคัสเทรดเดอร์ที่ผ่านการแก้ไขโปรแกรมแล้วโดยเรียกใช้คำสั่งดังต่อไปนี้

```
java com.ooc.CosTrading.Server -i -OAport <port No.> -eq
```

รูปที่ 4.7 คำสั่งการเรียกใช้บริการเทรดเดอร์ในงานวิจัยนี้

ตัวเลือก `-i` จะใช้กำหนดให้โปรแกรมออร์บาคัสเทรดเดอร์ทำการแสดงและจัดเก็บข้อมูลอ้างอิงของบริการเทรดเดอร์ลงในแฟ้มข้อมูลชื่อ `ob_ts`

ตัวเลือก `-OAport <port No.>` จะใช้ในการกำหนดหมายเลขของพอร์ตที่ออกเจ็ทอะแดปเตอร์จะได้รับการร้องขอ (Request) จากโปรแกรมผู้รับบริการบนโพรโตคอลที่ซีพีไอพี (TCP/IP)

ตัวเลือก `-eq` จะใช้กำหนดให้โปรแกรมออร์บาคัสเทรดเดอร์ทำการค้นหาข้อเสนอบริการที่ทำงานได้เท่าเทียมจากข้อมูลความสัมพันธ์แบบเท่าเทียมกันของบริการ ซึ่งเทรดเดอร์จะเรียกไปยังเมทอด `search_equ_trader` ของบริการค้นหา

ภายหลังจากที่ผู้ใช้งานเรียกคำสั่งตามรูปที่ 4.6 บริการเทรดเดอร์จะแสดงผลการทำงานดังนี้

```
<<Running Trading Service>>
```

```
<Option> Force trader to find equivalent service(s)
```

```
Trader's Object Ref :-
```

```
IOR:0000000000000002249444c3a6f6d672e6f72672f436f7354726164696e672f4c6f6f6b75703a312e300000000000001000000000000003700010000000000018636f7262617372762e6165726f746861692e6f722e74680013880000000000f54726164696e675365727669636500
```

```
<Option> Write Trader's IOR to file ... /home/ior/ob_ts
```

รูปที่ 4.8 ผลการเรียกใช้บริการเทรดเดอร์ตามคำสั่งในรูปที่ 4.7