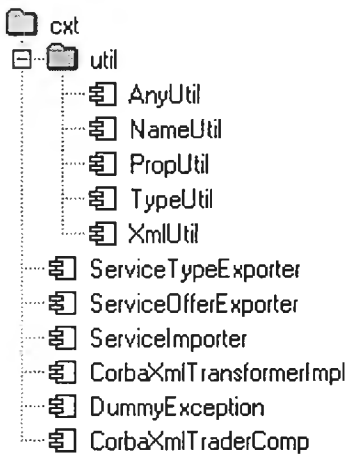
บทที่ 4
ต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยายการแปลงคำอธิบายระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล

ต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยายในวิทยานิพนธ์นี้จะเป็นการพัฒนาขึ้นภายใต้ข้อกำหนดของสถาปัตยกรรมคอร์บารุ่นที่ 2.2 โดยมีออร์บ (Object Request Broker (ORB)) และบริการเทรดเดอร์ของ JacORB รุ่นที่ 1.0 เบต้า 14 (Version 1.0 Beta 14) [14] คลังส่วนต่อประสานของ Visiborker สำหรับจาวารุ่น 3.4 [15] และดีโอเอ็มที่พัฒนาขึ้นโดย Sun Micorsystem สำหรับจาวา โดยภาษาโปรแกรมที่ใช้คือภาษาโปรแกรมจาวา (Java) รุ่น 1.1.7 ซึ่งสามารถนำไปใช้กับสภาวะแวดล้อมการทำงานของจาวา (Java Runtime Environment (JRE)) ตั้งแต่รุ่น 1.1.7 ขึ้นไปได้

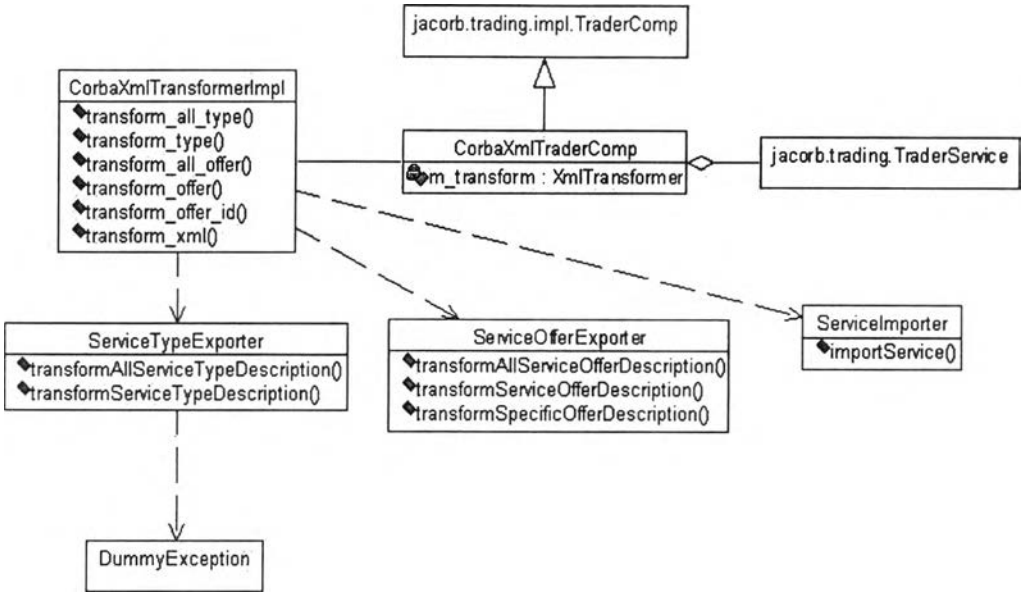
ในบทนี้จะแบ่งหัวข้อหลักออกเป็นสองส่วนคือ ส่วนการออกแบบและพัฒนาคลาสของต้นแบบเพื่อใช้ในการแปลงคำอธิบาย และส่วนการเพิ่มขยายเทรดเดอร์ของ JacORB เพื่อให้รองรับการแปลงคำอธิบาย โดยในหัวข้อแรกจะได้กล่าวถึงลักษณะโครงสร้างของแผนผังคลาส (Class Diagram) และความสัมพันธ์ของแต่ละคลาสในการแปลงคำอธิบาย ส่วนในหัวข้อที่สองจะได้นำเสนอการแก้ไขรหัสต้นฉบับ (Source Code) ของเทรดเดอร์ของ JacORB เพื่อให้รองรับการแปลงคำอธิบาย

4.1 การออกแบบและพัฒนาคลาสของต้นแบบเพื่อใช้ในการแปลงคำอธิบาย

คลาสในการแปลงคำอธิบายในวิทยานิพนธ์นี้ถูกแบ่งออกตามลักษณะหน้าที่ที่เกี่ยวเนื่องกัน และมีการจัดกลุ่มของคลาสที่ใช้ในการแปลงคำอธิบาย และส่วนเพิ่มขยายออกเป็นแพคเกจ (Package) ของจาวา โดยมีโครงสร้างแพคเกจดังรูปที่ 4.1 และแผนผังคลาสดังรูปที่ 4.2



รูปที่ 4.1 โครงสร้างแพคเกจของคลาสในการแปลงคำอธิบาย และเพิ่มขยายเทรดเดอร์



รูปที่ 4.2 แผนผังคลาสของแพคเกจในการแปลงคำอธิบาย และเพิ่มขยายเทรดเดอร์

จากรูปที่ 4.1 และ 4.2 สามารถนำมาอธิบายโดยละเอียดดังนี้คือ

แพคเกจ CXT - แพคเกจนี้เป็นแพคเกจหลักในการแปลงคำอธิบายระหว่างคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล ทั้งชนิดของบริการ และข้อเสนอบริการ รวมไปถึงคลาสหลักในการเพิ่มขยายเทรดเดอร์ของ JacORB ประกอบไปด้วยคลาสต่างๆ ดังนี้

- ServiceTypeExporter - สำหรับการแปลงคำอธิบายชนิดของบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล มีตัวกระทำการคือ
  - int transformAllServiceTypeDescription()
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายชนิดของบริการทั้งหมดที่มีอยู่ภายในคลังจัดเก็บชนิดของบริการโดยมีหลักการทำงานเช่นเดียวกับ transform_all_type ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.1.2 และ 3.1.3 โดยมีค่าที่ส่งกลับคือจำนวนชนิดของบริการที่ได้รับการแปลง
  - boolean transformServiceTypeDescription(String serviceTypeName)
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายชนิดของบริการตามชื่อชนิดของบริการที่กำหนดในพารามิเตอร์ โดยมีหลักการทำงานเช่นเดียวกับ transform_type ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.1.2 และ 3.1.3 โดยมีค่าที่ส่งกลับเป็น boolean เพื่อบ่งบอกว่าสามารถแปลงได้สำเร็จหรือไม่

- ServiceOfferExporter - สำหรับการแปลงคำอธิบายข้อเสนอบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล มีตัวกระทำการคือ
  - String[] transformAllServiceOfferDescription()
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายข้อเสนอบริการทั้งหมดที่มีอยู่ในคอร์บาเทรดเดอร์ โดยมีหลักการทำงานเช่นเดียวกับ transform_all_offer ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.2.2 โดยมีค่าที่ส่งกลับคือแถวลำดับ (Array) ของตัวระบุข้อเสนอบริการ
  - String[] transformServiceOfferDescription(String serviceTypeName)
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายข้อเสนอบริการทั้งหมดที่มีชนิดของบริการเป็นไปตามพารามิเตอร์ โดยมีหลักการทำงานเช่นเดียวกับ transform_offer ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.2.2 โดยมีค่าที่ส่งกลับคือแถวลำดับ (Array) ของตัวระบุข้อเสนอบริการ
  - boolean transformSpecificOfferDescription(String offerId)
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายข้อเสนอบริการตามตัวระบุข้อเสนอบริการที่กำหนดในพารามิเตอร์ โดยมีหลักการทำงานเช่นเดียวกับ transform_offer_id ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.2.2 โดยมีค่าที่ส่งกลับเป็น boolean เพื่อบ่งบอกว่าสามารถแปลงได้สำเร็จหรือไม่
- ServiceImporter - สำหรับการแปลงคำอธิบายบริการจากเอ็กซ์เอ็มแอลไปเป็นรูปแบบของคอร์บาเทรดเดอร์ มีตัวกระทำการคือ
  - String importService(String fileloc)
    การทำงานหลักของตัวกระทำการนี้คือการแปลงคำอธิบายบริการซึ่งมีที่อยู่ตามที่ระบุในพารามิเตอร์ fileloc โดยตัวกระทำการนี้จะทำการแยกแยะว่าเอกสารตามที่อยู่นี้เป็นคำอธิบายชนิดของบริการ หรือคำอธิบายข้อเสนอบริการเพื่อทำการแปลงเอกสารให้อยู่ในรูปของคอร์บาเทรดเดอร์ต่อไป ตัวกระทำการนี้มีหลักการทำงานเช่นเดียวกับ transform_xml ที่กล่าวมาแล้วในหัวข้อ 3.5 และแปลงตามกฎที่กำหนดในหัวข้อ 3.3 และ 3.4 แล้วแต่ชนิดของเอกสาร ค่าที่ส่งกลับจะเป็น String ที่เป็นตัวระบุข้อเสนอบริการในกรณีที่เอกสารเป็นคำอธิบายข้อเสนอบริการ หรือเป็นค่าพิเศษ "NA" เมื่อเอกสารเป็นคำอธิบายชนิดของบริการ

- CorbaXmlTransformerImpl - สำหรับการสร้างต้นแบบเพื่อให้เป็นบริการของคอร์บา โดยคลาสนี้เป็นคลาสการทำให้เกิดผล (Implementation Class) ของส่วนต่อประสาน CorbaXmlTransformer ตามไอดีแอลที่ได้กล่าวมาแล้วในหัวข้อ 3.5 ดังนั้นจึงมีตัวกระทำการหกตัวที่มีหน้าที่ตามที่กำหนดในไอดีแอล ซึ่งในแต่ละตัวกระทำการจะมีการเรียกใช้คลาส ServiceTypeExporter, ServiceOfferExporter หรือ ServiceImporter โดยจะกล่าวในหัวข้อ 4.1.1 ต่อไป
- DummyException - สำหรับข้อยกเว้นแบบดัมมีที่ใช้ในการเขียนโปรแกรมของคลาส ServiceTypeExporter
- CorbaXmlTraderComp - เป็นคลาสที่ทำการเพิ่มขยายส่วนโปรแกรมของ JacORB คือ คลาส jacorb.trading.impl.TraderComp ซึ่งเป็นจุดเชื่อมต่อไปยังคลาสอื่นๆ ที่เป็นบริการภายใน เทรดเดอร์ ซึ่งจะทำการเก็บตัวอ้างถึงวัตถุเช่น Lookup, Register, Link, Proxy และ Admin โดย CorbaXmlTraderComp นี้จะทำการสืบทอดคุณสมบัติมาจาก jacorb.trading.impl.TraderComp แต่จะมีลักษณะประจำเพิ่มขึ้นคือ CorbaXmlTransformer ดังนั้นจึงมีตัวกระทำการเพิ่มขึ้นสองตัวคือ
  - CorbaXmlTransformer getCorbaXmlTransformer()
    ตัวกระทำการนี้เป็นตัวกระทำการสำหรับการเข้าถึงลักษณะประจำ CorbaXmlTransformer ที่เรียกว่าแอคเซสเซอร์ (Accessor) หรือเก็ตเตอร์ (Getter)
  - void setCorbaXmlTransformer(CorbaXmlTransformer value)
    ตัวกระทำการนี้เป็นตัวกระทำการสำหรับกำหนดค่าของลักษณะประจำ XmlTransfomrer ที่เรียกว่ามิวเตเตอร์ (Mutator) หรือเซ็ตเตอร์ (Setter)

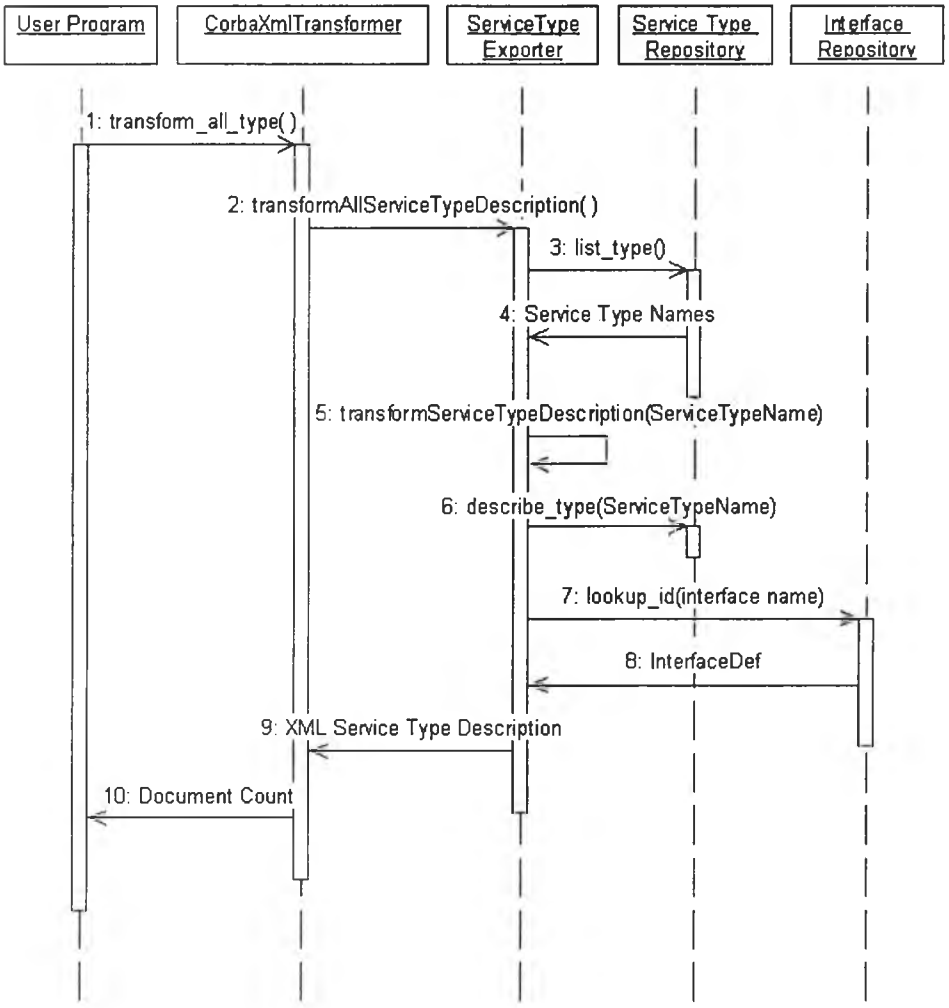
แพคเกจ Util - แพคเกจนี้เป็นแพคเกจที่รวบรวมคลาสที่คอยช่วยเหลือต้นแบบอันได้แก่

- AnyUtil - เป็นคลาสช่วยเหลือในการกำหนดค่า และนำค่าพื้นฐานจากตัวแปรแบบ any
- NameUtil - เป็นคลาสช่วยเหลือในการแปลงตัวระบุส่วนต่อประสานให้เป็นชื่อสัมบูรณ์ (Absolute Name) ชื่อโมดูล (Module Name) และช่วยในการแปลงชื่อแฟ้มข้อมูลเพื่อจัดเก็บเอกสารเอ็กซ์เอ็มแอล
- PropUtil - เป็นคลาสช่วยเหลือในการอ่านคุณสมบัติที่กำหนดในสภาวะแวดล้อมเริ่มต้นของ JacORB
- TypeUtil - เป็นคลาสช่วยเหลือในการแปลงระหว่างตัวอักษรในเอกสารเอ็กซ์เอ็มแอลกับ TypeCode, IDLType และตัวแปรพื้นฐานตามข้อกำหนดของคอร์บา
- XmlUtil - เป็นคลาสช่วยเหลือในการแปลงสภาวะ และคุณสมบัติต่างๆ ระหว่างคอร์บากับเอ็กซ์เอ็มแอล

4.1.1 ลำดับเหตุการณ์เมื่อมีการเรียกใช้การแปลงคำอธิบายบริการ

จากที่กล่าวมาแล้วข้างต้น และในบทที่ 3 ส่วนต่อประสานในไอดีแอลที่ออกแบบไว้ เป็นเสมือนส่วนการเรียกใช้บริการ CorbaXmlTransformer เพื่อให้แปลงคำอธิบายบริการ สำหรับกรรมวิธีในการแปลงภายในจริงๆ นั้นทำโดยตัวต้นแบบ CorbaXmlTransformerImpl อีกทีหนึ่งเพื่อเรียกใช้วัตถุที่ทำหน้าที่ตามที่ร้องขอมา ดังนั้นในหัวข้อนี้จะได้กล่าวถึงลำดับการเรียกใช้วัตถุของตัวต้นแบบเพื่อเรียกใช้การแปลงที่เตรียมไว้ให้ในไอดีแอลทั้งหกส่วนต่อประสานดังนี้

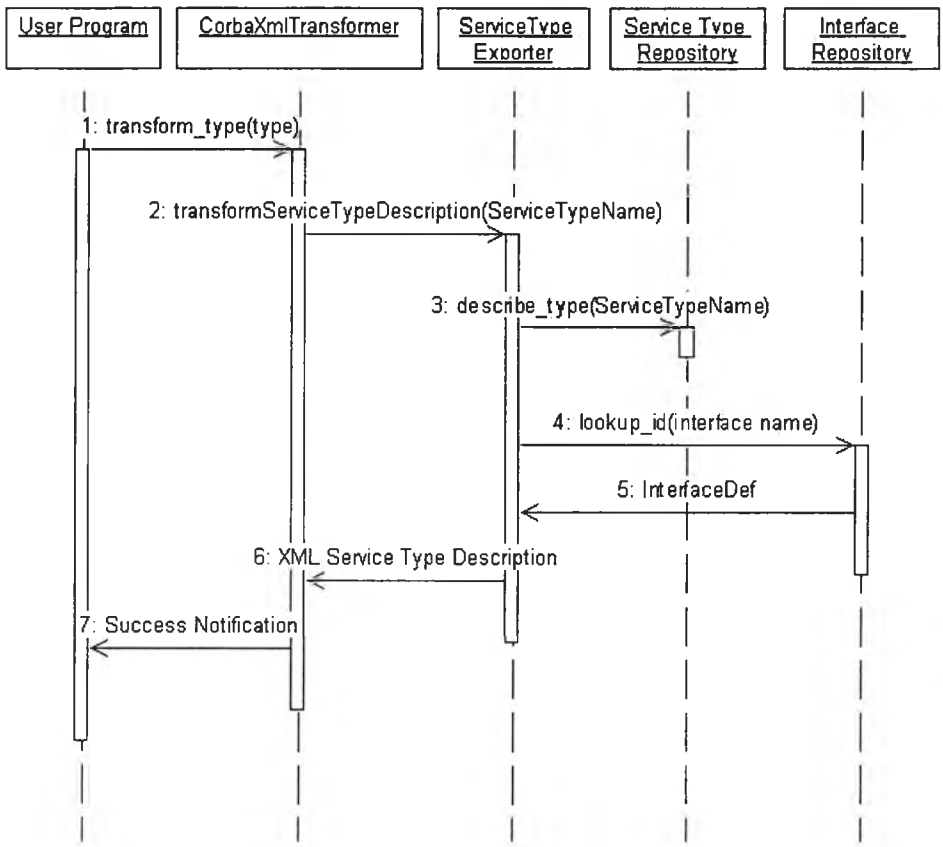
1. เมื่อมีการเรียกใช้ transform_all_type() เพื่อแปลงคำอธิบายชนิดของบริการทั้งหมดภายในเทรดเดอร์ให้เป็นเอ็กซ์เอ็มแอล



รูปที่ 4.3 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_all_type()

ลำดับเหตุการณ์ที่เกิดขึ้นคือเมื่อมีการเรียก CorbaXmlTransformer ผ่านบริการคอร์บา (CORBA Service) ส่วนคลาสการทำให้เกิดผลของส่วนต่อประสาน CorbaXmlTransformer อันได้แก่คลาส CorbaXmlTransformerImpl จะทำการสร้างวัตถุจากคลาส ServiceTypeExporter และเรียกใช้ transformAllServiceTypeDescription() เพื่อแปลงคำอธิบายชนิดของบริการ โดยวัตถุของ ServiceTypeExporter จะทำการเรียกดูชนิดของบริการทั้งหมดภายในคลังชนิดของบริการ จากนั้นเรียกใช้ transformServiceTypeDescription() สำหรับแต่ละชนิดของบริการ เพื่อทำการแปลงให้เป็นเอกสารเอ็กซ์เอ็มแอลต่อไป (ดูหัวข้อถัดไปในเรื่องการแปลงคำอธิบายของแต่ละชนิดของบริการ)

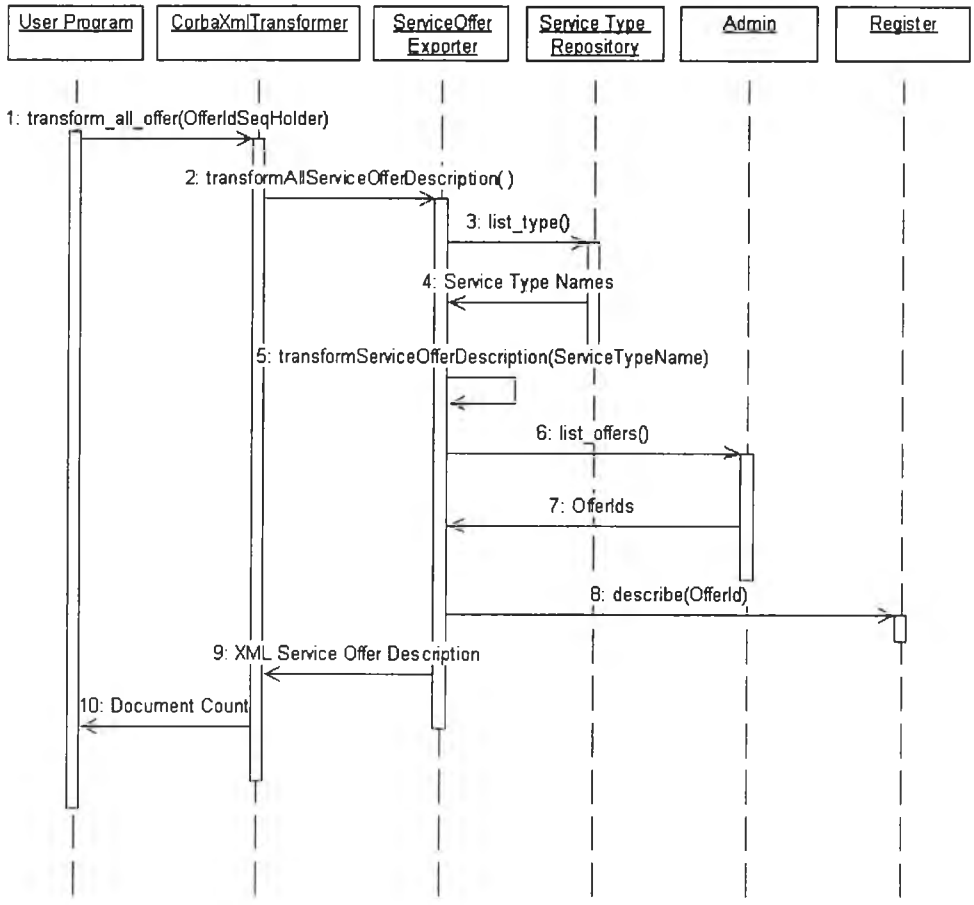
2. เมื่อมีการเรียกใช้ transform_type() เพื่อแปลงคำอธิบายชนิดของบริการที่กำหนดในพารามิเตอร์ ให้เป็นเอ็กซ์เอ็มแอล



รูปที่ 4.4 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_type()

ลำดับเหตุการณ์ที่เกิดขึ้นคือเมื่อมีการเรียก CorbaXmlTransformer ผ่านบริการคอร์บา วัตถุของคลาส ServiceTypeExporter จะถูกสร้างขึ้นเพื่อรองรับการเรียกใช้ transformServiceTypeDescription() เพื่อแปลงคำอธิบายชนิดของบริการที่กำหนด ตัวกระทำการนี้จะทำการร้องขอไปยังคลังชนิดของบริการเพื่อนำคำอธิบายชนิดของบริการนี้ไปสร้างวัตถุดีโอเอ็ม จากนั้นจึงนำตัวระบุส่วนต่อประสานไปค้นหาคำอธิบายส่วนต่อประสานจากคลังส่วนต่อประสานเพื่อทำการแปลงให้เป็นวัตถุดีโอเอ็มและสร้างเป็นเอกสารเอ็กซ์เอ็มแอลต่อไป

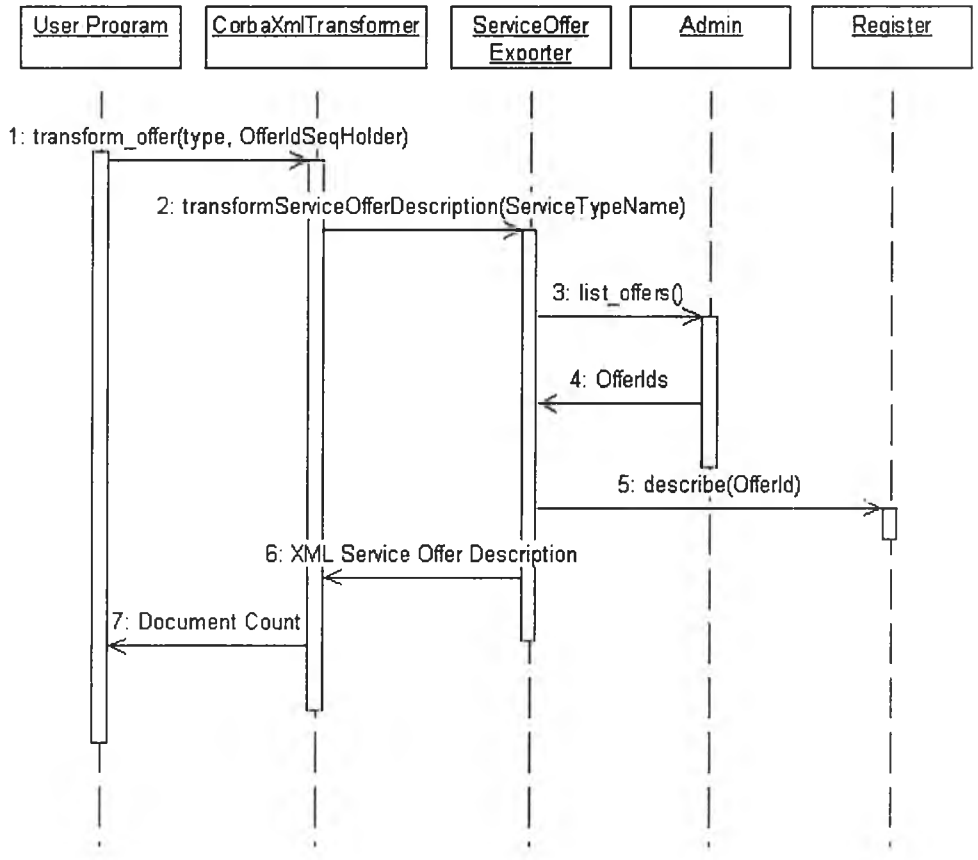
3. เมื่อมีการเรียกใช้ transform_all_offer() เพื่อแปลงคำอธิบายข้อเสนอบริการทั้งหมดภายในเทรดเดอร์ให้เป็นเอ็กซ์เอ็มแอล



รูปที่ 4.5 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_all_offer()

เมื่อมีการร้องขอจากโปรแกรมผู้ใช้งานเพื่อแปลงคำอธิบายข้อเสนอบริการทั้งหมดที่มีอยู่ในเทรดเดอร์ วัตถุของ ServiceOfferExporter จะถูกสร้างขึ้นเพื่อรองรับการแปลงนี้ โดยจะทำการเรียกดูชนิดของบริการทั้งหมดภายในเทรดเดอร์ (3: list_type()) จากนั้นจึงทำการแปลงคำอธิบายข้อเสนอบริการทั้งหมดของแต่ละชนิดของบริการจนครบทุกชนิดของบริการภายในเทรดเดอร์ จึงได้เอกสารเอ็กซ์เอ็มแอลของข้อเสนอบริการทั้งหมดออกมา (ดูหัวข้อถัดไปในการแปลงคำอธิบายข้อเสนอบริการของแต่ละชนิดของบริการ)

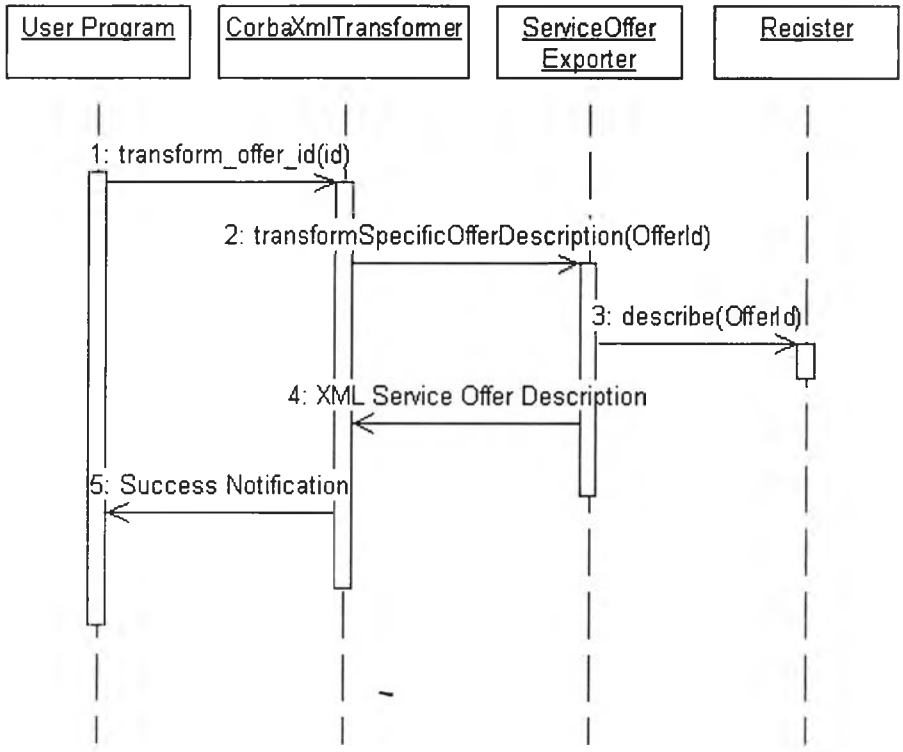
4. เมื่อมีการเรียกใช้ transform_offer() เพื่อแปลงคำอธิบายข้อเสนอบริการทั้งหมดที่มีชนิดของบริการตามที่กำหนดในพารามิเตอร์ ให้เป็นเอ็กซ์เอ็มแอล



รูปที่ 4.6 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_offer()

เมื่อมีการเรียกใช้ transform_offer() จากโปรแกรมผู้ใช้ วัตถุของ ServiceOfferExporter จะถูกสร้างขึ้นเพื่อเรียกใช้ transformServiceOfferDescription() โดยกำหนดชื่อชนิดของบริการเป็นพารามิเตอร์ วัตถุที่สร้างขึ้นจะทำการเรียกดูข้อเสนอบริการผ่านส่วนต่อประสาน Admin ภายในเทรดเดอร์ และนำชุดของตัวระบุข้อเสนอบริการที่มีชนิดของบริการเป็นไปตามที่กำหนดมาสอบถามคำอธิบายข้อเสนอบริการผ่านส่วนต่อประสาน Register เพื่อนำมาสร้างวัตถุดีโอเอ็มและแปลงคำอธิบายให้เป็นเอ็กซ์เอ็มแอลต่อไป

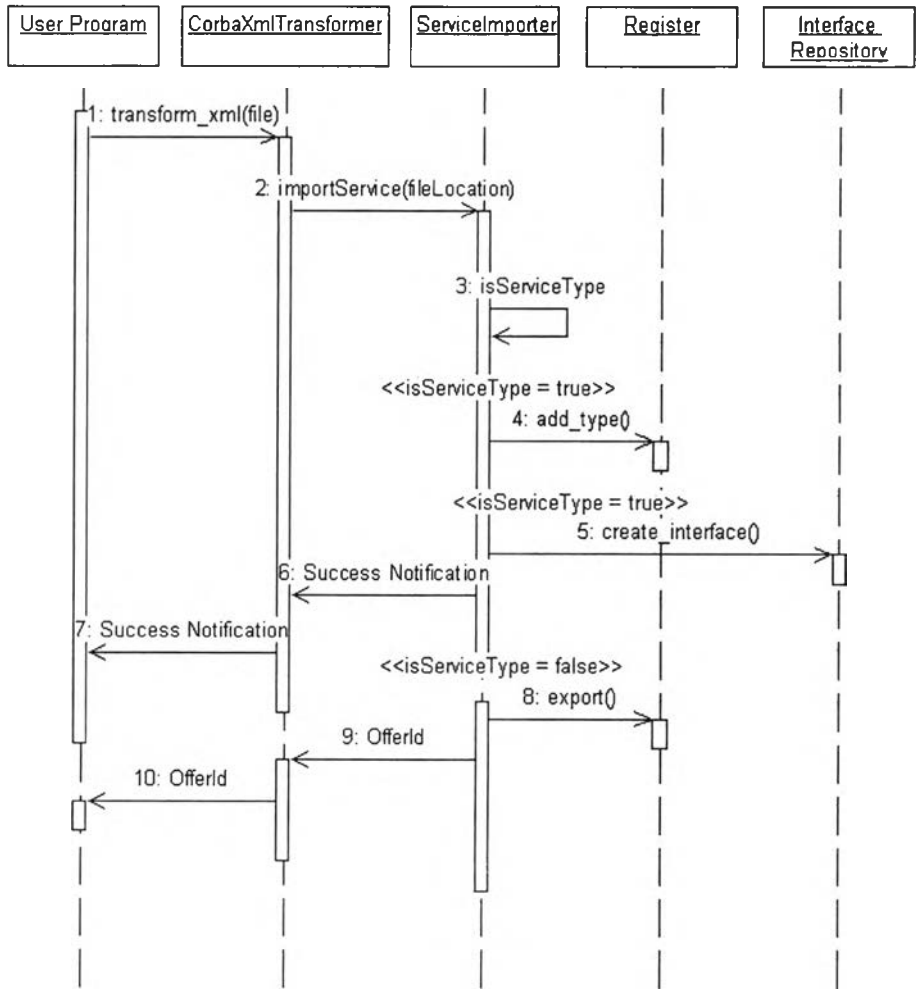
5. เมื่อมีการเรียกใช้ transform_offer_id() เพื่อแปลงคำอธิบายข้อเสนอบริการที่กำหนดโดยตัวระบุข้อเสนอบริการ ให้เป็นเอ็กซ์เอ็มแอล



รูปที่ 4.7 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_offer_id()

เมื่อมีการเรียกใช้ transform_offer_id() จากโปรแกรมผู้ใช้ วัตถุของ ServiceOfferExporter จะถูกสร้างขึ้นและเรียกใช้ transformSpecificOfferDescription() โดยระบุตัวระบุข้อเสนอบริการเพื่อค้นหาคำอธิบายข้อเสนอบริการจากส่วนต่อประสาน Register และสร้างวัตถุดีโอเอ็มของคำอธิบายข้อเสนอบริการ และแปลงเป็นเอกสารเอ็กซ์เอ็มแอลต่อไป

6. เมื่อมีการเรียกใช้ transform_xml() เพื่อแปลงคำอธิบายบริการ ทั้งชนิดของบริการ และข้อเสนอบริการที่อยู่ในรูปแบบเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคอร์บาเทรดเดอร์



รูปที่ 4.8 ผังลำดับเหตุการณ์เมื่อมีการเรียกใช้ transform_xml()

เมื่อมีการร้องขอให้บริการเทรดเดอร์ที่เพิ่มขยายทำการแปลงคำอธิบายจากเอ็กซ์เอ็มแอลให้อยู่ในรูปของคอร์บาเทรดเดอร์ วัตถุของ ServiceImporter จะถูกสร้างขึ้นเพื่อไปอ่านแฟ้มข้อมูลตามที่อยู่ที่ระบุในพารามิเตอร์ และสร้างวัตถุดีโอเอ็มจากแฟ้มนั้น รากของเอกสารจะถูกอ่านขึ้นมาเพื่อตรวจสอบว่าเป็นคำอธิบายชนิดของบริการ หรือของข้อเสนอบริการ โดยถ้าเป็นชนิดของบริการจะเรียกตัวกระทำการ add_type() ใน Register และ create_interface() ในคลังส่วนต่อประสานเพื่อสร้างคำอธิบายชนิดของบริการต่อไป หรือถ้าเอกสารเป็นคำอธิบายข้อเสนอบริการ บริการนี้จะถูกโฆษณาผ่าน export() ของ Register

4.2 การเพิ่มขยายเทรดเดอร์ของ JacORB เพื่อให้รองรับการแปลงคำอธิบาย

ดังที่กล่าวมาแล้วว่าตัวต้นแบบนี้พัฒนาขึ้นโดยอิงกับการพัฒนาคอร์บาของ JacORB นอกจากการแก้ไขส่วนของไอดีแอลมาตรฐานจากโอเอ็มจีแล้ว ยังต้องมีการแก้ไขในส่วนของรหัสต้นฉบับ (Source Code) ของบริการเทรดเดอร์ใน JacORB อีกด้วย โดยสามารถสรุปการแก้ไขในแฟ้มส่วนคลาสต่างๆ ดังนี้

- jacorb.trading.TradingService
  1. เพิ่มส่วนนำเข้าแพคเกจของ cxt โดยเพิ่มดังนี้

     ```
     import cxt.*;
     import cxt.util.*;
     ```

  2. เพิ่มส่วนประกาศสมาชิกของวัตถุ (Instance Member) สำหรับไดเรคทอรี่โดยปริยาย (Default Directory) ในการเก็บเอกสารเอ็กซ์เอ็มแอลดังนี้

     ```
     private static final String s_defaultXmlDbpath = "xmldb";
     ```

  3. เปลี่ยนส่วนการสร้าง jacorb.trading.impl.TraderComp ไปเป็น CorbaXmlTraderComp ดังนี้

     ```
     CorbaXmlTraderComp traderComp = new CorbaXmlTraderComp();
     ```

  4. เพิ่มส่วนการสร้างคลาส CorbaXmlTransformerImpl เพื่อใช้ในเทรดเดอร์ดังนี้

     ```
     CorbaXmlTransformerImpl transform = new CorbaXmlTransformerImpl(
                                                          getORB(),
                                                          intRep, lookup._this())
     transform._orb(orb);
     traderComp.setCorbaXmlTransformer(transform._this());
     ```

  5. เพิ่มส่วนอาร์กิวเมนต์ (Argument) เพื่อให้สามารถระบุที่เก็บเอกสารเอ็กซ์เอ็มแอลในส่วนการเริ่มบริการเทรดเดอร์ ดังนี้

     ```
     if(args.length == 4) {
           if(args[1].equals("-d")) {
             dbpath = args[2];
             xmldbpath = args[3];
             Properties xmldbProp = new Properties();
             xmldbProp.put("XmlDbPath",
                     dbpath+File.separator+xmldbpath+File.separator);
             jacorb.util.Environment.addProperties(xmldbProp);
           }
     }
     ```

- jacorb.trading.impl.AdminImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

- jacorb.trading.impl.LinkImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

- jacorb.trading.impl.LookupImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

- jacorb.trading.impl.ProxyImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

- jacorb.trading.impl.RegisterImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

- jacorb.trading.client.proxy.ProxyLookupImpl

  เพิ่มจุดเข้าถึงที่มาจากส่วนต่อประสาน CORBA::CosTrading::TraderComponents ดังนี้

```
public CorbaXmlTransformer transform_if() {
    cxt.CorbaXmlTraderComp traderComp =
                        (cxt.CorbaXmlTraderComp) m_traderComp;
    return traderComp.getCorbaXmlTransformer();
}
```

เมื่อทำการแก้ไขดังที่กล่าวมาแล้วจึงทำการแปลโปรแกรม (Compile) และนำไปใช้ต่อไป