

## บทที่ 5

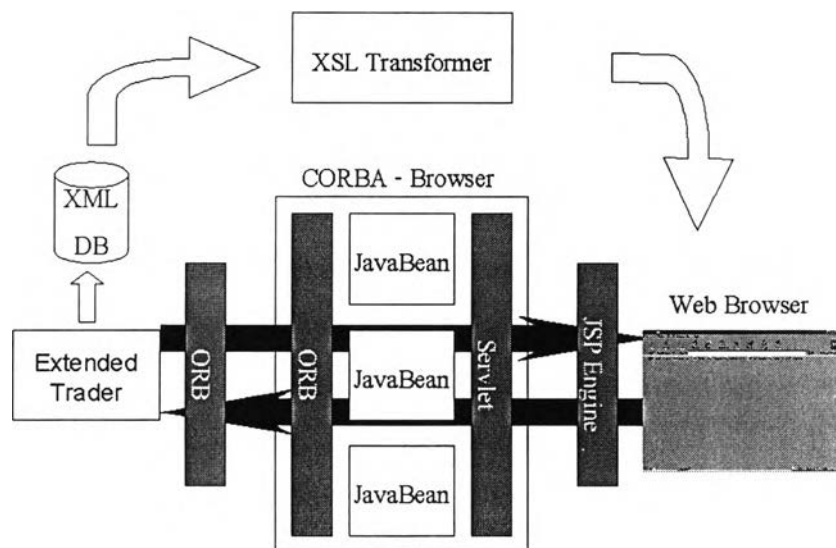
### การทดสอบการเรียกใช้งานต้นแบบ และการทดสอบการแปลงคำอธิบายบริการ ระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล

ในบทนี้จะได้กล่าวถึงตัวต้นแบบในการเรียกใช้งานเทรดเดอร์ที่มีการเพิ่มขยาย และการทดสอบการแปลงคำอธิบายทั้งชนิดของบริการ และข้อเสนอบริการจากรูปแบบคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล และจากเอ็กซ์เอ็มแอลไปเป็นรูปแบบคอร์บาเทรดเดอร์ ว่าเป็นไปตามที่กำหนดหรือไม่ โดยจะได้กล่าวถึงเป็นหัวข้อย่อยดังนี้คือ

- โปรแกรมการเรียกใช้งานต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย
- สภาพะที่ใช้ในการทดสอบ
- กรณีที่ใช้ในการทดสอบการเรียกใช้งาน และการแปลงคำอธิบาย
- ขั้นตอนการทดสอบ
- ผลการทดสอบ

#### 5.1 โปรแกรมการเรียกใช้งานต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย

ในการทดสอบการแปลงคำอธิบายบริการ ผู้วิจัยได้ทำการสร้างส่วนของโปรแกรมเพื่อเรียกใช้การแปลงคำอธิบาย และแสดงผลพันธในการแปลงโดยอยู่ในรูปแบบที่สามารถเรียกใช้ผ่านโปรแกรมค้นผ่านเว็บ ซึ่งในส่วนของโปรแกรมการเรียกใช้งานนี้จะทำงานบนเทคโนโลยีหลักคือ จาวาเซิร์ฟเลต (Java Servlet) รุ่น 2.3 และเจเอสพี (JavaServer Pages (JSP)) รุ่น 1.2 เพื่อสร้างเอกสารเอ็กซ์เอ็มแอลสำหรับโปรแกรมค้นผ่านเว็บ และเอ็กซ์เอ็มแอลที่จากออราเคิล ในการแปลงเอกสารเอ็กซ์เอ็มแอล สถาปัตยกรรมของโปรแกรมการเรียกใช้งานสามารถแสดงได้ในรูปที่ 5.1

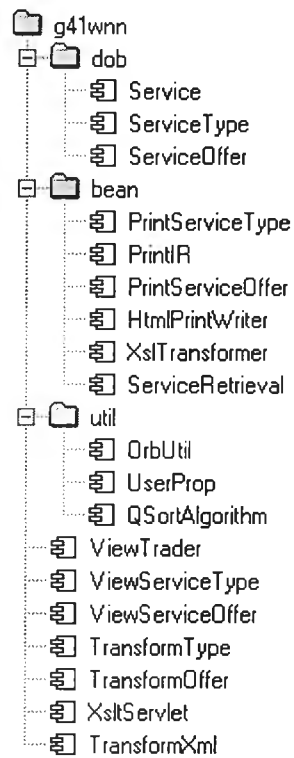


รูปที่ 5.1 สถาปัตยกรรมของโปรแกรมการเรียกใช้งานต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย

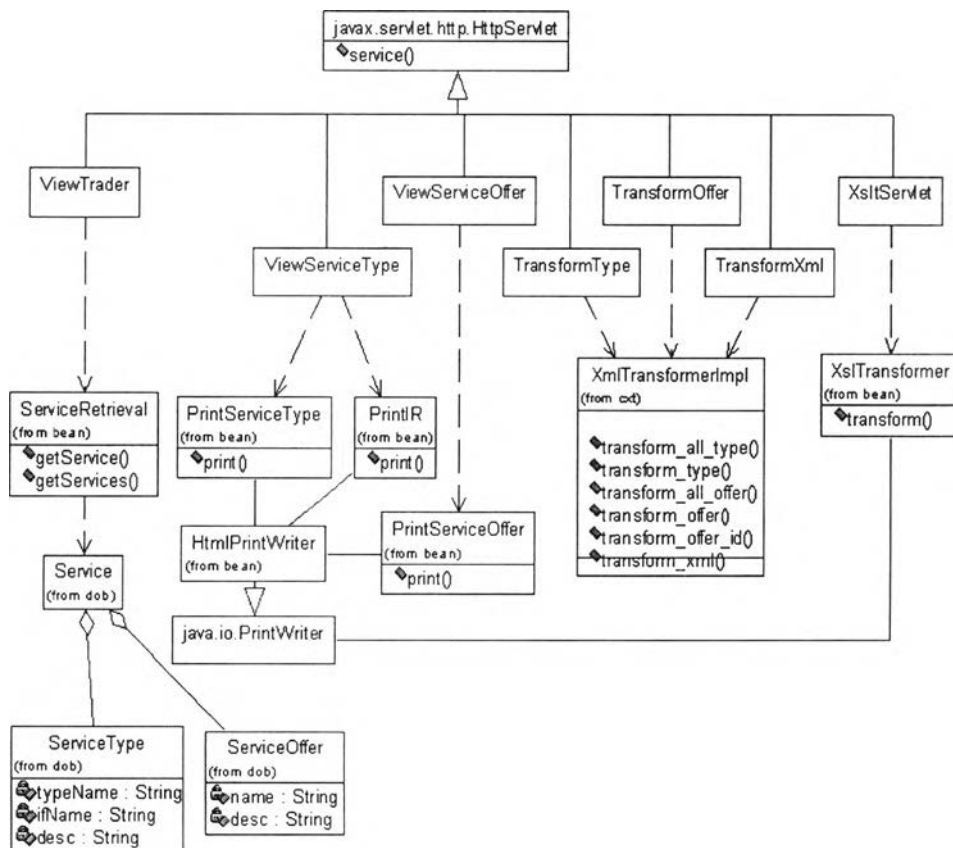
ส่วนของโปรแกรมจะประกอบไปด้วยคลาสหลักที่ทำหน้าที่ในการรับคำร้องขอจากโปรแกรมค้นผ่านเว็บสำหรับการเรียกใช้คอมโพเนนต์ของจาวาที่เรียกว่าจาวา빈 (JavaBeans) อยู่ 7 คลาสคือ

- `g41wnn.ViewTrader` สำหรับการดึงชนิดของบริการ และข้อเสนอบริการที่มีอยู่ทั้งหมดมาแสดงบนโปรแกรมค้นผ่านเว็บ โดยการเรียกใช้ `g41wnn.bean.ServiceRetrieval` และส่งไปยัง `TraderViewer.jsp`
- `g41wnn.ViewServiceType` สำหรับการแสดงคำอธิบายชนิดของบริการที่กำหนดให้อยู่ในรูปของเอชทีเอ็มแอล โดยการเรียกผ่านจาวาบีที่ทำงานเพื่อนำข้อมูลชนิดของบริการคือ `g41wnn.bean.PrintServiceType` และข้อมูลส่วนต่อประสานคือ `g41wnn.bean.PrintIR` มาสร้างเป็นเอชทีเอ็มแอล และส่งผลลัพธ์ที่ได้ไปยัง `ServiceTypeViewer.jsp`
- `g41wnn.ViewServiceOffer` สำหรับการแสดงคำอธิบายข้อเสนอบริการที่กำหนดให้อยู่ในรูปของเอชทีเอ็มแอล โดยการเรียกผ่าน `g41wnn.bean.PrintServiceOffer` มาสร้างเป็นเอชทีเอ็มแอล และส่งผลลัพธ์ที่ได้ไปยัง `ServiceOfferViewer.jsp`
- `g41wnn.TransformType` สำหรับการเรียกใช้งาน `transform_type()` ในบริการเทรดเดอร์เพื่อแปลงคำอธิบายชนิดของบริการจากรูปแบบของคอร์บาไปเป็นเอ็กซ์เอ็มแอลตามชนิดของบริการที่กำหนด และส่งต่อไปยัง `XmlServiceViewer.jsp`
- `g41wnn.TransformOffer` สำหรับการเรียกใช้งาน `transform_offer_id()` ในบริการเทรดเดอร์เพื่อแปลงคำอธิบายข้อเสนอบริการจากรูปแบบของคอร์บาไปเป็นเอ็กซ์เอ็มแอลตามตัวระบุข้อเสนอบริการที่กำหนด และส่งต่อไปยัง `XmlServiceViewer.jsp`
- `g41wnn.TransformXml` สำหรับการเรียกใช้งาน `transform_xml()` ในบริการเทรดเดอร์เพื่อแปลงคำอธิบายจากเอ็กซ์เอ็มแอลไปเป็นรูปแบบของคอร์บาเทรดเดอร์จากเพิ่มข้อมูลเอ็กซ์เอ็มแอลที่กำหนด
- `g41wnn.XslServlet` สำหรับการเรียก `g41wnn.bean.XslTransformer` ซึ่งเป็นตัวเอ็กซ์เอสแอลที่สำหรับการแปลง และนำเอกสารเอ็กซ์เอ็มแอลภายในเทรดเดอร์มาแสดงเป็นเอชทีเอ็มแอลบนโปรแกรมค้นผ่านเว็บ

รูปที่ 5.2 และ 5.3 จะแสดงถึงโครงสร้างแพ็คเกจและแผนผังคลาสโปรแกรมการเรียกใช้งานต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยายตามลำดับ ในส่วนของการใช้งานโปรแกรมการเรียกใช้งานต้นแบบนี้สามารถดูรายละเอียดได้ในภาคผนวก ก



รูปที่ 5.2 โครงสร้างแพ็คเกจของคลาสโปรแกรมการเรียกใช้งานต้นแบบบริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย



รูปที่ 5.3 แผนผังคลาสของโปรแกรมการเรียกใช้งานต้นแบบเทรดเดอร์ที่มีส่วนเพิ่มขยาย

## 5.2 สภาพะที่ใช้ในการทดสอบ

ในการทดสอบการเรียกใช้ และการแปลงคำอธิบายเป็นการทดสอบโดยผ่านข้อเสนอบริการเว็บ (Web Server) Apache รุ่น 1.3.6 และข้อเสนอบริการเซิร์ฟเลต และเจเอสพี (Servlet and JSP Server) Tomcat รุ่น 3.1 ภายใต้ระบบปฏิบัติการ Linux รุ่น 2.2.13 ในขณะที่โปรแกรมค้นผ่านเว็บคือ Internet Explorer รุ่น 4.0 บนระบบปฏิบัติการ Windows98

สำหรับบริการเทอร์ตเตอร์ คลังส่วนต่อประสาน และบริการที่ใช้ในการทดสอบจะให้บริการบนเครื่องระบบปฏิบัติการ Linux รุ่น 2.2.13 ภายใต้เครื่องเสมือนของจาวา (Java Virtual Machine) รุ่น 1.2.2

## 5.3 กรณีที่ใช้ในการทดสอบการเรียกใช้ และการแปลงคำอธิบาย

ในการทดสอบการเรียกใช้ และการแปลงคำอธิบาย ส่วนของการแปลงคำอธิบายชนิดของบริการ จากคอร์บาเทอร์ตเตอร์ไปเป็นเอ็กซ์เอ็มแอลจะใช้คำอธิบายส่วนต่อประสานในตารางที่ 5.1 และคำอธิบายชนิดของบริการในตารางที่ 5.2 โดยกำหนดให้มีชนิดของบริการหกชนิดคือ

- Account
- FixedAccount
- CheckingAccount
- SocketConnection
- TCPConnection
- UDPConnection
- Fraction

โดยชนิดของบริการทั้งหกชนิดนี้จะครอบคลุมชนิดของข้อมูลที่เป็นพื้นฐานสำหรับไอดีแอลที่กำหนดคือ short, long, unsigned short, unsigned long, float, double, char, string, boolean และ octet และเมื่อแปลงเป็นเอ็กซ์เอ็มแอลแล้วจะประกอบไปด้วยส่วนย่อยทุกส่วนที่กำหนดในดีทีดีของคำอธิบายชนิดของบริการ

ในส่วนการแปลงคำอธิบายข้อเสนอบริการจากคอร์บาเทอร์ตเตอร์ไปเป็นเอ็กซ์เอ็มแอล ในแต่ละชนิดของบริการจะประกอบไปด้วยข้อเสนอบริการสองข้อเสนอ โดยมีข้อเสนอบริการที่ใช้คุณสมบัติแบบพลวัตอย่างน้อยหนึ่งคุณสมบัติตามตารางที่ 5.3 และมีการใช้ส่วนย่อยครบทุกส่วนที่กำหนดในดีทีดีของคำอธิบายข้อเสนอบริการ

ในการทดสอบการแปลงคำอธิบายบริการจากเอ็กซ์เอ็มแอลไปเป็นรูปแบบของคอร์บาเทอร์ตเตอร์ จะใช้เอกสารเอ็กซ์เอ็มแอลที่ได้จากการแปลงข้างต้นมาแปลงกลับอีกครั้งให้อยู่ในรูปแบบของคอร์บาเทอร์ตเตอร์ โดยนำคำอธิบายเหล่านี้แปลงไปยังบริการเทอร์ตเตอร์ และคลังส่วนต่อประสานอีกชุดหนึ่งซึ่งยังไม่มีข้อมูลโดยอยู่ ผลลัพธ์จากการแปลงจะถูกเปรียบเทียบกับคำอธิบายบริการต้นฉบับเพื่อตรวจสอบความถูกต้องอีกครั้งหนึ่ง

## ตารางที่ 5.1 ส่วนต่อประสานที่ใช้ในการทดสอบ

```
1. module bank {
2.     interface Account {
3.         const unsigned short min_balance = 0;
4.         attribute string acc_name;
5.         readonly attribute string acc_no;
6.         exception Overdrawn {
7.             double amount;
8.         };
9.         void deposit(in double amount);
10.        void withdraw(in double amount) raises (Overdrawn);
11.        double balance();
12.    };
13.    interface FixedAccount : Account {
14.        const short interval = 3;
15.        attribute float interest_rate;
16.        double cal_interest();
17.    };
18.    interface CheckingAccount : Account {
19.        attribute string checking_no;
20.        attribute float apr;
21.        double cal_interest();
22.        void transfer(in double amount) raises (Overdrawn);
23.    };
24. };
```

## ตารางที่ 5.1 ส่วนต่อประสานที่ใช้ในการทดสอบ (ต่อ)

```
25. module connection {
26.     interface PlainConnection {
27.         attribute unsigned long port;
28.         attribute long timeout;
29.     };

30.     interface TCPConnection : PlainConnection {
31.         boolean send(in octet data);
32.         boolean sendChar(in char data);
33.         boolean send_and_receive(inout octet data) context("Environment");
34.         void receive(out octet data);
35.         void receiveChar(out char data);
36.     };

37.     interface UDPConnection : PlainConnection {
38.         oneway void send(in octet data);
39.         oneway void sendChar(in char data);
40.         void receive(out octet data);
41.         void receiveChar(out char data);
42.     };
43. };

44. module number {
45.     interface Fraction {
46.         struct Numeric {
47.             float base;
48.             short exponent;
49.         };
50.         typedef string frac_name;

51.         attribute Numeric value;
52.         attribute frac_name alphabetic;
53.     };
54. };
```

ตารางที่ 5.2 ชนิดของบริการที่ใช้ในการทดสอบ

ชนิดของบริการ	คัมภีร์อธิบายชนิดของบริการ
Account	ส่วนต่อประสาน: bank::Account คุณสมบัติ: mandatory string BankName double ReservedFund
FixedAccount	ส่วนต่อประสาน: bank::FixedAccount คุณสมบัติ: -
CheckingAccount	ส่วนต่อประสาน: bank::CheckingAccount คุณสมบัติ: -
PlainConnection	ส่วนต่อประสาน: connection::PlainConnection คุณสมบัติ: readonly long max_conn mandatoryreadonly long max_buffer
TCPConnection	ส่วนต่อประสาน: connection::TCPConnection คุณสมบัติ: boolean TCP_NODELAY
UDPConnection	ส่วนต่อประสาน: connection::UDPConnection คุณสมบัติ: unsigned short TTL
Fraction	ส่วนต่อประสาน: number::Fraction คุณสมบัติ: boolean NaN

ตารางที่ 5.3 ข้อเสนอแนะบริการที่ใช้ในการทดสอบ

ชนิดของบริการ	ข้อเสนอแนะบริการ
Account	คุณสมบัติ: BankName = "Chula1" ReservedFund = 1000000
	คุณสมบัติ: BankName = "Chula2" ReservedFund = {Dynamic Extra_info = "ReservedFund Extra Info" }
FixedAccount	คุณสมบัติ: BankName = "Chula3" ReservedFund = 3000000
	คุณสมบัติ: BankName = "Chula4"
CheckingAccount	คุณสมบัติ: BankName = "Chula5"
	คุณสมบัติ: BankName = "Chula6"
PlainConnection	คุณสมบัติ: max_conn = 3000 max_buffer = 6000000
	คุณสมบัติ: max_conn = 5000 max_buffer = 90000000
TCPConnection	คุณสมบัติ: max_conn = 6000 max_buffer = 5000000 TCP_NODELAY = true
	คุณสมบัติ: max_conn = 2000 max_buffer = 10000000
UDPConnection	คุณสมบัติ: max_conn = 2000 max_buffer = 500000 TTL = {Dynamic extra_info = "TTL Extra Info" }
	คุณสมบัติ: max_conn = 3000 max_buffer = 300000 TTL = 1054



## 5.4 ขั้นตอนการทดสอบ

ในการทดสอบการเรียกใช้งานต้นแบบ และการทดสอบการแปลงคำอธิบายบริการ ระบบที่จะทดสอบจะต้องถูกเตรียมพร้อมก่อน โดยมีลำดับขั้นตอนดังนี้

1. เริ่มใช้งานคลังส่วนต่อประสานโดยใช้นิยามส่วนต่อประสานตัวอย่างในตารางที่ 5.1 ซึ่งบันทึกไว้ในแฟ้มที่ชื่อว่า TestCase.idl
2. เริ่มใช้งานบริการชื่อของ JacORB
3. สร้างแฟ้มไอโออาร์ของคลังส่วนต่อประสานโดยใช้โปรแกรมในการสร้างที่ชื่อว่า IR2Obj เพื่อใช้ในการระบุที่อยู่ของคลังส่วนต่อประสานสำหรับ JacORB เพื่อการเรียกใช้ CORBA::ORB::resolve\_initial\_references("InterfaceRepository")
4. เริ่มใช้งานบริการทรอดเดอร์ของ JacORB
5. ทำการลงทะเบียนบริการทรอดเดอร์ และคลังส่วนต่อประสานกับบริการชื่อ โดยใช้ชื่อว่า ts1 และ ir1 ตามลำดับ
6. ทำการเพิ่มชนิดของบริการในตาราง 5.2 และข้อเสนอบริการในตารางที่ 5.3 โดยใช้โปรแกรมที่ชื่อว่า Server
7. เริ่มใช้งานคลังส่วนต่อประสานเปล่าขึ้นภายใต้เครื่องเสมือนอื่น และทำเช่นเดียวกับข้อ 2-5 โดยใช้ชื่อของทรอดเดอร์ว่า ts2 และชื่อของคลังส่วนต่อประสานว่า ir2
8. เปิดโปรแกรมค้นผ่านเว็บ และเรียกใช้งานเจฟเจ็ต g41wnn.ViewTrader โดยมีพารามิเตอร์คือ ?TSName=ts1&IRName=ir1 เพื่อทำการเรียกใช้การแปลง ซึ่งการใช้งานโปรแกรมทดสอบการเรียกใช้สามารถดูรายละเอียดได้ในภาคผนวก ก
9. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ Account
10. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ CheckingAccount
11. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ FixedAccount
12. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ PlainConnection
13. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ TCPConnection
14. ทดสอบการแปลงคำอธิบายชนิดของบริการ และข้อเสนอบริการ UDPConnection
15. ทดสอบการแปลงคำอธิบายชนิดของบริการ Fraction

## 5.5 ผลการทดสอบ

ในการทดสอบการแปลงคำอธิบายบริการจากรูปแบบของคอร์บาทรอดเดอร์ไปเป็นเอ็กซ์เอ็มแอลผลลัพธ์ที่ได้จะอยู่ในรูปเอกสารเอ็กซ์เอ็มแอลของคำอธิบายชนิดของบริการทั้งหกชนิด และข้อเสนอบริการทั้งหมดสิบสองรายการตามตารางที่ 5.2 และ 5.3 ตามลำดับ โดยเอกสารผลลัพธ์ที่ได้จากการทดสอบจะแสดงในภาคผนวก ข สำหรับชนิดของบริการ และภาคผนวก ค สำหรับข้อเสนอบริการ เอกสารเอ็กซ์เอ็มแอลเหล่านี้จะถูกพิจารณาโดยตัวผู้วิจัยว่าข้อมูลที่แปลงแล้วสอดคล้องกับข้อมูลต้นฉบับหรือไม่ จากการ

ตรวจสอบพบว่าเอกสารอิเล็กทรอนิกส์ที่ได้จากการแปลงมีความสอดคล้องกับคำอธิบายบริการที่อยู่ในรูปแบบของคอร์ปอเรเตอร์ โดยสามารถรองรับชนิดของข้อมูลพื้นฐานที่กำหนด และถูกต้องตามที่ดีของเอกสาร

สำหรับการทดสอบการแปลงคำอธิบายบริการจากอิเล็กทรอนิกส์ไปเป็นรูปแบบของคอร์ปอเรเตอร์ เทเรเตอร์ ts2 และคลังส่วนต่อประสาน ir2 จะถูกตรวจสอบว่ามีข้อมูลสอดคล้องกับเทเรเตอร์ และคลังส่วนต่อประสานต้นทางหรือไม่ จากโปรแกรมทดสอบการเรียกใช้งานต้นแบบในการเรียกดูภายในของเทเรเตอร์ และผลลัพธ์ของแฟ้มไอดีแอลจากคลังส่วนต่อประสาน และรายละเอียดชนิดของบริการปลายทางดังมีรายละเอียดในภาคผนวก ง ผลลัพธ์ที่ได้แสดงความสอดคล้องกันของเทเรเตอร์ และคลังส่วนต่อประสานทั้งสอง จึงสามารถบ่งบอกได้ว่าการแปลงคำอธิบายบริการจากอิเล็กทรอนิกส์ไปเป็นรูปแบบของคอร์ปอเรเตอร์เป็นไปอย่างถูกต้อง