

## บทที่ 4

### ต้นแบบคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

ต้นแบบคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันในวิทยานิพนธ์นี้ มีการพัฒนาตามขั้นตอนที่สอดคล้องกับข้อกำหนดของสถาปัตยกรรมแบบคอร์บาและนิยามของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันที่ได้กล่าวถึงรายละเอียดไปแล้วในบทที่ 3 โดยใช้ภาษาการโปรแกรมจาวาเวอร์ชัน 1.1.7 ร่วมกับ สวิง (Swing) เวอร์ชัน 1.1.1 และเจดีบีซี (JDBC – Java Database Connectivity) ซึ่งสามารถนำไปใช้ในสภาวะแวดล้อมการทำงานของจาวา (Java Runtime Environment -- JRE) ตั้งแต่เวอร์ชัน 1.1.7 ขึ้นไปได้

การออกแบบต้นแบบจะประกอบไปด้วย 3 ส่วนคือ

- 1) ส่วนฐานข้อมูลที่ใช้เก็บรายละเอียดของข้อมูลของความสัมพันธ์แบบเท่าเทียมกัน
- 2) ส่วนการออกแบบคลาสต่างๆของต้นแบบ (Class Diagram)
- 3) ส่วนต่อประสานสำหรับผู้ใช้ของต้นแบบ

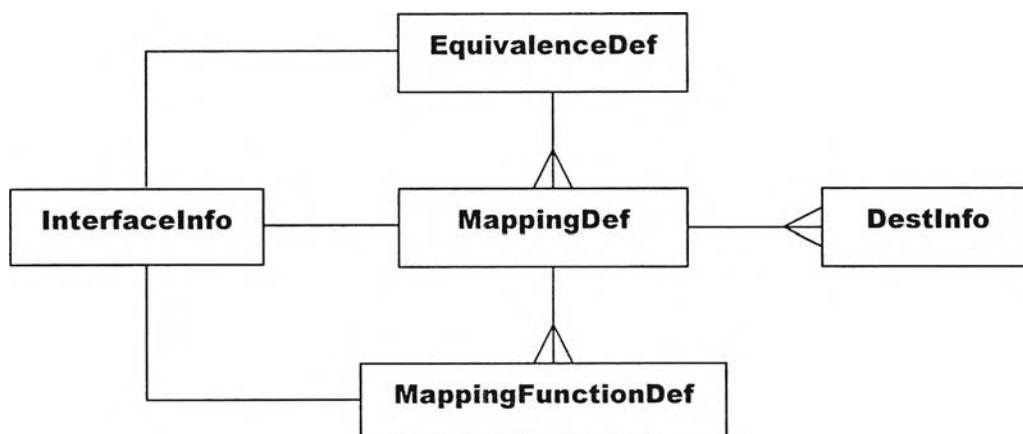
ภาพรวมการทำงานของต้นแบบ มีส่วนเกี่ยวข้องกับทั้ง 3 ส่วนที่ได้ออกแบบไว้ดังนี้ เมื่อผู้เปลี่ยนรุ่นของบริการทำการดำเนินงาน (Run) โปรแกรมต้นแบบ โปรแกรมต้นแบบจะแสดงส่วนต่อประสานสำหรับผู้ใช้ที่สามารถใส่ข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสาน ได้แก่ข้อมูลชนิด EquivalenceDef, MappingDef และ MappingFunctionDef จากนั้นโปรแกรมต้นแบบก็จะทำการจัดเก็บข้อมูลเหล่านั้นลงสู่ฐานข้อมูล ซึ่งสามารถดูขั้นตอนการใส่ข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสานสามารถดูได้ในบทที่ 5 ในบทนี้จะอธิบายรายละเอียดของการออกแบบ 2 ส่วนแรก สำหรับส่วนที่ 3 คือส่วนต่อประสานสำหรับผู้ใช้ของต้นแบบสามารถดูได้ที่ภาคผนวก ก.

#### 4.1 ส่วนการออกแบบฐานข้อมูลของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

ส่วนของฐานข้อมูลของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันที่ออกแบบในวิทยานิพนธ์นี้ แบ่งย่อยออกเป็น 2 ส่วนคือ

##### 4.1.1 ส่วนแบบจำลองข้อมูล (Data Modeling)

แบบจำลองข้อมูลดังแสดงในรูปที่ 4.1 เป็นแบบจำลองข้อมูลเชิงสัมพันธ์ ดังนั้นเราสามารถนำแบบจำลองข้อมูลนี้ไปทำให้เกิดผลได้ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational DBMS) ใดก็ได้



รูปที่ 4.1 แบบจำลองข้อมูล (Data Modeling) ของข้อมูลความสัมพันธ์แบบเท่าเทียมกัน  
ในคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

แบบจำลองที่ได้ออกแบบในวิทยานิพนธ์นี้มีทั้งสิ้น 5 เอนทิตี (Entity) คือ

- เอนทิตี InterfaceInfo ซึ่งเป็นเอนทิตีที่สะท้อนถึงข้อมูลพื้นฐานของส่วนต่อประสาน และจะเกี่ยวกับทุกๆ เอนทิตีในทางอ้อมคือ หากต้องการข้อมูลพื้นฐานของแต่ละส่วนต่อประสาน ก็จะต้องใช้ข้อมูลจากเอนทิตีนี้ กล่าวได้อีกนัยว่าเอนทิตี InterfaceInfo นี้เป็นเอนทิตีอรรถประโยชน์ (Utility Entity) สำหรับเอนทิตีอื่นๆ
- เอนทิตี EquivalenceDef ซึ่งเป็นเอนทิตีที่บ่งบอกถึงความสัมพันธ์แบบเท่าเทียมกันในเบื้องต้น โดยเอนทิตีนี้มีความสัมพันธ์แบบหนึ่งต่อหลาย (One-to-Many Relationship) กับเอนทิตี MappingDef
- เอนทิตี MappingDef ซึ่งเป็นเอนทิตีที่บ่งบอกถึงความสัมพันธ์แบบเท่าเทียมกันในระดับของอินสแตนซ์ต้นทางและปลายทางของส่วนต่อประสานที่มีความสัมพันธ์แบบเท่าเทียมกัน โดยเอนทิตีนี้มีความสัมพันธ์แบบหนึ่งต่อหลายกับเอนทิตี MappingFunctionDef และเอนทิตี DestInfo
- เอนทิตี DestInfo ซึ่งเป็นเอนทิตีที่บ่งบอกถึงอินสแตนซ์ปลายทางของส่วนต่อประสานที่มีความสัมพันธ์แบบเท่าเทียมกัน
- เอนทิตี MappingFunctionDef ซึ่งเป็นเอนทิตีที่บ่งบอกถึงรูปแบบความสัมพันธ์แบบเท่าเทียมกันในลักษณะของฟังก์ชันการแปลง

#### 4.1.2 ส่วนตารางในฐานข้อมูลเชิงสัมพันธ์

เราสามารถนำแบบจำลองข้อมูลในรูปที่ 4.1 ไปทำให้เกิดผลได้ในระบบจัดการฐานข้อมูลเชิงสัมพันธ์ โดยในวิทยานิพนธ์นี้ได้ใช้ระบบจัดการฐานข้อมูล Oracle ในการทำให้เกิดผล โดยมีตารางที่ใช้เก็บข้อมูลต่างๆที่สอดคล้องกับแบบจำลองข้อมูลที่กล่าวไปแล้ว ดังนี้

1. EquivalenceDef เก็บรายละเอียดของข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสาน (ดังตารางที่ 4.1)
2. MappingDef เก็บรายละเอียดเพิ่มเติมของข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสานโดยเน้นข้อมูลของอินสแตนซ์ต้นทางของส่วนต่อประสานที่มีความสัมพันธ์แบบเท่าเทียมกัน (ดังตารางที่ 4.2)
3. DestInfo เก็บรายละเอียดเพิ่มเติมของข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสานโดยเน้นข้อมูลของอินสแตนซ์ปลายทางของส่วนต่อประสานที่มีความสัมพันธ์แบบเท่าเทียมกัน (ดังตารางที่ 4.3)
4. MappingFunctionDef เก็บรายละเอียดเพิ่มเติมของข้อมูลความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสานโดยเน้นข้อมูลของฟังก์ชันการแปลงที่ทำให้เกิดความโปร่งใสในการเปลี่ยนแปลงชนิดของบริการ (ดังตารางที่ 4.4)
5. InterfaceInfo เก็บรายละเอียดของข้อมูลพื้นฐานของแต่ละส่วนต่อประสาน (ดังตารางที่ 4.5)

ตารางที่ 4.1 ตาราง EquivalenceDef

เขตข้อมูล	ชนิดข้อมูล	รายละเอียด
eq_def_id	VARCHAR(100)	Repository Id ของ EquivalenceDef Interface
origin_if_id	VARCHAR(100)	Repository Id ของ Original Interface
eq_if_id	VARCHAR(100)	Repository Id ของ Equivalence Interface ที่มีความสัมพันธ์แบบเท่าเทียมกันกับ Original Interface
evolve_level	VARCHAR(8)	ระดับของการแปลงซึ่งมี 2 ระดับคือ ระดับชนิดและระดับอินสแตนซ์

ตารางที่ 4.2 ตาราง MappingDef

เขตข้อมูล	ชนิดข้อมูล	รายละเอียด
mapping_if_id	VARCHAR(100)	Repository Id ของ MappingDef Interface
mapping_ref	LONG	IOR ของ MappingDef Interface
src_obj_name	VARCHAR(50)	ชื่อวัตถุที่แหล่งกำเนิด
src_server_name	VARCHAR(50)	ชื่อเครื่องบริการที่แหล่งกำเนิด
src_host_name	VARCHAR(50)	ชื่อแม่ข่ายที่แหล่งกำเนิด
eq_def_id	VARCHAR(100)	RepositoryId ที่เป็นคีย์นอก (foreign key) ที่ใช้อ้างถึง ตาราง EquivalenceDef

ตารางที่ 4.3 ตาราง DestInfo

เขตข้อมูล	ชนิดข้อมูล	รายละเอียด
mapping_if_id	VARCHAR(100)	RepositoryId ที่เป็นคีย์นอกที่ใช้อ้างถึงตาราง MappingDef
dest_obj_name	VARCHAR(50)	ชื่อวัตถุที่ปลายทาง
dest_server_name	VARCHAR(50)	ชื่อเครื่องบริการที่ปลายทาง
dest_host_name	VARCHAR(50)	ชื่อแม่ข่ายที่ปลายทาง

ตารางที่ 4.4 ตาราง MappingFunctionDef

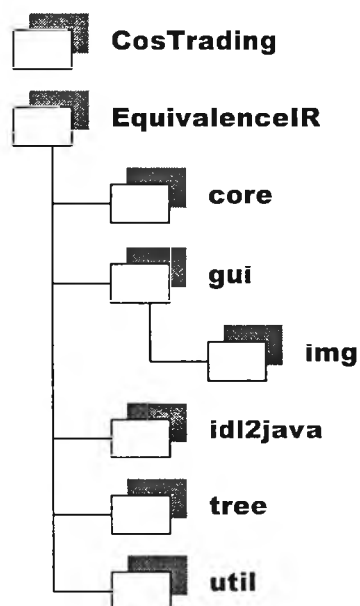
เขตข้อมูล	ชนิดข้อมูล	รายละเอียด
mapping_fn_id	VARCHAR(100)	Repository Id ของ MappingFunctionDef Interface
gen_mode	VARCHAR(12)	โหมดในการสร้างตัวดำเนินการแปลง ซึ่งมีได้ 2 ค่าคือ direct และ user_defined
map_mode	VARCHAR(12)	โหมดในการแปลงข้อมูล ซึ่งมีได้ 5 ค่าคือ attribute, argument, result, operation, exception และ instance
mapFnBody	LONG	เพิ่มฟังก์ชันการแปลงซึ่งจะนำมาใช้ในการสร้าง ตัวดำเนินการแปลงและสามารถใช้ดูพฤติกรรมกรรมการแปลง
mapping_if_id	VARCHAR(100)	RepositoryId ที่เป็นคีย์นอกที่ใช้อ้างถึงตาราง MappingDef

ตารางที่ 4.5 ตาราง InterfaceInfo

เขตข้อมูล	ชนิดข้อมูล	รายละเอียด
id	VARCHAR(100)	Repository Id ของ Interface
name	VARCHAR(40)	ชื่อ
version	VARCHAR(10)	รุ่น
defined_in	VARCHAR(100)	Repository Id ของ Parent Interface
absolute_name	VARCHAR(100)	absolute name ของ Interface ที่เป็นไปตามข้อกำหนดของคอร์บา
containing_repository		วัตถุ Repository ที่เป็นตัวบรรจุวัตถุต่างๆในคลังส่วนต่อประสาน

#### 4.2 ส่วนการออกแบบคลาสต่างๆสำหรับต้นแบบของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

การออกแบบคลาสต่างๆสำหรับต้นแบบในวิทยานิพนธ์นี้ ใช้หลักการแบ่งออกเป็นหมวดหมู่ตามหน้าที่การทำงานที่เกี่ยวข้องกัน โดยใช้แนวคิดการจัดแบ่งคลาสต่างๆออกเป็นแพ็คเกจ (Package) ของจาวา ซึ่งช่วยให้การพัฒนาต้นแบบมีความชัดเจน เข้าใจง่าย และเป็นระเบียบ โดยแพ็คเกจที่แบ่งออกมามีโครงสร้างดังรูปที่ 4.2



รูปที่ 4.2 โครงสร้างแพ็คเกจของคลาสในคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

#### 4.2.1 แพคเกจ CosTrading

แพคเกจนี้เป็นแพคเกจที่เกี่ยวข้องกับบริการเทรดเดอร์ ในวิทยานิพนธ์นี้ไม่ได้มุ่งเน้นที่การทำงานที่เกี่ยวข้องกับเทรดเดอร์ แต่มีไว้เพื่อความสมบูรณ์ของนิยามของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน เพื่อให้สามารถนำนิยามนี้ไปใช้ขยายผลได้ในงานอื่น ๆ ที่ต้องการข้อมูลจากคลังส่วนต่อประสานนี้เพื่อทำงานร่วมกับเทรดเดอร์ (ดูข้อเสนอแนะในบทที่ 6) ทั้งนี้เพราะตามแนวคิดของระบบกระจายนั้นคลังชนิดของบริการจะหมายรวมถึงคลังที่เทรดเดอร์ใช้ในการทำงานด้วย แต่ในสถาปัตยกรรมคอร์บาได้ทำการแยกคลังส่วนต่อประสานกับคลังชนิดที่เทรดเดอร์ใช้ออกจากกัน ดังนั้นคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันในวิทยานิพนธ์นี้จึงมีนิยามของเทรดเดอร์รวมอยู่ด้วย เพื่อความสมบูรณ์ตามแนวคิดเดิมของระบบกระจาย

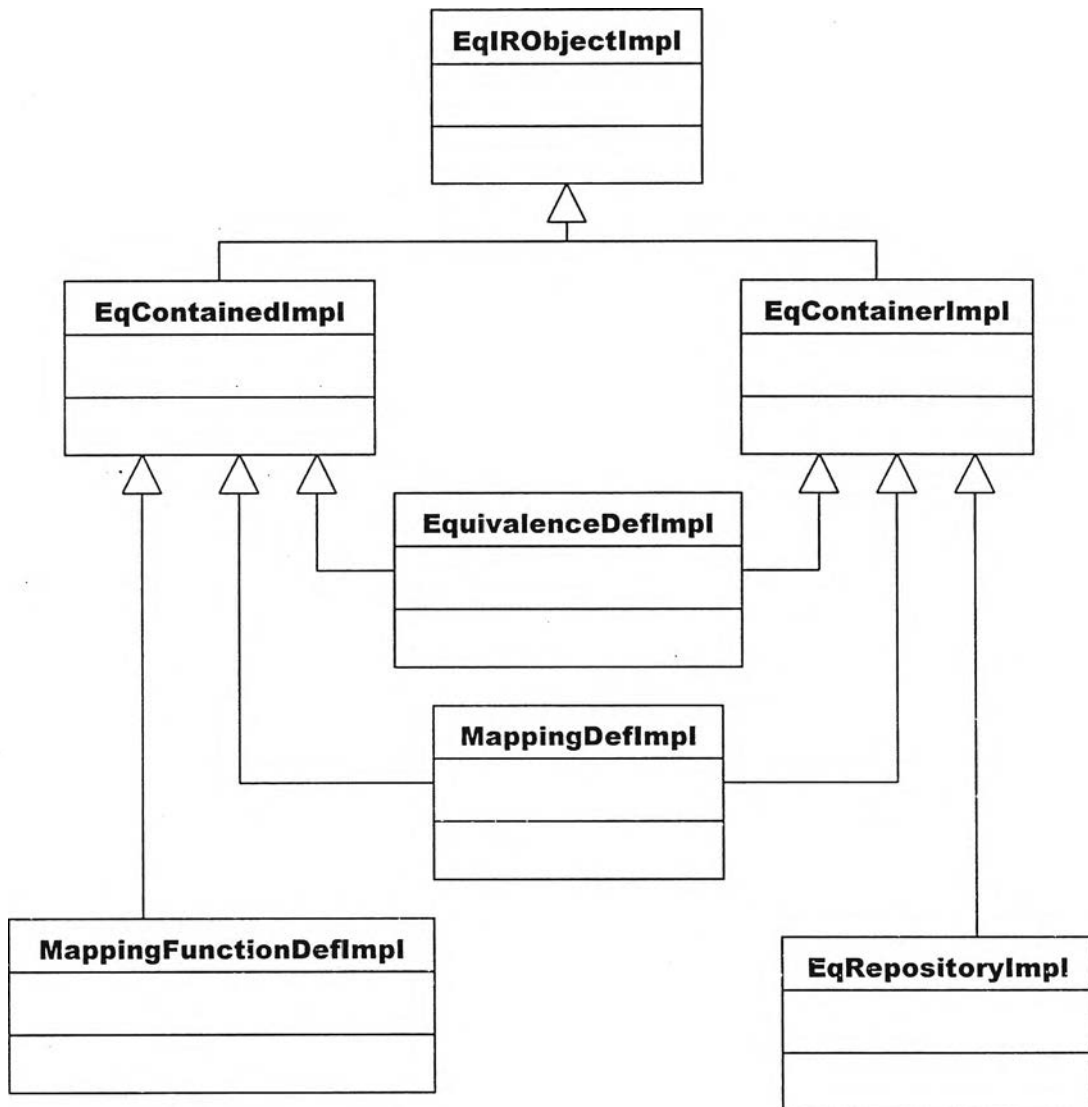
#### 4.2.2 แพคเกจ EquivalenceIR

แพคเกจนี้เปรียบเสมือนจุดเข้าของการเริ่มต้นการทำงานของโปรแกรมต้นแบบ โดยภายในแพคเกจนี้จะมีแพคเกจย่อยบรรจุอยู่อีก 5 แพคเกจดังจะได้อธิบายในหัวข้อ 4.2.3-4.2.7

#### 4.2.3 แพคเกจ core

แพคเกจนี้เป็นแพคเกจหลักของต้นแบบ เป็นแพคเกจที่มีคลาสที่ทำหน้าที่หลักๆ ตามนิยามของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน เราสามารถแบ่งคลาสในแพคเกจนี้ออกเป็น 2 กลุ่มคือ

- 1) คลาสที่ทำหน้าที่จัดการกับการทำงานพื้นฐานคือ
  - คลาส ConnectDb ซึ่งทำหน้าที่เริ่มต้นติดต่อกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่ใช้ในต้นแบบโดยใช้เจตีสี่ ซึ่งจะส่งผลให้การเปลี่ยนแปลงระบบจัดการฐานข้อมูลเชิงสัมพันธ์สามารถทำได้อย่างสะดวกและไม่ส่งผลกระทบต่อรหัสต้นฉบับที่ทำงานกับฐานข้อมูล
  - คลาส BasicService เป็นเสมือนคลาสตัวกลางที่คอยช่วยเหลือในการเรียกใช้บริการพื้นฐานต่างๆจากคลาสพื้นฐาน 4 คลาส คือ EqIRObject, EqContained, EqContainer และ EqRepository โดยทำการรับคำร้องขอจากคลาสอื่นที่ต้องการใช้บริการของคลาสพื้นฐานทั้งสิ้น แล้วทำการแจกจ่ายงานต่อไปยังอินสแตนซ์ของคลาสที่ให้บริการตรงกับที่ต้องการ
- 2) คลาสที่ทำหน้าที่พื้นฐานตามนิยามของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน คลาสเหล่านี้ได้รับการอธิบายไปแล้วในบทที่ 3 ซึ่งมีทั้งสิ้น 7 คลาส ความสัมพันธ์ของคลาสสามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 แผนผังคลาสของคลาสที่ทำหน้าที่พื้นฐานตามนิยามของ  
คลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันในแพคเกจ core

รายละเอียดของคลาสที่ทำหน้าที่พื้นฐานตามนิยามของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันมีดังต่อไปนี้

- 4.2.3.1 คลาส EqIRObjImpl
- 4.2.3.2 คลาส EqContainedImpl
- 4.2.3.3 คลาส EqContainerImpl
- 4.2.3.4 คลาส EqRepositoryImpl
- 4.2.3.5 คลาส EquivalenceDefImpl
- 4.2.3.6 คลาส MappingDefImpl
- 4.2.3.7 คลาส MappingFunctionDefImpl

4.2.3.1 คลาส `EqIRObjImpl` เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน `EqIRObj` (ดังหัวข้อ 3.2.2)

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
conn	Connection	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
stmt	Statement	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
rs	ResultSet	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ

เมธอด	รายละเอียด
<code>void _destroy( String repositoryID )</code>	ลบข้อมูลความสัมพันธ์แบบเท่าเทียมกัน
<code>void destroyEquivalenceDefTable( String repld )</code>	เมธอดช่วยเหลือในการลบข้อมูลออกจากตาราง <code>EquivalenceDef</code>
<code>void destroyMappingDefTable( String repld )</code>	เมธอดช่วยเหลือในการลบข้อมูลออกจากตาราง <code>MappingDef</code>

4.2.3.2 คลาส `EqContainedImpl` เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน `EqContained` (ดังหัวข้อ 3.2.3) ซึ่งคลาสนี้จะจัดการเกี่ยวกับข้อมูลพื้นฐานทั้งหมดของทุกส่วนต่อประสาน

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
conn	Connection	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
stmt	Statement	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
Rs	ResultSet	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ
eqRepository	EqRepository	ตัวแปรที่มีชนิดเป็นวัตถุของคลาส <code>EqRepository</code>



เมธอด	รายละเอียด
String convertRepIDtoAbsoluteName( String repID )	แปลงจาก RepositoryId ไปเป็น absolute name
void _setID( String oldRepID, String newRepID )	กำหนดค่า RepositoryId ของส่วนต่อประสาน
String _getName( String repID )	ค้นคืนชื่อของส่วนต่อประสาน
void _setName( String name, String repID )	กำหนดชื่อของส่วนต่อประสาน
String _getVersion( String repID )	ค้นคืนรุ่นของส่วนต่อประสาน
void _setVersion( String version, String repID )	กำหนดรุ่นของส่วนต่อประสาน
org.omg.CORBA.Object _getDefined_in( String repID )	ค้นคืนวัตถุที่บรรจุส่วนต่อประสาน
void _setDefined_in( String defined_in, String repID )	กำหนดวัตถุที่บรรจุส่วนต่อประสาน
String _getAbsolute_name( String repID )	ค้นคืนชื่อสัมบูรณ์ของส่วนต่อประสาน
void _setAbsolute_name( String repID )	กำหนดชื่อสัมบูรณ์ของส่วนต่อประสาน
EqRepository _getContaining_repository()	ค้นคืนวัตถุชนิด EqRepository ที่บรรจุส่วนต่อประสาน

4.2.3.3 คลาส EqContainerImpl เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน EqContainer (ดังหัวข้อ 3.2.4) คลาสนี้สามารถสร้างวัตถุของคลาส EquivalenceDefImpl ได้

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
conn	Connection	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
stmt	Statement	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
rs	ResultSet	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ

เมธอด	รายละเอียด
EqContained[] _eq_contents( String replID )	ค้นคืนรายการของวัตถุที่ถูกบรรจุ อยู่ใน โดยเมธอดนี้ใช้ได้กับ วัตถุที่มีคุณสมบัติเป็นตัวบรรจุ ชนิด EquivalenceDef และ MappingDef เท่านั้น
EquivalenceDef _create_equivalence( String id, String name, String version, org.omg.CORBA.InterfaceDef original_interface, org.omg.CORBA.InterfaceDef equivalence_interface )	สร้างวัตถุของคลาส EquivalenceDef

4.2.3.4 คลาส EqRepositoryImpl เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน  
EqRepository (ดังหัวข้อ 3.3.1)

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
eq_definition_kind	EquivalenceDefinitionKind	ตัวแปรที่ใช้ระบุชนิดของวัตถุซึ่งสำหรับวัตถุ นี้จะมีค่าเป็น eq_dk_Repository
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้ เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์ แบบเท่าเทียมกันทั้งหมดในต้นแบบ

เมธอด	รายละเอียด
EqContained eq_lookup_id( String search_id )	ค้นคืนวัตถุที่มี RepositoryId ตรงกับที่ระบุ
org.omg.CORBA.InterfaceDef[] list_of_equivalence_objs( String original_interface_id )	ค้นคืนรายการของวัตถุชนิด InterfaceDef ทั้งหมดที่มีความสัมพันธ์แบบเท่าเทียมกันกับส่วนต่อประสานที่ระบุ
EqContained[] eq_contents()	ค้นคืนรายการของวัตถุที่ถูกบรรจุอยู่ภายในโดยเมธอดในที่นี้ใช้ได้กับวัตถุที่มีคุณสมบัติเป็นตัวบรรจุชนิด EqRepository เท่านั้น
EquivalenceDef create_equivalence( String id, String name, String version, org.omg.CORBA.InterfaceDef original_interface, org.omg.CORBA.InterfaceDef equivalence_interface )	ไม่ได้ทำให้เกิดผลจริง แต่คงไว้เนื่องจาก การสืบทอดคุณลักษณะมาจากคลาส EqContainer
void destroy()	ลบข้อมูลความสัมพันธ์แบบเท่าเทียมกัน ซึ่งการลบข้อมูลที่วัตถุชนิด EqRepository เท่ากับเป็นการลบข้อมูลทั้งหมดภายใน EqRepository นี้ เพราะ EqRepository เป็นเสมือนจุดเข้าของต้นแบบ
EquivalenceDefinitionKind eq_def_kind()	ค้นคืนชนิดของวัตถุ ซึ่งในที่นี้จะคืนค่า eq_dk_Repository

4.2.3.5 คลาส `EquivalenceDefImpl` เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน `EquivalenceDef` (ดังหัวข้อ 3.3.2) คลาสนี้สามารถสร้างวัตถุของคลาส `MappingDefImpl` ได้

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
<code>conn</code>	<code>Connection</code>	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
<code>stmt</code>	<code>Statement</code>	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
<code>rs</code>	<code>ResultSet</code>	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
<code>basicService</code>	<code>BasicService</code>	ตัวแปรที่ใช้ในการเรียกใช้บริการพื้นฐาน
<code>eqTree</code>	<code>EqIRDefTree</code>	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้ในเป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ
<code>replID</code>	<code>String</code>	<code>RepositoryId</code> ของวัตถุ <code>EquivalenceDef</code> นี้
<code>name</code>	<code>String</code>	ชื่อของวัตถุ <code>EquivalenceDef</code> นี้
<code>version</code>	<code>String</code>	รุ่นของวัตถุ <code>EquivalenceDef</code> นี้
<code>defined_in</code>	<code>String</code>	<code>RepositoryId</code> ของวัตถุที่บรรจุวัตถุ <code>EquivalenceDef</code> นี้
<code>absolute_name</code>	<code>String</code>	ชื่อสมบูรณ์ของวัตถุ <code>EquivalenceDef</code> นี้
<code>containing_repository</code>	<code>EqRepository</code>	วัตถุที่เป็น <code>EqRepository</code> ที่บรรจุวัตถุ <code>EquivalenceDef</code> นี้
<code>eq_definition_kind</code>	<code>Equivalence DefinitionKind</code>	ตัวแปรที่ใช้ระบุชนิดของวัตถุซึ่งสำหรับวัตถุนี้จะมีค่าเป็น <code>eq_dk_EquivalenceDef</code>
<code>original_interface</code>	<code>InterfaceDef</code>	ส่วนต่อประสานต้นทาง
<code>equivalence_interface</code>	<code>InterfaceDef</code>	ส่วนต่อประสานปลายทางที่มีความสัมพันธ์แบบเท่าเทียมกันกับส่วนต่อประสานต้นทาง
<code>evolve_level</code>	<code>EvolveLevel</code>	ระดับของความสัมพันธ์แบบเท่าเทียมกันระหว่างส่วนต่อประสานต้นทางและปลายทาง

เมธอด	รายละเอียด
boolean equivalence_with( String interface_id )	ตรวจสอบว่าส่วนต่อประสานที่ระบุ คู่หนึ่งมีความสัมพันธ์แบบเท่า เทียมกันกับวัตถุ InterfaceDef ที่ บรรจุวัตถุ EquivalenceDef นี้หรือ ไม่
EquivalenceDescription describe_equivalence()	อธิบายวัตถุชนิด EquivalenceDef
EquivalenceIR.idl2java.MappingDef create_mapping( String id, String name, String version, EquivalenceSourceObjectPair eq_source_obj_pair, EquivalenceDestinationObjectPair[] eq_dest_obj_pair )	สร้างวัตถุชนิด MappingDef
void destroy()	ลบข้อมูลความสัมพันธ์แบบเท่า เทียมกัน
EqContained[] eq_contents ()	ค้นคืนรายการของวัตถุที่ถูกบรรจุ อยู่ในวัตถุ EquivalenceDef นี้ โดยรายการของวัตถุที่ได้จะมี ชนิดเป็น MappingDef
EquivalenceDef create_equivalence( String id, String name, String version, org.omg.CORBA.InterfaceDef original_interface, org.omg.CORBA.InterfaceDef equivalence_interface )	ไม่ได้ทำให้เกิดผลจริง แต่คงไว้ เนื่องจากการสืบทอดคุณลักษณะ มาจากคลาส EqContainer

เมธอด (ต่อ)	รายละเอียด
void original_interface( org.omg.CORBA.InterfaceDef original_interface )	กำหนดวัตถุของส่วนต่อประสาน original_interface
org.omg.CORBA.InterfaceDef original_interface()	คืนคืนวัตถุของส่วนต่อประสาน original_interface
void equivalence_interface( org.omg.CORBA.InterfaceDef equivalence_interface )	กำหนดวัตถุของส่วนต่อประสาน equivalence_interface
public org.omg.CORBA.InterfaceDef equivalence_interface()	คืนคืนวัตถุของส่วนต่อประสาน equivalence_interface
void evolve_level( EvolveLevel evolve_level )	กำหนดระดับของความสัมพันธ์แบบเท่า เทียมกัน
EvolveLevel evolve_level()	คืนคืนค่าระดับของความสัมพันธ์แบบเท่า เทียมกัน
void id( String id )	กำหนด RepositoryId ของวัตถุ EquivalenceDef นี้
String id()	คืนคืน RepositoryId ของวัตถุ EquivalenceDef นี้
void name( String name )	กำหนดชื่อของวัตถุ EquivalenceDef นี้
String name()	คืนคืนชื่อของวัตถุ EquivalenceDef นี้
void version( String version )	กำหนดรุ่นของวัตถุ EquivalenceDef นี้
String version()	คืนคืนรุ่นของวัตถุ EquivalenceDef นี้
void defined_in( String parentID )	กำหนดวัตถุที่บรรจุวัตถุ EquivalenceDef นี้
org.omg.CORBA.Object defined_in()	คืนคืนวัตถุที่บรรจุวัตถุ EquivalenceDef นี้
void absolute_name( String repID )	กำหนดชื่อสมบูรณ์ของวัตถุ EquivalenceDef นี้
String absolute_name()	คืนคืนชื่อสมบูรณ์ของวัตถุ EquivalenceDef นี้
EqRepository containing_repository()	คืนคืนวัตถุชนิด EqRepository ที่บรรจุวัตถุ EquivalenceDef นี้

เมธอด (ต่อ)	รายละเอียด
EquivalenceDefinitionKind eq_def_kind()	คืนชนิดชนิดของวัตถุ ซึ่งในที่นี้จะคืนค่า eq_dk_EquivalenceDef
void initEquivalenceDefVal( String id, String name, String version, org.omg.CORBA.InterfaceDef original_interface, org.omg.CORBA.InterfaceDef equivalence_interface )	เมธอดช่วยเหลือในการกำหนดค่าเริ่มต้นต่างๆ ของวัตถุชนิด EquivalenceDef นี้

4.2.3.6 คลาส MappingDefImpl เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน MappingDef (ดังหัวข้อ 3.3.3) คลาสนี้สามารถสร้างวัตถุของคลาส MappingFunctionDefImpl ได้

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
conn	Connection	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
stmt	Statement	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
rs	ResultSet	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
basicService	BasicService	ตัวแปรที่ใช้ในการเรียกใช้บริการพื้นฐาน
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ
repID	String	RepositoryId ของวัตถุ MappingDef นี้
name	String	ชื่อของวัตถุ MappingDef นี้
version	String	รุ่นของวัตถุ MappingDef นี้
defined_in	String	RepositoryId ของวัตถุที่บรรจุวัตถุ MappingDef นี้
absolute_name	String	ชื่อสมบูรณ์ของวัตถุ MappingDef นี้
containing_repository	EqRepository	วัตถุที่เป็น EqRepository ของวัตถุ MappingDef นี้

ลักษณะประจำ (ต่อ)	ชนิดข้อมูล	รายละเอียด
eq_definition_kind	EquivalenceDefinitionKind	ตัวแปรที่ใช้ระบุชนิดของวัตถุซึ่ง สำหรับวัตถุจะมีค่าเป็น eq_dk_MappingDef
eq_source_obj_pair	Equivalence SourceObjectPair	ตัวแปรวัตถุชนิด EquivalenceSourceObjectPair
eq_dest_obj_pair	Equivalence DestinationObjectPair[]	รายการตัวแปรวัตถุชนิด EquivalenceDestinationObjectPair
mapping_ref	String	IOR ของตัวดำเนินการแปลง
eqDefRepID	String	ตัวแปรช่วยเหลือในการสร้าง RepositoryId ที่เป็นคีย์นอก

เมธอด	รายละเอียด
MappingDescription describe_mapping()	อธิบายวัตถุชนิด MappingDef
MappingFunctionDef create_mapping_function( String id, String name, String version, org.omg.CORBA.TypeCode result, org.omg.CORBA.OperationMode mode, org.omg.CORBA.ParameterDescription[] params, org.omg.CORBA.ExceptionDef[] exceptions, GenMode gen_mode, MapMode map_mode )	สร้างวัตถุของคลาส MappingFunctionDef
void destroy()	ลบข้อมูลความสัมพันธ์แบบเท่า เทียมกัน



เมธอด (ต่อ)	รายละเอียด
EqContained[] eq_contents ()	คืนคืนรายการของวัตถุที่ถูกบรรจุอยู่ในวัตถุ MappingDef นี้ โดยรายการของวัตถุที่ได้จะมีชนิดเป็น MappingFunctionDef
EquivalenceDef create_equivalence( String id, String name, String version, org.omg.CORBA.InterfaceDef original_interface, org.omg.CORBA.InterfaceDef equivalence_interface )	ไม่ได้ทำให้เกิดผลจริง แต่คงไว้เนื่องจากการสืบทอดคุณลักษณะมาจากคลาส EqContainer
void eq_source_obj_pair( EquivalenceSourceObjectPair eq_source_obj_pair )	กำหนดวัตถุของส่วนต่อประสาน EquivalenceSourceObjectPair
EquivalenceSourceObjectPair eq_source_obj_pair()	คืนคืนวัตถุของส่วนต่อประสาน EquivalenceSourceObjectPair
void eq_dest_obj_pair( EquivalenceDestinationObjectPair[] eq_dest_obj_pair )	กำหนดวัตถุของส่วนต่อประสาน Equivalence DestinationObjectPair
EquivalenceDestinationObjectPair[] eq_dest_obj_pair()	คืนคืนวัตถุของส่วนต่อประสาน Equivalence DestinationObjectPair
void mapping_ref( String mapping_ref )	กำหนด IOR ของตัวดำเนินการแปลง
String mapping_ref()	คืนคืน IOR ของตัวดำเนินการแปลง
void id( String id )	กำหนด RepositoryId ของวัตถุ MappingDef นี้
String id()	คืนคืน RepositoryId ของวัตถุ MappingDef นี้
void name( String name )	กำหนดชื่อของวัตถุ MappingDef นี้
String name()	คืนคืนชื่อของวัตถุ MappingDef นี้
void version( String version )	กำหนดรุ่นของวัตถุ MappingDef นี้
String version()	คืนคืนรุ่นของวัตถุ MappingDef นี้

เมธอด (ชื่อ)	รายละเอียด
void defined_in( String parentID )	กำหนดวัตถุที่บรรจุวัตถุ MappingDef นี้
org.omg.CORBA.Object defined_in()	ค้นคืนวัตถุที่บรรจุวัตถุ MappingDef นี้
void absolute_name( String repID )	กำหนดชื่อสมบูรณ์ของวัตถุ MappingDef นี้
String absolute_name()	ค้นคืนชื่อสมบูรณ์ของวัตถุ MappingDef นี้
EqRepository containing_repository()	ค้นคืนวัตถุชนิด EqRepository ที่บรรจุวัตถุ MappingDef นี้
EquivalenceDefinitionKind eq_def_kind()	ค้นคืนชนิดของวัตถุ ซึ่งในที่นี้จะคืนค่า eq_dk_MappingDef
void initMappingDefVal( String id, String name, String version, String parentID, EquivalenceSourceObjectPair eq_source_obj_pair, EquivalenceDestinationObjectPair[] eq_dest_obj_pair )	เมธอดช่วยเหลือในการกำหนดค่าเริ่มต้นต่างๆ ของวัตถุชนิด MappingDef นี้
void createFK( String fkString )	เมธอดช่วยเหลือในการสร้างค่าคีย์นอก

4.2.3.7 คลาส MappingFunctionDefImpl เป็นคลาสที่ทำให้เกิดผลตามนิยามในส่วนของส่วนต่อประสาน MappingFunctionDef (ดังหัวข้อ 3.3.4)

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
conn	Connection	ตัวแปรที่ใช้ในการติดต่อกับฐานข้อมูล
stmt	Statement	ตัวแปรที่ใช้ในการจัดการกับฐานข้อมูล
rs	ResultSet	ตัวแปรที่ใช้ในการจัดการกับข้อมูลที่ได้จากฐานข้อมูล
basicService	BasicService	ตัวแปรที่ใช้ในการเรียกใช้บริการพื้นฐาน
eqTree	EqIRDefTree	ตัวแปรที่มีโครงสร้างข้อมูลแบบต้นไม้ที่ใช้เป็นสื่อกลางระหว่างข้อมูลความสัมพันธ์แบบเท่าเทียมกันทั้งหมดในต้นแบบ

ลักษณะประจำ (ต่อ)	ชนิดข้อมูล	รายละเอียด
repID	String	RepositoryId ของวัตถุ MappingFunctionDef นี้
name	String	ชื่อของวัตถุ MappingFunctionDef นี้
version	String	รุ่นของวัตถุ MappingFunctionDef นี้
defined_in	String	RepositoryId ของวัตถุที่บรรจุวัตถุ MappingFunctionDef นี้
absolute_name	String	ชื่อสมบูรณ์ของวัตถุ MappingFunctionDef นี้
containing_repository	EqRepository	วัตถุที่เป็น EqRepository ของวัตถุ MappingFunctionDef นี้
eq_definition_kind	Equivalence DefinitionKind	ตัวแปรที่ใช้ระบุชนิดของวัตถุซึ่งสำหรับวัตถุ นี้จะมีค่าเป็น eq_dk_MappingFunctionDef
result	org.omg.CORBA. TypeCode	ตัวแปรที่ใช้ระบุค่าที่ส่งคืนกลับสำหรับ MappingFunction นี้
params	org.omg.CORBA. ParameterDescription[]	ตัวแปรที่ใช้ระบุรายการของพารามิเตอร์ สำหรับ MappingFunction นี้
mode	org.omg.CORBA. OperationMode	ตัวแปรที่ใช้ระบุโหมดของการดำเนินการ ของ MappingFunction นี้
exceptions	org.omg.CORBA. ExceptionDef[]	ตัวแปรที่ใช้ระบุรายการของข้อยกเว้น สำหรับ MappingFunction นี้
map_mode	MapMode	ตัวแปรที่ใช้ระบุโหมดของการแปลงสำหรับ MappingFunction นี้
mapFnBody	String	ตัวแปรที่เก็บตำแหน่งที่อยู่ของแฟ้ม MappingFunction
mappingDefRepID	String	ตัวแปรช่วยเหลือในการสร้าง RepositoryId ที่เป็นคีย์นอก

เมธอด	รายละเอียด
MappingFunctionDescription describe_mapping_function()	อธิบายวัตถุชนิด MappingFunctionDef
void destroy()	ลบข้อมูลความสัมพันธ์แบบเท่าเทียมกัน แต่ในต้นแบบนี้ไม่ได้ทำให้เกิดให้ผลจริง เนื่องจากต้นแบบไม่ให้ลบข้อมูลที่ระดับ MappingFunction
void params( org.omg.CORBA. ParameterDescription[] params )	กำหนดรายการพารามิเตอร์สำหรับฟังก์ชันการแปลง
org.omg.CORBA. ParameterDescription[] params()	ค้นคืนรายการพารามิเตอร์ของฟังก์ชันการแปลง
void mode( org.omg.CORBA.OperationMode mode )	กำหนดโหมดของการดำเนินการสำหรับฟังก์ชันการแปลง
org.omg.CORBA.OperationMode mode()	ค้นคืนโหมดของการดำเนินการสำหรับฟังก์ชันการแปลง
void exceptions( org.omg.CORBA.ExceptionDef[] exceptions )	กำหนดรายการข้อยกเว้นสำหรับฟังก์ชันการแปลง
org.omg.CORBA.ExceptionDef[] exceptions()	ค้นคืนรายการข้อยกเว้นสำหรับฟังก์ชันการแปลง
void gen_mode( GenMode gen_mode )	กำหนดโหมดการสร้างฟังก์ชันการแปลง ซึ่งในต้นแบบนี้ไม่ได้ทำให้เกิดผลจริง เนื่องจากต้นแบบไม่ทำการสร้างตัวดำเนินการแปลง
GenMode gen_mode()	ค้นคืนโหมดการสร้างฟังก์ชันการแปลง ซึ่งไม่ได้ทำให้เกิดผลจริง แต่คงไว้เพื่อความสมบูรณ์
void map_mode( MapMode map_mode )	กำหนดโหมดการแปลงของฟังก์ชันการแปลง
MapMode map_mode()	ค้นคืนโหมดการแปลงของฟังก์ชันการแปลง
void mapFnBody( String mapFnBody )	กำหนดเพิ่มฟังก์ชันการแปลง
String mapFnBody()	ค้นคืนเพิ่มฟังก์ชันการแปลง

เมธอด (ชื่อ)	รายละเอียด
void result( org.omg.CORBA.TypeCode result )	กำหนดชนิดของค่าส่งกลับของฟังก์ชันการแปลง
org.omg.CORBA.TypeCode result()	คืนคืนชนิดของค่าส่งกลับของฟังก์ชันการแปลง
void id( String id )	กำหนด RepositoryId ของวัตถุ MappingFunctionDef นี้
String id()	คืนคืน RepositoryId ของวัตถุ MappingFunctionDef นี้
void name( String name )	กำหนดชื่อของวัตถุ MappingFunctionDef นี้
String name()	คืนคืนชื่อของวัตถุ MappingFunctionDef นี้
void version( String version )	กำหนดรุ่นของวัตถุ MappingFunctionDef นี้
String version()	คืนคืนรุ่นของวัตถุ MappingFunctionDef นี้
void defined_in( String parentID )	กำหนดวัตถุที่บรรจุวัตถุ MappingFunctionDef นี้
org.omg.CORBA.Object defined_in()	คืนคืนวัตถุที่บรรจุวัตถุ MappingFunctionDef นี้
void absolute_name( String repID )	กำหนดชื่อสมบูรณ์ของวัตถุ MappingFunctionDef นี้
String absolute_name()	คืนคืนชื่อสมบูรณ์ของวัตถุ MappingFunctionDef นี้
EqRepository containing_repository()	คืนคืนวัตถุชนิด EqRepository ที่บรรจุวัตถุ MappingFunctionDef นี้
EquivalenceDefinitionKind eq_def_kind()	คืนคืนชนิดของวัตถุ ซึ่งในที่นี้จะคืนค่า eq_dk_MappingFunctionDef

เมธอด (ต่อ)	รายละเอียด
<pre>public void nitMappingFunctionDefVal( String id, String name, String version, String parentID, org.omg.CORBA.TypeCode result, org.omg.CORBA.OperationMode mode, org.omg.CORBA.ParameterDescription[] params, org.omg.CORBA.ExceptionDef[] exceptions, MapMode map_mode )</pre>	เมธอดช่วยเหลือในการกำหนดค่าเริ่มต้นต่างๆของวัตถุชนิด MappingFunctionDef นี้
<pre>void createFK( String fkString )</pre>	เมธอดช่วยเหลือในการสร้างค่าคีย์นอก
<pre>Object clone()</pre>	เมธอดช่วยเหลือในการทำสำเนาวัตถุฟังก์ชันการแปลง ซึ่งจะใช้ในกรณีการนำกลับมาใช้ใหม่

#### 4.2.4 แพคเกจ gui

แพคเกจนี้เป็นแพคเกจที่รวบรวมคลาสที่ทำหน้าที่เกี่ยวกับการจัดการส่วนต่อประสานสำหรับผู้ใช้ทั้งหมด โดยมีคลาสทั้งสิ้น 4 คลาสหลักคือ

1. EqIRFrame คลาสหลักในการจัดการส่วนต่อประสานสำหรับผู้ใช้
2. EqIRTree คลาสในการจัดการกับการแสดงต้นไม้ของความสัมพันธ์แบบเท่าเทียมกันบนส่วนต่อประสานสำหรับผู้ใช้
3. IconData คลาสที่ช่วยการทำงานของคลาส EqIRTree
4. VerticalLayout คลาสในการจัดการกับตำแหน่งบางตำแหน่งขององค์ประกอบต่างๆบนส่วนต่อประสานสำหรับผู้ใช้ที่ตัวจัดการเลย์เอาต์ (Layout Manager) ของจาวาไม่สามารถรองรับได้

โดยคลาส EqIRFrame จะเป็นคลาสหลักที่ทำการเรียกใช้คลาสอื่นๆที่เหลือ และภายในคลาส EqIRFrame เองก็มีคลาสภายใน (inner class) อีกหลายคลาส ส่งผลให้คลาส EqIRFrame มีขนาดค่อนข้างใหญ่ การที่ออกแบบให้ใช้คลาสภายในก็เพื่อให้การทำงานของคลาส EqIRFrame ได้รับความสะดวก โดยสามารถเรียกใช้คลาสภายในของตัวเองได้เพื่อทำงานหลายๆอย่าง นอก

จากนั้นภายในแพ็คเกจ gui ยังมีแพ็คเกจ img บรรจุอยู่ ซึ่งเป็นแพ็คเกจที่จัดเก็บรูปภาพสำหรับการแสดงผลบนส่วนต่อประสานสำหรับผู้ใช้

#### 4.2.5 แพ็คเกจ **idl2java**

แพ็คเกจนี้เป็นแพ็คเกจที่ได้มาจากการเรียกใช้คำสั่ง `idl2java` ของวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 เพื่อสร้างแฟ้มรหัสต้นฉบับของคลาสต่างๆ โดยคลาสที่อยู่ในแพ็คเกจจะเป็นคลาสที่จัดการการติดต่อในรูปแบบของคอร์บา

#### 4.2.6 แพ็คเกจ **tree**

แพ็คเกจนี้เป็นแพ็คเกจที่รวบรวมคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลทั้งหมดของต้นแบบนี้ อันได้แก่

- 4.2.6.1 `IfDefNode` เป็นคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลที่เป็นบัพ (node) ของวัตถุชนิด `InterfaceDef` สำหรับคลาส `IfDefList`
- 4.2.6.2 `IfDefList` เป็นคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลแบบรายการโยง (linked list) ที่จัดการกับวัตถุชนิด `InterfaceDef`
- 4.2.6.3 `EqIRDefTreeNode` เป็นคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลที่เป็นบัพ (node) ของวัตถุชนิด `EqRepository`, `EquivalenceDef`, `MappingDef` และ `MappingFunctionDef` สำหรับคลาส `EqIRDefTree`
- 4.2.6.4 `EqIRDefTree` เป็นคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลแบบต้นไม้ (tree) ที่จัดการกับวัตถุชนิด `EqRepository`, `EquivalenceDef`, `MappingDef` และ `MappingFunctionDef`

รายละเอียดของคลาสที่ทำหน้าที่เป็นโครงสร้างข้อมูลทั้งหมดของต้นแบบของคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกันมีดังต่อไปนี้

## 4.2.6.1 คลาส IfDefNode

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
ifDefnodeName	String	ชื่อของบัพที่ใช้ในการแสดงผลบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
order	int	ลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
ifDefObj	org.omg.CORBA.InterfaceDef	วัตถุชนิด InterfaceDef ของคลังส่วนต่อประสานของคอรีบา
next	IfDefNode	วัตถุตัวชี้ที่ชี้ไปยังบัพถัดไปในรายการโยงของคลาส IfDefList
evolved	boolean	ค่าที่ไว้ระบุว่าวัตถุชนิด InterfaceDef ในบัพปัจจุบันมีการเปลี่ยนแปลงชนิดของบริการหรือไม่

เมธอด	รายละเอียด
String getIfDefnodeName()	ค้นคืนชื่อของบัพที่ใช้ในการแสดงผลบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
void setOrder( int value )	กำหนดค่าลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
int getOrder()	ค้นคืนค่าลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
org.omg.CORBA.InterfaceDef getIfDefObj()	ค้นคืนวัตถุชนิด InterfaceDef ภายในบัพ
Boolean isEvolved()	ตรวจสอบว่าวัตถุชนิด InterfaceDef ในบัพปัจจุบันมีการเปลี่ยนแปลงชนิดของบริการหรือไม่
IfDefNode getNext()	ค้นคืนวัตถุตัวชี้ที่ชี้ไปยังบัพถัดไปในรายการโยงของคลาส IfDefList



## 4.2.6.2 คลาส IfDefList

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
headNode	IfDefNode	วัตถุที่เป็นบัพของคลาส IfDefNode ที่เป็นตัวชี้ที่ตำแหน่งเริ่มต้นของรายการโยงในคลาส IfDefList
lastNode	IfDefNode	วัตถุที่เป็นบัพของคลาส IfDefNode ที่เป็นตัวชี้ที่ตำแหน่งสุดท้ายของรายการโยงในคลาส IfDefList

เมธอด	รายละเอียด
synchronized void addNode( String tname, org.omg.CORBA.InterfaceDef obj, boolean ev )	เพิ่มบัพเข้าไปในรายการโยง
synchronized boolean isEmptyList()	ตรวจสอบว่ารายการโยงว่างเปล่าหรือไม่
IfDefNode getHeadNode()	คืนคืนวัตถุที่เป็นบัพของคลาส IfDefNode ที่อยู่ที่ตำแหน่งเริ่มต้นของรายการโยงในคลาส IfDefList
org.omg.CORBA.InterfaceDef find( int orderNumber )	คืนคืนวัตถุชนิด InterfaceDef ในรายการโยงตามลำดับที่ระบุเข้ามา

## 4.2.6.3 คลาส EqIRDefTreeNode

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
nodeName	String	ชื่อของบัพที่ใช้ในการแสดงผลบนต้นไม้ของส่วนต่อประสานสำหรับผู้ไ้
order	int	ลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ไ้
eqObj	EqContained	วัตถุชนิด EqContained ของต้นแบบนี้
parent	org.omg.CORBA. Object	วัตถุใดๆที่เป็นตัวบรรจวัตถุในบัพปัจจุบัน
next	EqIRDefTreeNode	วัตถุชนิด EqIRDefTreeNode ที่เป็นบัพพี่น้อง (sibling node) ในต้นไม้ของคลาส EqIRDefTree
child	EqIRDefTreeNode	วัตถุชนิด EqIRDefTreeNode ที่เป็นบัพลูก (child node) ในต้นไม้ของคลาส EqIRDefTree

เมธอด	รายละเอียด
String getNodeName()	คืนคืนชื่อของบัพที่ใช้ในการแสดงผลบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
void setOrder( int value )	กำหนดค่าลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
int getOrder()	คืนคืนค่าลำดับของบัพบนต้นไม้ของส่วนต่อประสานสำหรับผู้ใช้
EqContained getEqObj()	คืนคืนวัตถุชนิด EqContained ภายในบัพ
org.omg.CORBA.InterfaceDef getParentIFDef()	คืนคืนวัตถุ ที่เป็นตัวบรรจุบัพปัจจุบัน ที่มีชนิดเป็น InterfaceDef
EqContained getParent()	คืนคืนวัตถุ ที่เป็นตัวบรรจุบัพปัจจุบัน ที่มีชนิดเป็น EqContained
EqIRDefTreeNode getNext()	คืนคืนวัตถุที่เป็นบัพพี่น้องในต้นไม้ของคลาส EqIRDefTree
EqIRDefTreeNode getChild()	คืนคืนวัตถุที่เป็นบัพลูกในต้นไม้ของคลาส EqIRDefTree
Object clone()	เมธอดช่วยเหลือในการทำสำเนาวัตถุชนิด EqIRDefTreeNode

#### 4.2.6.4 คลาส EqIRDefTree

ลักษณะประจำ	ชนิดข้อมูล	รายละเอียด
headNodeOfheaderList	EqIRDefTreeNode	วัตถุที่เป็นบัพของคลาส EqIRDefTreeNode ที่อยู่ที่ตำแหน่งเริ่มต้นของต้นไม้ในคลาส EqIRDefTree
currentNodeOfChildList	EqIRDefTreeNode	วัตถุที่เป็นบัพของคลาส EqIRDefTreeNode ที่อยู่ที่ตำแหน่งของบัพปัจจุบันของต้นไม้ในคลาส EqIRDefTree

เมธอด	รายละเอียด
EqIRDefTreeNode getHeaderListHeadNode()	คืนคืนวัตถุที่เป็นบัพของคลาส EqIRDefTreeNode ที่เป็นตัวชี้ที่ตำแหน่งเริ่มต้นของต้นไม้ในคลาส EqIRDefTree
synchronized void addNode( String captionName, EqContained eqContainedObj, org.omg.CORBA.Object parentObj )	เพิ่มบัพเข้าไปในต้นไม้
Void addOrderNode( EqIRDefTreeNode addedNode )	เพิ่มบัพเข้าไปในต้นไม้แบบมีลำดับ
EqContained find( EquivalenceDefinitionKind eq_definition_kind, int orderNumber )	คืนคืนวัตถุที่เป็นบัพในต้นไม้ของคลาส EqIRDefTree โดยค้นหาตามชนิดของวัตถุ และลำดับที่รับเข้ามา
EqContained find( String search_id )	คืนคืนวัตถุที่เป็นบัพในต้นไม้ของคลาส EqIRDefTree โดยค้นหาตาม RepositoryId ที่รับเข้ามา
void destroy( String destroyRepld )	พิจารณาบัพของข้อมูลในโครงสร้างข้อมูลแบบต้นไม้ที่ต้องการลบ โดยเมธอดนี้จะทำการเรียกใช้เมธอด destroyHelper
synchronized void destroyHelper( String nodeRepld )	ทำการลบบัพของข้อมูลในโครงสร้างข้อมูลแบบต้นไม้
void showTreeContent()	แสดงข้อมูลทั้งหมดในต้นไม้

#### 4.2.7 แพคเกจ util

แพคเกจนี้เป็นแพคเกจที่รวบรวมคลาสที่คอยช่วยเหลืองานบางงานของต้นแบบนี้ อันได้แก่

1. CheckReplD คลาสช่วยเหลือในการตรวจสอบว่ามี RepositoryId อยู่หรือไม่
2. PrintIDL คลาสช่วยเหลือในการแสดงนิยามของส่วนต่อประสานที่อยู่ในคลังส่วนต่อประสานของคอร์บา

บทนี้ได้อธิบายถึงการออกแบบส่วนประกอบต่างๆที่จะนำมาทำให้เกิดผลเป็นต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน อันได้แก่ การออกแบบฐานข้อมูลที่จะจัดเก็บข้อมูลความสัมพันธ์แบบเท่าเทียมกัน และการออกแบบคลาสต่างๆที่ต้องทำให้เกิดผล ซึ่งจากรายละเอียดในบทนี้ทำให้เราสามารถพัฒนาต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกันได้ และในบทถัดไปจะเป็นการทดสอบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกันที่ได้พัฒนาขึ้นตามนิยามส่วนต่อประสานที่อธิบายในบทที่ 3 และการออกแบบส่วนประกอบต่างๆในบทนี้ โดยประเด็นหลักของการทดสอบคือความสามารถในการจัดการกับข้อมูลความสัมพันธ์แบบเท่าเทียมกันที่สอดคล้องกับการเปลี่ยนบริการรุ่นใหม่แบบอิสระทุกกรณี ดังจะแสดงในบทถัดไป