

รายการอ้างอิง

- [1] T. Senivongse. Evolution Transparency for Distributed Service Types. Ph.D. Thesis, University of Kent, UK., January 1997.
- [2] J. Indulska, M. Bearman, and K. Raymond. A Type Management System for an ODP Trader. In Proceedings of the Intl. Conf. On Open Distributed Processing (ICODP'93), Berlin: 1993.
- [3] B. Meyer et al. Enabling Internetworking between Heterogeneous Distributed Platforms. In Distributed Platforms, (n.p.): A. Schill et al (eds.), Chapman & Hall, 1996.
- [4] G. Coulouris, J. Dollimore, and T. Kindberg. Distributed Systems Concepts and Design. U.S.A: Addison Wesley Longman, 1994.
- [5] OMG. The Common Object Request Broker: Architecture and Specification. Revision 2.2 (n.p.):1998.
- [6] A. Pope. The CORBA Reference Guide: Understanding the Common Object Request Broker Architecture. U.S.A: Addison Wesley Longman, 1998.
- [7] R. Orfali, D. Harkey. Client/Server Programming with Java and CORBA. U.S.A: John Wiley and Sons, Inc. 1998.
- [8] P. E. Chung, et al. DCOM and CORBA Side by Side. Step by Step. and Layer by Layer. <http://www.cs.wustl.edu/~schmidt/submit/Paper.html>.
- [9] ISO/IEC. 10746-1 – ODP Reference Model Part 1: Overview. (n.p.): 1995.
- [10] J. Shirley, et al. Guide to Writing DCE Application. 2nd ed. (n.p.): O'Reilly & Associates, 1992.
- [11] ISO/IEC. Draft Rec. X.950-1 | ISO/IEC 13235-1 – ODP Trading Function : Specification. (n.p.): 1997.
- [12] ITU/ISO. Tutorial on ODP Trading Function. (n.p.): 1997.

ภาคผนวก

ภาคผนวก ก
คู่มือการใช้งานต้นแบบคลังชนิด
ของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

ภาคผนวกนี้จะแสดงหน้าจอต่างๆของส่วนต่อประสานสำหรับผู้ทำการเปลี่ยนรุ่นบริการซึ่งเป็นผู้ใช้ของโปรแกรมต้นแบบ และแสดงการใช้งานต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกันร่วมกับคลังส่วนต่อประสานของวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 บนระบบปฏิบัติการ Windows NT 4.0 เนื่องจากแนวคิดของวิทยานิพนธ์นี้ที่ต้องการรองรับการเปลี่ยนบริการรุ่นใหม่แบบอิสระและต้องการให้เป็นไปในแบบโปร่งใส นั่นคือระบบเดิมที่ยังไม่พร้อมในการเปลี่ยนบริการรุ่นใหม่จะยังคงสามารถทำงานต่อไปได้จนกว่าจะพร้อมในการเปลี่ยนบริการรุ่นใหม่ซึ่งผู้รับบริการยังคงเรียกใช้บริการรุ่นเก่าในรูปแบบเดิมต่อไปได้อีกระยะหนึ่งจนกว่าระบบจะพร้อมในการเปลี่ยนบริการรุ่นใหม่ ดังนั้นการดำเนินงานของโปรแกรมต้นแบบจะต้องทำงานร่วมกับระบบเดิมซึ่งได้ดำเนินงานกับคลังส่วนต่อประสานอยู่ก่อนหน้าแล้ว รายละเอียดของการใช้งานวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 สามารถอ้างอิงได้จาก [12]

ก.1. การสั่งดำเนินการโปรแกรมต้นแบบ

ก.1.1) สั่งดำเนินงาน osagent ดังนี้

```
prompt>start osagent -C
```

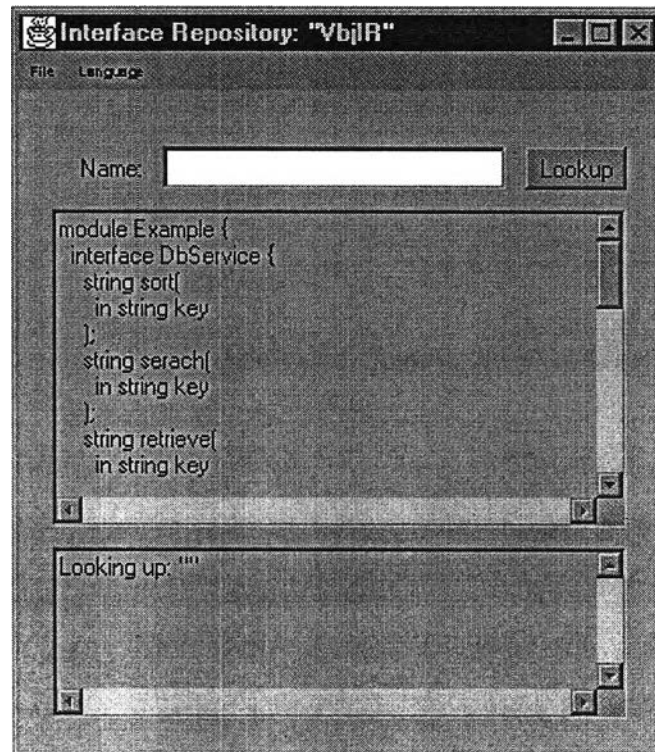
- osagent คือคำสั่งในการดำเนินงาน osagent ของวิสิโบริกเคอร์
- -C คือตัวเลือก (Option) ของคำสั่ง osagent ในการสั่งให้ทำงานในโหมดคอนโซล (Console Mode)

ก.1.2) สั่งดำเนินงานคลังส่วนต่อประสานของวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 ดังนี้

```
prompt>irep VbjIR ex.idl
```

- irep คือคำสั่งในการดำเนินงานคลังส่วนต่อประสานของวิสิโบริกเคอร์
- VbjIR คือชื่อที่ตั้งให้กับคลังส่วนต่อประสานที่จะดำเนินงาน
- ex.idl คือแฟ้มนิยามส่วนต่อประสานที่ป้อนเข้าสู่คลังส่วนต่อประสาน

ซึ่งจากการใช้คำสั่งดังกล่าว จะได้ผลลัพธ์เป็นส่วนต่อประสานสำหรับผู้ใช้ของคลังส่วนต่อประสานดังแสดงในรูป ก1 ซึ่งจะแสดงรายละเอียดของเพื่มนิยามส่วนต่อประสานที่ป้อนเข้าสู่คลังส่วนต่อประสานไว้ภายใน



รูป ก1 คลังส่วนต่อประสานของวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4

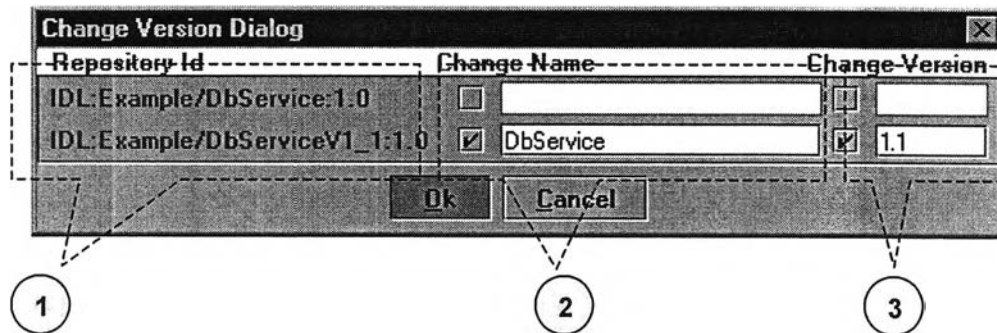
ก.1.3) สั่งดำเนินงานโปรแกรมต้นแบบ ดังนี้

```
prompt>vbj EqIR VbjIR
```

- vbj คือคำสั่งในการดำเนินงานโปรแกรมที่เขียนด้วยภาษาจาวาและมีการกำหนดค่าตัวแปรแวดล้อม (environment variable) ที่ต้องการสำหรับระบบที่จะดำเนินงานภายใต้วิสิโบริกเคอร์สำหรับจาวารุ่น 3.4
- EqIR คือชื่อโปรแกรมเริ่มต้นของโปรแกรมต้นแบบ
- VbjIR คือชื่อที่ตั้งให้กับคลังส่วนต่อประสานของวิสิโบริกเคอร์ที่ได้ดำเนินงานไว้แล้ว

ซึ่งจากการใช้คำสั่งดังกล่าว จะได้ผลลัพธ์เป็นส่วนต่อประสานสำหรับผู้ใช้ของต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน ดังแสดงในรูป ก3 แต่ด้วยปัญหาเรื่องการเปลี่ยนรุ่น (ดังที่กล่าวไปแล้วในหัวข้อ 5.1.1.1) ก่อนที่จะแสดงส่วนต่อประสานสำหรับผู้ใช้ของโปรแกรมต้นแบบ จะแสดงหน้าจอ "Change Version Dialog" เพื่อให้ผู้ทำการเปลี่ยนรุ่นสามารถ

เปลี่ยนรุ่นของส่วนต่อประสานซึ่งวิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 ไม่สนับสนุนได้ ดังแสดงในรูป ก 2



รูป ก2 หน้าจอ "Change Version Dialog" สำหรับการรองรับการเปลี่ยนรุ่น
ก่อนการแสดงผลส่วนต่อประสานสำหรับผู้ใช้ของโปรแกรมต้นแบบ

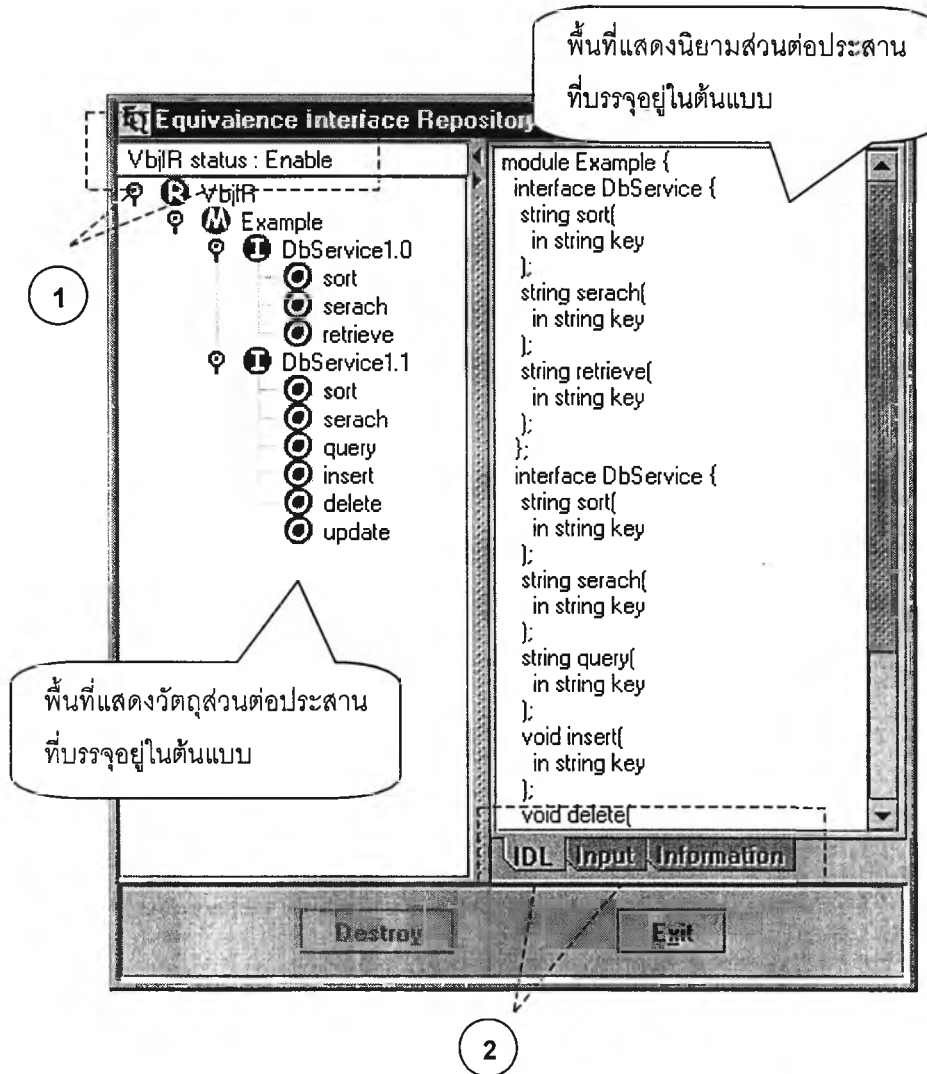
ก.2. การใช้งานหน้าจอ Change Version Dialog

ก.2.1) จากรูป ก2 บริเวณที่ 1 จะแสดง RepositoryId ทั้งหมดที่มีอยู่ในคลังส่วนต่อประสานของระบบ ซึ่งจากรูปจะมีส่วนต่อประสาน "IDL:Example/DbService:1.0" ซึ่งในที่นี้คือบริการรุ่นเก่า และ "IDL:Example/DbServiceV1_1:1.0" ซึ่งในที่นี้คือบริการรุ่นใหม่ โดยที่ตามแนวคิดในการเปลี่ยนรุ่นของบริการนั้นควรจะเป็น "IDL:Example/DbService:1.1" แต่จำเป็นต้องตั้งชื่อให้แตกต่างออกไปแบบชั่วคราว เนื่องจากปัญหาการเปลี่ยนรุ่นที่วิสิโบริกเคอร์สำหรับจาวารุ่น 3.4 ไม่รองรับ

ก.2.2) บริเวณที่ 2 เป็นส่วนที่ใช้ในการเปลี่ยนชื่อของบริการโดยจะสามารถทำการเปลี่ยนได้ก็ต่อเมื่อมีการเลือกกล่องเลือก (checkbox) หากไม่ได้เลือกไว้ ส่วนที่เป็นช่องเติมข้อความ (text field) จะปิดทาง (disable) หากเลือกกล่องเลือกที่ตรงกับ RepositoryId ไต ก็จะสามารถเปลี่ยนชื่อใหม่ให้กับส่วนต่อประสานของ RepositoryId นั้น

ก.2.3) บริเวณที่ 3 เป็นส่วนที่ใช้ในการเปลี่ยนรุ่นของบริการโดยจะสามารถทำการเปลี่ยนได้ก็ต่อเมื่อมีการเลือกกล่องเลือก (checkbox) หากไม่ได้เลือกไว้ ส่วนที่เป็นช่องเติมข้อความ (text field) จะปิดทาง (disable) หากเลือกกล่องเลือกที่ตรงกับ RepositoryId ไต ก็จะสามารถเปลี่ยนรุ่นใหม่ให้กับส่วนต่อประสานของ RepositoryId นั้น

ก.2.4) หากกดปุ่ม Ok ก็จะทำให้การดำเนินงานโปรแกรมต้นแบบต่อไป แล้วแสดงผลเป็นส่วนต่อประสานสำหรับผู้ใช้ ดังแสดงในรูป ก3 หากกดปุ่ม Cancel จะเป็นยกเลิกการดำเนินงานโปรแกรมต้นแบบ



รูป ก3 หน้าจอหลักของส่วนต่อประสานสำหรับผู้ใช้ของโปรแกรมต้นแบบ
คลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

ก.3. การใช้งานหน้าจอหลักของส่วนต่อประสานสำหรับผู้ใช้งานโปรแกรมต้นแบบ

หน้าจอหลักของส่วนต่อประสานสำหรับผู้ใช้งานโปรแกรมต้นแบบที่แสดงในรูป ก3 จะแบ่งพื้นที่ออกเป็น 3 ส่วนคือ

- พื้นที่ด้านซ้ายเป็นการแสดงโครงสร้างความสัมพันธ์ในลักษณะการบรรจุของวัตถุต่างๆที่อยู่ในคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน (ดูหัวข้อ ก.3.1)
- พื้นที่ด้านขวาเป็นพื้นที่สำหรับแสดงผลข้อมูลและส่วนสำหรับการป้อนข้อมูลความสัมพันธ์แบบเท่าเทียมกันเข้าสู่คลังส่วนต่อประสาน (ดูหัวข้อ ก.3.2)
- พื้นที่ปุ่มประมวลผลที่อยู่ด้านล่าง

ก.3.1) บริเวณที่ 1 จะแสดงสถานะการเปิดทาง (Enable) และปิดทางของต้นไม้โครงสร้างความสัมพันธ์ในลักษณะการบรรจุของวัตถุต่างๆที่อยู่ในคลังส่วนต่อประสานที่แสดงอยู่ด้านล่างบริเวณ 1 ในรูป ก3 โดยสัญรูปที่มีในต้นไม้ดังกล่าวมีทั้งสิ้น 11 สัญรูปคือ

1. (R) เป็นสัญรูปที่แสดงวัตถุชนิด Repository
2. (M) เป็นสัญรูปที่แสดงวัตถุชนิด ModuleDef
3. (I) เป็นสัญรูปที่แสดงวัตถุชนิด InterfaceDef
4. (A) เป็นสัญรูปที่แสดงวัตถุชนิด AttributeDef
5. (O) เป็นสัญรูปที่แสดงวัตถุชนิด OperationDef
6. (T) เป็นสัญรูปที่แสดงวัตถุชนิด TypedefDef
7. (C) เป็นสัญรูปที่แสดงวัตถุชนิด ConstantDef
8. (E) เป็นสัญรูปที่แสดงวัตถุชนิด ExceptionDef
9. (Eq) เป็นสัญรูปที่แสดงวัตถุชนิด EquivalenceDef
10. (Mp) เป็นสัญรูปที่แสดงวัตถุชนิด MappingDef
11. (Fn) เป็นสัญรูปที่แสดงวัตถุชนิด MappingFunctionDef

หากมีการเลือกบัพโศปหนึ่งทีแสดงถึงวัตถุชนิด InterfaceDef และ EquivalenceDef บัพต่างๆจะปิดทางจนกว่าจะมีการยกเลิกการทำงานหรือทำการป้อนข้อมูลเรียบร้อย (ดูหัวข้อ ก.3.2.2.1)

ก.3.2) ส่วนการแสดงผลทางด้านขวามีแบ่งออกเป็น 3 ส่วนย่อยโดยใช้แถบเลือก (tab) ดังนี้

- ส่วนแสดงนิยามส่วนต่อประสานทีบรรจุอยู่ในคลังส่วนต่อประสาน (IDL tab)

- ส่วนรับข้อมูลความสัมพันธ์แบบเท่าเทียมกันเข้าสู่คลังส่วนต่อประสาน (Input tab)
- ส่วนแสดงสารสนเทศของต้นแบบของคลังส่วนต่อประสาน (Information tab)

ก.3.2.1) ส่วนแสดงนิยามส่วนต่อประสานที่บรรจุอยู่ในคลังส่วนต่อประสาน (IDL tab)
แถบเลือกนี้จะแสดงนิยามส่วนต่อประสานที่อยู่ในคลังส่วนต่อประสาน โดยจะแสดงเป็นรูปแบบของข้อความภายในแฟ้มรหัสต้นทาง ดังรูป ก3

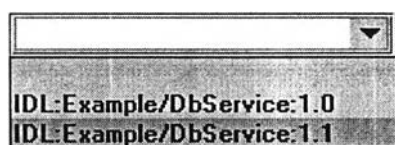
ก.3.2.2) ส่วนป้อนข้อมูลความสัมพันธ์แบบเท่าเทียมกันเข้าสู่คลังส่วนต่อประสาน (Input tab)

แถบเลือกนี้จะเป็นส่วนต่อประสานสำหรับผู้ใช้ในการป้อนข้อมูลความสัมพันธ์แบบเท่าเทียมกันเข้าสู่ต้นแบบ

ก.3.2.2.1) หน้าจอการป้อนข้อมูล EquivalenceDef เข้าสู่คลังส่วนต่อประสาน ดังแสดงในรูป ก4 มีขั้นตอนการทำงานดังนี้

รูป ก4 หน้าจอของส่วนต่อประสานสำหรับผู้ใช้ในการป้อนข้อมูล EquivalenceDef เข้าสู่โปรแกรมต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

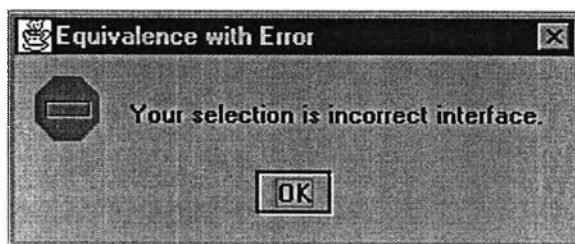
1. เลือกระดับของการแปลง
2. เลือกส่วนต่อประสานที่จะเป็น equivalence interface



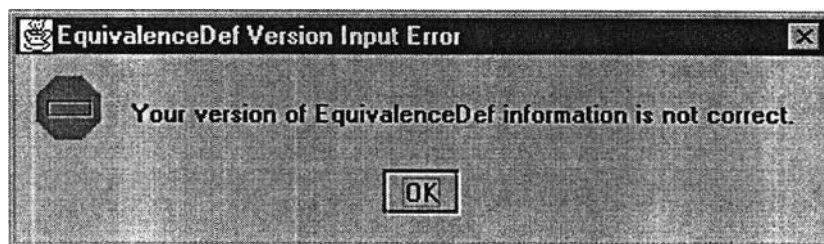
หากยังไม่ทำการเลือก ช่องเติมข้อความชื่อและรุ่นของวัตถุ EquivalenceDef จะปิดทาง

3. ป้อนชื่อวัตถุ EquivalenceDef
4. ป้อนรุ่นวัตถุ EquivalenceDef

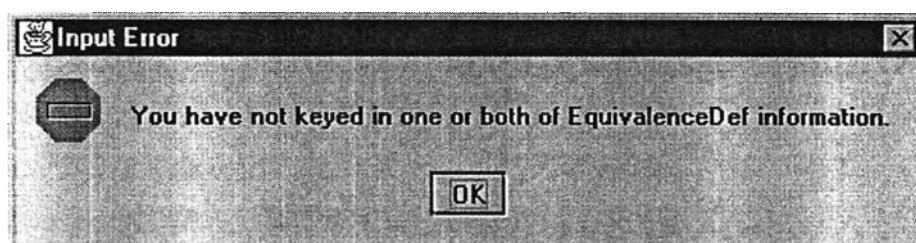
5. กดปุ่ม **Next** (ดูหัวข้อ ก.3.3) เพื่อแสดงหน้าจอการป้อนข้อมูล MappingDef เข้าสู่คลังส่วนต่อประสาน ดังแสดงในรูป ก8 ในการกดปุ่ม Next นี้ หากเกิดข้อผิดพลาดในการรับค่า จะมีหน้าจอแจ้งข้อผิดพลาดดังแสดงในรูป ก5 – ก7



รูป ก5 หน้าจอแสดงข้อผิดพลาดของกรณี que เลือกข้อมูล ส่วนต่อประสานซ้ำกับส่วนต่อประสานที่เป็น original interface



รูป ก6 หน้าจอแสดงข้อผิดพลาดของกรณี que ป้อนข้อมูลรุ่นของวัตถุ EquivalenceDef ไม่ถูกต้อง



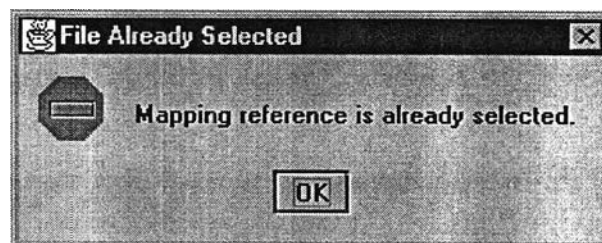
รูป ก7 หน้าจอแสดงข้อผิดพลาดของกรณี que ป้อนข้อมูลของวัตถุ EquivalenceDef ไม่ครบ

ก.3.2.2.2) หน้าจอการป้อนข้อมูล MappingDef เข้าสู่คลังส่วนต่อประสาน ดังแสดงในรูป ก8 มีขั้นตอนการทำงานดังนี้

รูป n8 หน้าจอของส่วนต่อประสานสำหรับผู้ใช้ในการป้อนข้อมูล MappingDef เข้าสู่โปรแกรมต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

1. ป้อนชื่อวัตถุ MappingDef
2. ป้อนรุ่นวัตถุ MappingDef
3. กด ปุ่ม **Browse...** เพื่อเลือกเพิ่มเก็บข้อมูล IOR เมื่อทำการเลือกไปแล้วปุ่มก็จะ

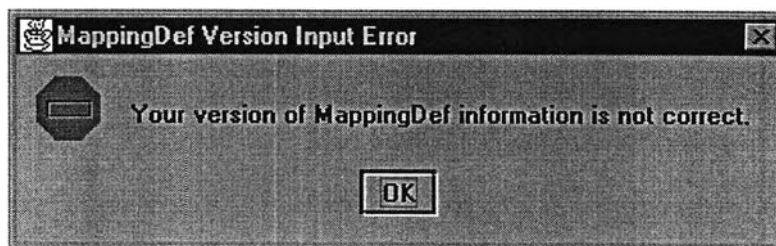
เปลี่ยนชื่อไปเป็นชื่อเพิ่มทำการเลือก และจะเลือกซ้ำไม่ได้ หากกดเลือกอีกก็จะแสดงหน้าจอรายงานความผิดพลาดดังนี้



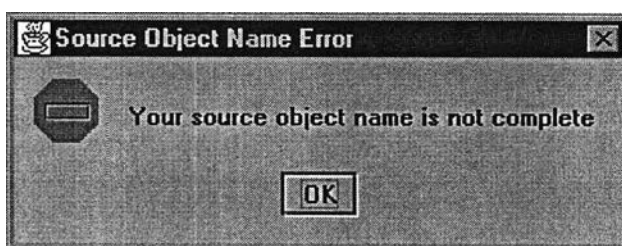
รูป n9 หน้าจอแสดงข้อผิดพลาดของกรณีที่จะเลือกเพิ่ม IOR อีกครั้ง

4. ป้อนข้อมูลเกี่ยวกับวัตถุส่วนต่อประสานต้นทาง
5. ป้อนข้อมูลเกี่ยวกับวัตถุส่วนต่อประสานปลายทาง สำหรับข้อมูลนี้สามารถมีได้มากกว่า 1 ชุด ขึ้นอยู่กับความต้องการ โดยการกดปุ่ม **Add Destination**
6. กดปุ่ม **Next** (ดู หัว ข้อ ก. 3.3) เพื่อแสดงหน้าจอการป้อนข้อมูล MappingFunctionDef เข้าสู่คลังส่วนต่อประสาน ดังแสดงในรูป n13 ในการกดปุ่ม Next นี้

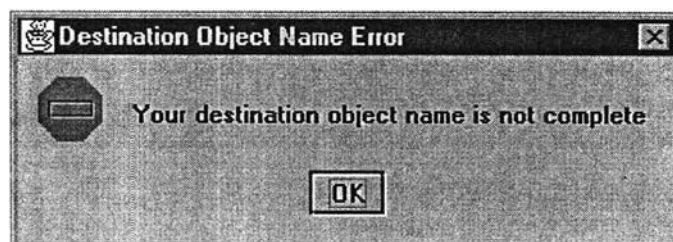
หากเกิดข้อผิดพลาดในการรับค่า จะมีหน้าจอแจ้งข้อผิดพลาดดังแสดงในรูป ก10 – ก12



รูป ก10 หน้าจอแสดงข้อผิดพลาดของกรณีที่ย้อนข้อมูลรุ่นของวัตถุ MappingDef ไม่ถูกต้อง



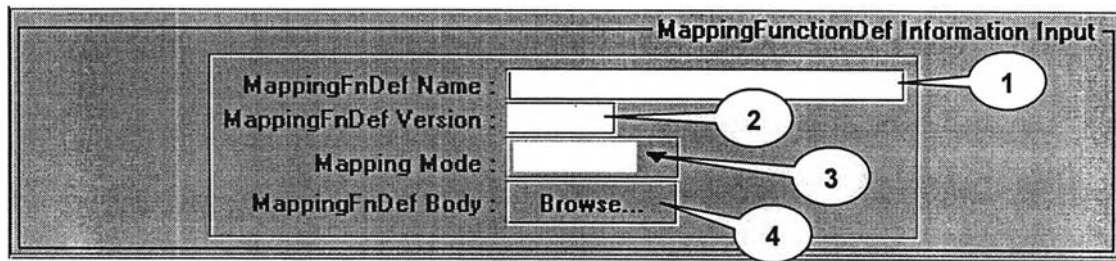
รูป ก11 หน้าจอแสดงข้อผิดพลาดของกรณีที่ยังไม่ป้อนชื่อวัตถุส่วนต่อประสานต้นทาง



รูป ก12 หน้าจอแสดงข้อผิดพลาดของกรณีที่ยังไม่ป้อนชื่อวัตถุส่วนต่อประสานปลายทาง

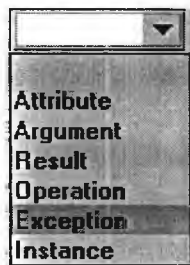
ก.3.2.2.3) หน้าจอการป้อนข้อมูล MappingFunctionDef เข้าสู่คลังส่วนต่อประสาน แบ่งออกเป็น 2 กรณีขึ้นอยู่กับทางเลือกระดับการแปลง (ดูหัวข้อ ก.3.2.2.1) ดังนี้

ก.3.2.2.3.1) หากระดับการแปลงที่เลือกเป็นระดับอินสแตนซ์ (Instance Level) แต่ยังไม่เคยป้อนข้อมูลฟังก์ชันการแปลงมาก่อนหน้า หรือเป็นระดับชนิด (Type Level) ที่เป็นการป้อนข้อมูลฟังก์ชันการแปลงครั้งแรกสำหรับส่วนต่อประสานคู่นี้ จะไม่มีหน้าจอสำหรับการนำฟังก์ชันการแปลงกลับมาใช้ใหม่ โดยมีขั้นตอนการทำงานดังแสดงในรูป ก13



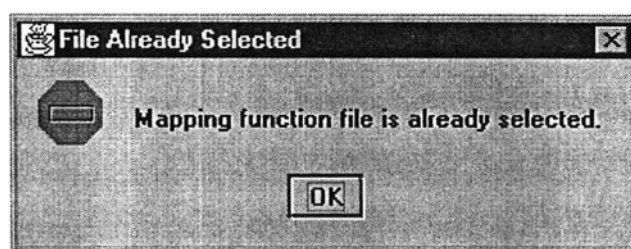
รูป ก13 หน้าจอของส่วนต่อประสานสำหรับผู้ใช้สำหรับการป้อนข้อมูล MappingFunctionDef
เข้าสู่โปรแกรมต้นแบบคลังส่วนต่อประสานที่รองรับความสัมพันธ์แบบเท่าเทียมกัน

1. ป้อนชื่อวัตถุ MappingFunctionDef
2. ป้อนรุ่นวัตถุ MappingFunctionDef
3. เลือกโหมดการแปลงที่ต้องการ



4. กด ปุ่ม **Browse...** เพื่อเลือกแฟ้มฟังก์ชันการแปลง เมื่อทำการเลือกไปแล้วปุ่มก็จะ

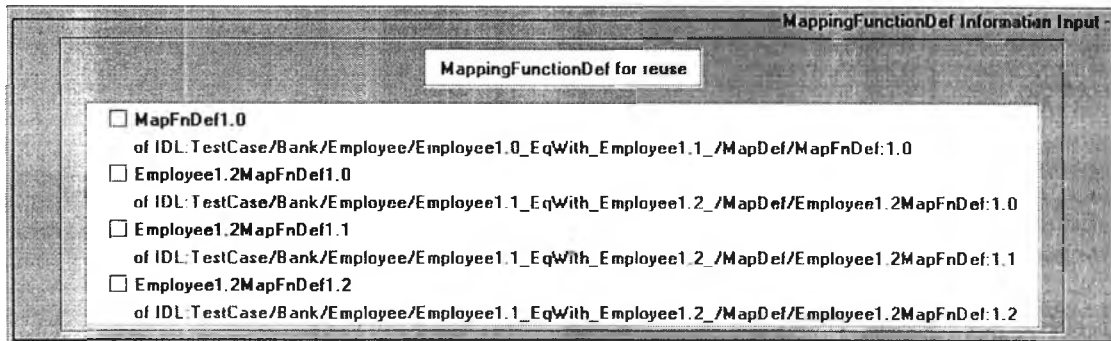
เปลี่ยนชื่อไปเป็นชื่อแฟ้มที่เลือก และจะเลือกซ้ำไม่ได้ หากกดเลือกอีกก็จะแสดงหน้าจอรายงานความผิดพลาดดังนี้



รูป ก14 หน้าจอแสดงข้อผิดพลาดของกรณีที่จะเลือกแฟ้มฟังก์ชันการแปลงอีกครั้ง

5. หากกดปุ่ม **Add MappingFnDef** ก็จะเป็นการเพิ่มฟังก์ชันการแปลง และจะย้อนกลับไปทำข้อ 1 ใหม่จนกว่าจะกดปุ่ม **Evolve** เพื่อทำการนำข้อมูลความสัมพันธ์แบบเท่าเทียมกันเข้าสู่ต้นแบบของคลังส่วนต่อประสาน หรือกดปุ่ม **Cancel** เพื่อยกเลิกการป้อนข้อมูลเข้าสู่ต้นแบบของคลังส่วนต่อประสาน

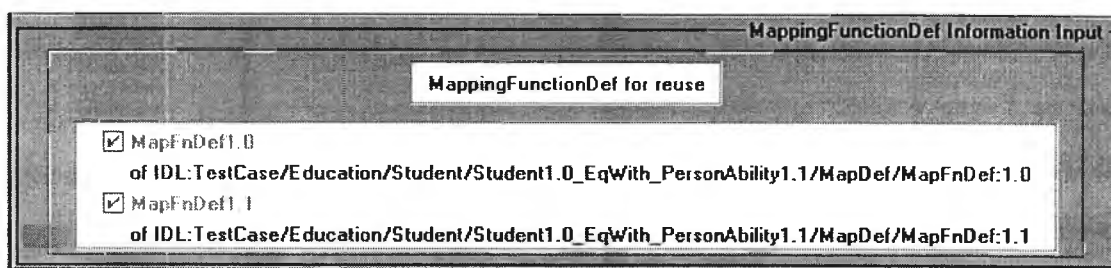
ก.3.2.2.3.2) หากระดับการแปลงที่เลือกเป็นระดับอินสแตนซ์ (Instance Level) และเคยป้อนข้อมูลฟังก์ชันการแปลงมาก่อนหน้า จะมีหน้าจอสำหรับการนำฟังก์ชันการแปลงกลับมาใช้ใหม่ ดังแสดงในรูป ก15 โดยตัวอย่างในที่ปรากฏในรูป ก15 จะเป็นฟังก์ชันการแปลงที่เคยป้อนเข้ามา และมีขั้นตอนในการทำงานดังนี้



รูป ก15 หน้าจอสำหรับการนำฟังก์ชันการแปลงที่เคยใส่เข้าสู่โปรแกรมต้นแบบกลับมาใช้ใหม่ ในกรณีที่ระดับการแปลงถูกเลือกเป็นระดับอินสแตนซ์ และเคยมีการป้อนข้อมูลฟังก์ชันการแปลงเข้าสู่โปรแกรมต้นแบบ

1. เลือกฟังก์ชันการแปลงที่ผู้ทำการเปลี่ยนรุ่นต้องการนำกลับมาใช้ใหม่โดยเลือกจากกล่องเลือกที่อยู่ด้านหน้าของฟังก์ชันการแปลงที่ต้องการ ซึ่งในขั้นตอนนี้ผู้ทำการเปลี่ยนรุ่นอาจจะเลือกหรือไม่ก็ได้ขึ้นอยู่กับความต้องการ
2. ทำตามข้อ 5. ในหัวข้อ ก.3.2.2.3.1

ก.3.2.2.3.3) หากระดับการแปลงที่เลือกเป็นระดับชนิด (Type Level) และเคยป้อนข้อมูลฟังก์ชันการแปลงสำหรับอินสแตนซ์ของส่วนต่อประสานคู่นี้มาแล้ว จะมีหน้าจอสำหรับการนำฟังก์ชันการแปลงกลับมาใช้ใหม่ ดังแสดงในรูป ก16 โดยตัวอย่างที่ปรากฏในรูป ก16 นั้นเป็นชุดของฟังก์ชันการแปลงสำหรับอินสแตนซ์ของส่วนต่อประสานคู่นี้ที่เคยป้อนเข้า สำหรับขั้นตอนในการทำงานในกรณีนี้คือ ให้กดปุ่ม Evolve ได้เลยเพราะจะเห็นได้ว่าชุดของฟังก์ชันการแปลงที่ปรากฏอยู่ในหน้าจอสำหรับการนำฟังก์ชันการแปลงกลับมาใช้ใหม่ ได้ถูกเลือกไว้แล้ว ซึ่งสอดคล้องกับการแปลงระดับชนิด ที่จะนำฟังก์ชันการแปลงชุดเดียวกันกลับมาใช้ใหม่ทั้งหมดในทุกๆ คู่ของอินสแตนซ์ของส่วนต่อประสานที่มีความสัมพันธ์แบบเท่าเทียมกัน

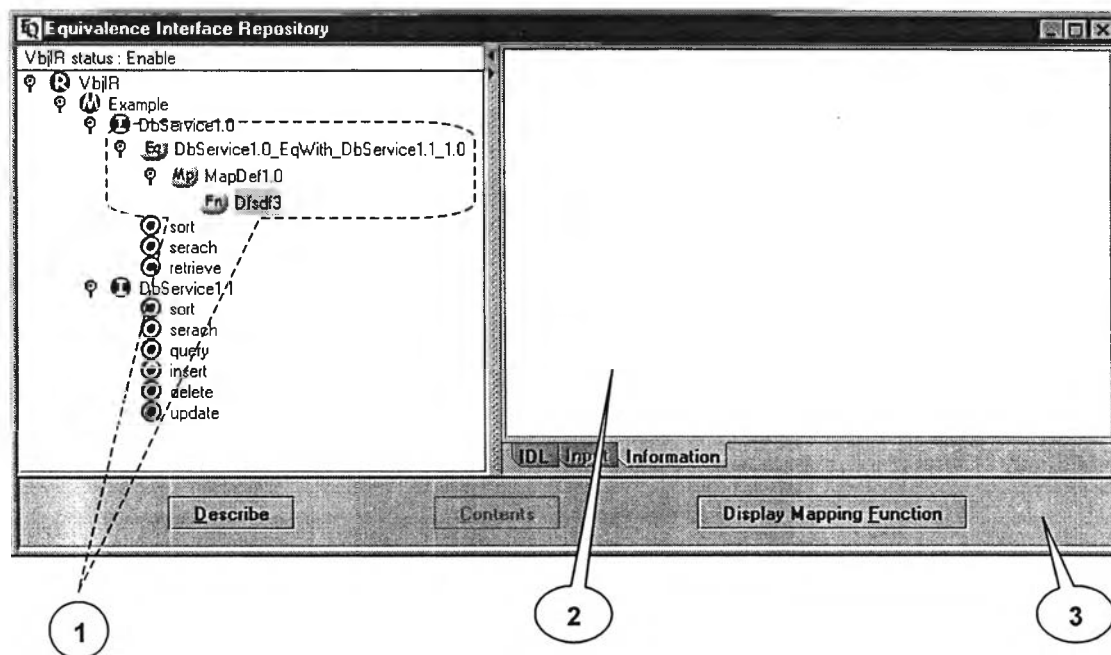


รูป ก16 หน้าจอสำหรับการนำฟังก์ชันการแปลงที่เคยใส่เข้าสู่โปรแกรมต้นแบบกลับมาใช้ใหม่ ในกรณีที่มีการเปลี่ยนแปลงถูกเลือกเป็น Type Level และเคยมีการป้อนข้อมูลฟังก์ชันการแปลงเข้าสู่โปรแกรมต้นแบบ

ก.3.2.3) ส่วนแสดงสารสนเทศของต้นแบบของคลังส่วนต่อประสาน (Information tab) แถบเลือกนี้จะเป็นส่วนต่อประสานสำหรับผู้ใช้ในการแสดงข้อมูลความสัมพันธ์แบบเท่าเทียมกัน ดังแสดงในรูป ก17 โดยจะมีปุ่มให้เลือกทั้งสิ้น 3 ปุ่ม การเปิดหรือปิดทางของปุ่มต่างๆ ขึ้นอยู่กับบัพที่เลือกจากพื้นที่แสดงวัตถุส่วนต่อประสานที่บรรจุอยู่ในต้นแบบคลังส่วนต่อประสาน ดังนี้

ก.3.2.3.1) บัพที่เลือกแล้วทำให้แถบเลือก Information เปิดทาง มีทั้งสิ้น 3 ชนิด

คือ



รูป ก17 หน้าจอแสดงสารสนเทศของต้นแบบของคลังส่วนต่อประสาน

- ถ้าเลือกบัพชนิด EquivalenceDef ปุ่ม Describe และปุ่ม Contents จะเปิดทาง ส่วนปุ่ม Display Mapping Function จะปิดทาง
- ถ้าเลือกบัพชนิด MappingDef ปุ่ม Describe และปุ่ม Contents จะเปิดทาง ส่วนปุ่ม Display Mapping Function จะปิดทาง
- ถ้าเลือกบัพชนิด MappingFunctionDef ปุ่ม Describe และปุ่ม Display Mapping Function จะเปิดทาง ส่วนปุ่ม Contents จะปิดทาง

ก.3.2.3.2) พื้นที่แสดงข้อมูลความสัมพันธ์แบบเท่าเทียมกัน แสดงไว้ในบริเวณ (2)

ก.3.2.3.3) พื้นที่แสดงปุ่มทางเลือกชนิดสารสนเทศ

โดยการทำงานของปุ่มทั้ง 3 มีดังนี้

- ปุ่ม **Describe** จะแสดงข้อมูลทั้งหมดเกี่ยวกับวัตถุที่สอดคล้องกับบัพที่เลือก
- ปุ่ม **Contents** จะแสดงข้อมูลของวัตถุทั้งหมดที่บรรจุอยู่ในวัตถุที่สอดคล้องกับบัพที่เลือกนั้น
- ปุ่ม **Display Mapping Function** จะแสดงรหัสต้นทางของเพิ่มฟังก์ชันการแปลงที่สอดคล้องกับบัพที่เลือก

ประวัติผู้วิจัย

นายสรยุทธ อังคนานุกิจ เกิดวันที่ 19 พฤศจิกายน พ.ศ.2518 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ จากภาควิชาวิทยาศาสตร์คอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยรังสิต เมื่อปี พ.ศ. 2540 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เคยได้รับทุนการศึกษาจากมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสาร เมื่อปี 2540-2541

