บทที่ 3
การแปลงคำอธิบายบริการระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล

ในบทนี้จะได้กล่าวถึงการแปลงคำอธิบายระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล รวมถึงการเพิ่มขยายบริการเทรดเดอร์ของคอร์บาให้มีความสามารถในการแปลงคำอธิบายบริการทั้งจากคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล และจากเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคอร์บาเทรดเดอร์

บริการเทรดเดอร์ของคอร์บานั้นมีคำอธิบายบริการอยู่สองชนิดคือ

1. คำอธิบายชนิดของบริการ
2. คำอธิบายข้อเสนอบริการ

ดังนั้นการแปลงคำอธิบายบริการสามารถแบ่งเป็นหัวข้อใหญ่ๆ ได้ดังนี้คือ

- การแปลงคำอธิบายชนิดของบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล
- การแปลงคำอธิบายข้อเสนอบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล
- การแปลงคำอธิบายชนิดของบริการจากรูปแบบเอ็กซ์เอ็มแอลไปเป็นรูปแบบของคอร์บาเทรดเดอร์
- การแปลงคำอธิบายข้อเสนอบริการจากรูปแบบเอ็กซ์เอ็มแอลไปเป็นรูปแบบของคอร์บาเทรดเดอร์

ในแต่ละหัวข้อจะได้กล่าวถึงการแปลงส่วนข้อมูลของคอร์บาเทรดเดอร์โดยผ่านทางไอดีแอลที่กำหนดโดยโอเอ็มจี เพื่อสร้างวัตถุดีโอเอ็มผ่านทางไอดีแอลซึ่งกำหนดโดยดับเบิลยูทีซี ซึ่งสามารถนำไปประยุกต์กับการใช้งานในภาษาโปรแกรมอื่นๆ ที่ทำการสร้างส่วนต่อประสานในการเขียนโปรแกรมประยุกต์ตามข้อกำหนดเหล่านี้

ดีโอเอ็มจะมีส่วนต่อประสานในการเขียนโปรแกรมประยุกต์ที่มีตัวกระทำการพื้นฐานในการสร้างวัตถุดังนี้คือ

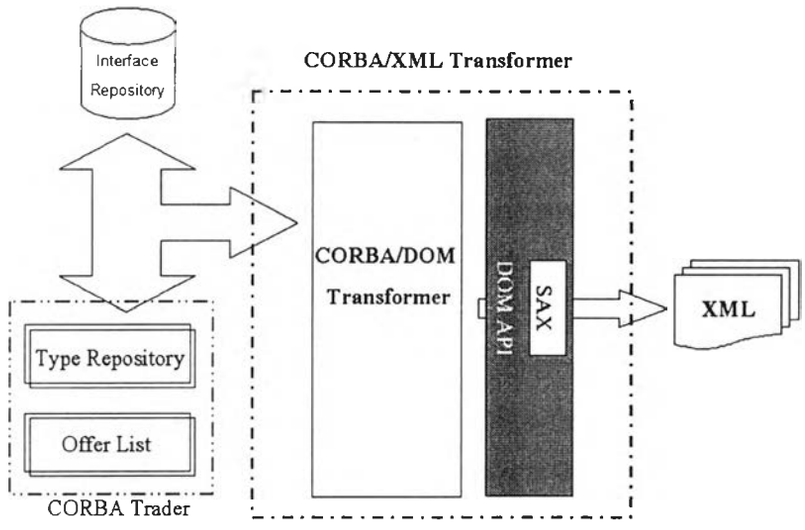
- Element Document::createElement(in DOMString tagName) – สำหรับการสร้างส่วนย่อยที่ระบุในพารามิเตอร์ ผลลัพธ์ที่ได้คือวัตถุส่วนย่อยที่สามารถนำไประบุลักษณะประจำ (Attribute) ภายหลังได้
- Text Document::createTextNode(in DOMString data) – สำหรับการสร้างโหนดข้อความ (Text Node) ของส่วนย่อย
- Node Node::appendChild(in Node newChild) – สำหรับการเพิ่มโหนดเช่นโหนดข้อความหรือโหนดส่วนย่อยลงไปในส่วนย่อยที่เรียกใช้ โดยจะได้ผลลัพธ์เป็นโหนดที่ถูกเพิ่มเข้าไป
- void Element::setAttribute(in DOMString name, in DOMString value) – สำหรับการเพิ่มชื่อ และค่าของลักษณะประจำให้แก่ส่วนย่อย

ในการเข้าถึงข้อมูลภายในวัตถุดีโอเอ็มสามารถทำได้โดยผ่านข้อกำหนดส่วนต่อประสานในการเขียนโปรแกรมประยุกต์เช่นเดียวกันคือ

- Document::documentElement – สำหรับการนำวัตถุส่วนย่อยที่เป็นรากของเอกสารเอ็กซ์เอ็มแอลเพื่อใช้ดึงข้อมูลอื่นๆ ต่อไป
- NodeList Element::getElementsByTagName(in DOMString name) – สำหรับการนำโหนดล่าง (Descendant Node) ทั้งหมดของส่วนย่อยนี้ที่มีชื่อตาม name ที่อยู่ในรูปของรายการของโหนด ใช้เพื่อการค้นหาส่วนย่อยภายในภายใต้ส่วนย่อยหนึ่งๆ
- DOMString Element::getAttribute(in DOMString name) – สำหรับการหาค่าของลักษณะประจำของส่วนย่อย
- NodeList Node::childNodes – สำหรับการดึงรายการของโหนดที่เป็นโหนดล่างของส่วนย่อย ซึ่งอาจเป็นโหนดข้อความ หรือส่วนย่อยภายในก็ได้

3.1 การแปลงคำอธิบายชนิดของบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล

ในการนำข้อมูลของคำอธิบายชนิดของบริการในรูปแบบของคอร์บามาสร้างเป็นเอกสารเอ็กซ์เอ็มแอลนั้นข้อมูลชนิดของบริการที่จะนำมาแปลงประกอบไปด้วยสองส่วนคือ จากคลังชนิดของบริการ และจากคลังส่วนต่อประสาน ข้อมูลทั้งสองส่วนนี้จะถูกนำมาสร้างเป็นวัตถุดีโอเอ็มโดยที่วัตถุดีโอเอ็มนี้จะต้องเป็นไปตามดีทีดีของคำอธิบายชนิดที่ถูกออกแบบไว้ดังที่จะกล่าวถึงในหัวข้อ 3.1.1 จากนั้นจึงนำวัตถุดีโอเอ็มนี้ไปสร้างเป็นเอกสารเอ็กซ์เอ็มแอลต่อไป ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 การแปลงคำอธิบายชนิดของบริการจากคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล

3.1.1 ดีทีดีของคำอธิบายชนิดของบริการ

ดังที่กล่าวไว้แล้วในหัวข้อที่ 2.2.3 ดีทีดีของคำอธิบายชนิดของบริการนี้จะเปรียบเสมือนไวยากรณ์สำหรับกำหนดโครงร่างของเอกสารเอ็กซ์เอ็มแอลที่จะถูกสร้างขึ้นให้เป็นไปตามที่กำหนด ดังนั้นผู้ที่จะนำเอกสารเอ็กซ์เอ็มแอลของคำอธิบายชนิดของบริการไปใช้จะสามารถทราบถึงลักษณะของเอกสารได้จากดีทีดีนี้

ตารางที่ 3.1 ดีทีดีของคำอธิบายชนิดของบริการ

```
1  <!ELEMENT ServiceTypeDescription (Interface?, TraderServiceType)>
2  <!ELEMENT Interface (BaseInterfaces?, Constant*, Attribute*, Operation*)>
3  <!ATTLIST   Interface      Id          CDATA       #REQUIRED
4                             Name        CDATA       #REQUIRED
5                             Version     CDATA       #REQUIRED>
6  <!ELEMENT BaseInterfaces (BaseInterface*, Link*)>
7  <!ELEMENT BaseInterface EMPTY>
8  <!ATTLIST BaseInterface    Id          CDATA       #REQUIRED
9                             Name        CDATA       #REQUIRED>
10 <!ELEMENT Link EMPTY>
11 <!ATTLIST Link             Source      CDATA       #REQUIRED
12                            Dest        CDATA       #REQUIRED>
13 <!ELEMENT Constant EMPTY>
14 <!ATTLIST Constant         Id          CDATA       #REQUIRED
15                            Name        CDATA       #REQUIRED
16                            Version     CDATA       #REQUIRED
17                            Type        CDATA       #REQUIRED
18                            Value       CDATA       #REQUIRED
19                            Derived     (YES | NO) "NO">
20 <!ELEMENT Attribute EMPTY>
21 <!ATTLIST Attribute        Id          CDATA       #REQUIRED
22                            Name        CDATA       #REQUIRED
23                            Version     CDATA       #REQUIRED
24                            Type        CDATA       #REQUIRED
25                            Mode        (NORMAL | READONLY) "NORMAL"
26                            Derived     (YES | NO) "NO">
```

ตารางที่ 3.1 ดีทีดีของคำอธิบายชนิดของบริการ (ต่อ)

```
27 <!ELEMENT Operation (Parameter*, Exception*, Context*)>
28 <!ATTLIST Operation      Id          CDATA       #REQUIRED
29                          Name        CDATA       #REQUIRED
30                          Version     CDATA       #REQUIRED
31                          Type        CDATA       #REQUIRED
32                          Mode        (NORMAL | ONEWAY) "NORMAL"
33                          Derived     (YES | NO) "NO">
34 <!ELEMENT Parameter EMPTY>
35 <!ATTLIST Parameter      Name        CDATA       #REQUIRED
36                          Type        CDATA       #REQUIRED
37                          Mode        (IN | OUT | INOUT) "IN">
38 <!ELEMENT Exception (Member)*>
39 <!ATTLIST Exception      Id          CDATA       #REQUIRED
40                          Name        CDATA       #REQUIRED
41                          Version     CDATA       #REQUIRED
42                          Derived     (YES | NO) "NO">
43 <!ELEMENT Member EMPTY>
44 <!ATTLIST Member         Name        CDATA       #REQUIRED
45                          Type        CDATA       #REQUIRED>
46 <!ELEMENT Context (#PCDATA)>
47 <!ELEMENT TraderServiceType (BaseServiceTypes?, Property*)>
48 <!ATTLIST TraderServiceType  Id      CDATA       #REQUIRED
49                              Name    CDATA       #REQUIRED
50                              Masked  (YES | NO) "NO">
51 <!ELEMENT BaseServiceTypes (BaseServiceType*, Link*)>
52 <!ELEMENT BaseServiceType EMPTY>
53 <!ATTLIST BaseServiceType   Name CDATA       #REQUIRED>
54 <!ELEMENT Property EMPTY>
55 <!ATTLIST Property       Name        CDATA       #REQUIRED
56                          Type        CDATA       #REQUIRED
57                          Mode        (NORMAL|READONLY|MANDATORY|
58                                       MANDATORY_READONLY) "NORMAL"
                                         Derived (YES | NO) "NO">
```

จากตารางที่ 3.1 แสดงถึงดีทีดีของคำอธิบายชนิดของบริการ ในส่วนของวากยสัมพันธ์ (Syntax) สามารถดูรายละเอียดได้ใน [6] และสามารถสรุปข้อกำหนดของเอกสารเอ็กซ์เอ็มแอลที่จะถูกสร้างขึ้นตามดีทีดีของชนิดของบริการได้ดังนี้คือ

บรรทัดที่	คำอธิบาย
1	ราก (Root) ของเอกสารจะต้องเป็น ServiceTypeDescription โดยประกอบด้วยส่วนย่อย (Element) อีกสองส่วนคือ • Interface แสดงรายละเอียดคำอธิบายของส่วนต่อประสานซึ่งจะมีหรือไม่ก็ได้ • TraderServiceType แสดงรายละเอียดคำอธิบายชนิดของบริการ
2 – 46	ส่วนย่อย Interface จะประกอบไปด้วยส่วนย่อยภายใน อีกสี่ส่วนที่อธิบายรายละเอียดของส่วนต่อประสาน ประกอบไปด้วย • BaseInterfaces เป็นส่วนแสดงโครงสร้างของส่วนต่อประสานที่ถูกสืบทอดมาโดยส่วนต่อประสานนี้ซึ่งจะมีหรือไม่ก็ได้ • Constant เป็นการประกาศค่าคงที่ภายในส่วนต่อประสานซึ่งอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ • Attribute เป็นการประกาศลักษณะประจำของส่วนต่อประสานซึ่งอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ • Operation เป็นการประกาศตัวกระทำการของส่วนต่อประสานนี้ซึ่งอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ โดยส่วนย่อยนี้อาจจะประกอบไปด้วยส่วนย่อยอีกสามส่วนคือ Parameter, Exception และ Context รายละเอียดของส่วนต่อประสาน ซึ่งจะต้องประกอบไปด้วยข้อมูลลักษณะประจำเหล่านี้คือ • Id เป็นส่วนเชื่อมต่อระหว่างคลังชนิดของบริการ และคลังส่วนต่อประสาน • Name คือชื่อของส่วนต่อประสาน • Version คือรุ่นของส่วนต่อประสาน ในส่วนย่อย Constant, Attribute, Operation, Exception และ Context จะต้องประกอบไปด้วยรายละเอียดข้อมูลภายในที่เป็นลักษณะประจำที่ต้องมีอีกคือ Id, Name และ Version • Id คือตัวระบุ (Identifier) โดยตัวระบุนี้จะไม่ซ้ำกันภายใต้คลังส่วนต่อประสานหนึ่งๆ โดยมีลักษณะเป็นข้อมูลเป็นอักขระ • Name คือชื่อของส่วนย่อย โดยมีลักษณะเป็นข้อมูลอักขระ • Version คือรุ่นของส่วนย่อย โดยมีลักษณะเป็นข้อมูลอักขระ สำหรับส่วนย่อย Constant, Attribute, Operation และ Parameter จะประกอบไปด้วยรายละเอียดข้อมูลภายในที่เป็นลักษณะประจำที่ต้องมีอีกหนึ่งส่วนคือ

บรรทัดที่	คำอธิบาย
	• Type เป็นชนิดของส่วนย่อย ซึ่งอาจเป็นชนิดข้อมูลพื้นฐานสำหรับไอดีแอล (Interface Definition Language (IDL) Primitive Type) อยู่ในรูปของอักขระดังนี้คือ short, long, unsigned short, unsigned long, float, double, char, string, boolean, octet, longlong, longdouble, ulonglong, wchar, wstring และ void หรือชนิดที่กำหนดขึ้นเอง (User Defined Type) ซึ่งจะอยู่ในรูปของอักขระที่แสดงถึงชื่อของชนิดภายในคลังส่วนต่อประสานนั้น
6	BaseInterfaces เป็นชุดของส่วนต่อประสานที่ถูกสืบทอดมา โดยมีส่วนย่อยภายในสองส่วนย่อยที่ดัดแปลงจากการอธิบายโครงสร้างของเว็บด้วยเอ็กซ์เอ็มแอล [13] ประกอบไปด้วย • BaseInterface ซึ่งแสดงตัวระบุ และชื่อของส่วนต่อประสานที่สืบทอดมาทั้งหมด • Link เป็นส่วนแสดงความสัมพันธ์ลำดับชั้นของคลาส (Class Hierarchy) ที่สืบทอดมาทั้งหมด
7 – 9	ส่วนย่อย BaseInterface จะแสดงถึงตัวระบุ และชื่อของส่วนต่อประสานของส่วนต่อประสานที่สืบทอดมาที่อาจมีมากกว่าหนึ่ง หรือไม่มีก็ได้ โดยส่วนย่อยนี้จะแสดงถึงส่วนต่อประสานที่ได้รับการสืบทอดมาทั้งหมดตั้งแต่รากของลำดับชั้นของคลาส ส่วนย่อยนี้จะประกอบไปด้วยลักษณะประจำสองชนิดคือ • Id ที่เป็นตัวระบุส่วนต่อประสานภายในคลังส่วนต่อประสานที่ส่วนต่อประสานของเอกสารเอ็กซ์เอ็มแอลนี้ได้รับการสืบทอดมา • Name ซึ่งเป็นชื่อของส่วนต่อประสานนี้ที่ส่วนต่อประสานของเอกสารเอ็กซ์เอ็มแอลนี้ได้รับการสืบทอดมา
10 – 12	ส่วนย่อย Link เป็นส่วนแสดงความสัมพันธ์ของลำดับชั้นของคลาสที่ทำการสืบทอดมาทั้งหมดของส่วนต่อประสาน โดยหนึ่งส่วนย่อยจะแสดงถึงความสัมพันธ์ของส่วนต่อประสานหนึ่งคู่ที่มีความสัมพันธ์แบบแม่และลูก (Parent and Child) ประกอบไปด้วย • Source จะมีข้อมูลเป็นตัวระบุของส่วนต่อประสาน โดยเป็นตัวระบุส่วนต่อประสานของส่วนต่อประสานย่อย (Sub Interface) ที่สืบทอดจากส่วนต่อประสานแม่ (Parent Interface) ในส่วนย่อย Dest • Dest จะมีข้อมูลเป็นตัวระบุของส่วนต่อประสาน โดยเป็นตัวระบุส่วนต่อประสานของส่วนต่อประสานแม่
13 – 19	ส่วนย่อย Constant แสดงถึงรายละเอียดของค่าคงที่ภายในส่วนต่อประสาน ประกอบไปด้วยข้อมูลลักษณะประจำที่สำคัญที่ต้องมีดังนี้คือ • Id เป็นตัวระบุภายในคลังส่วนต่อประสานสำหรับค่าคงที่นี้ • Name เป็นชื่อของค่าคงที่

บรรทัดที่	คำอธิบาย
	• Version คือรุ่นของค่าคงที่ • Type เป็นชนิดของค่าคงที่ • Value เป็นค่าของค่าคงที่ โดยมีข้อมูลเป็นอักขระที่ตรงกับชนิดของค่าคงที่ข้างต้น • Derived เป็นส่วนแสดงว่าค่าคงที่นี้เป็นค่าคงที่ที่ได้รับการสืบทอดมาจากส่วนต่อประสานอื่นหรือไม่ โดยถ้าเป็นค่าคงที่ที่ได้รับการสืบทอดมาจะมีค่าเป็น YES และถ้าเป็นค่าคงที่ที่ไม่ได้รับการสืบทอดมาจะมีค่าเป็น NO (ค่าโดยปริยาย (Default) คือ NO)
20 – 26	ส่วนย่อย Attribute จะแสดงถึงรายละเอียดของลักษณะประจำของส่วนต่อประสาน ประกอบไปด้วยข้อมูลลักษณะประจำของส่วนย่อย Attribute ดังนี้คือ • Id เป็นตัวระบุภายในคลังส่วนต่อประสานสำหรับลักษณะประจำนี้ • Name เป็นชื่อของลักษณะประจำ • Version เป็นรุ่นของลักษณะประจำ • Type เป็นชนิดของลักษณะประจำ • Mode เป็นภาวะของลักษณะประจำซึ่งจะมีค่าเป็น NORMAL หรือ READONLY โดย NORMAL คือภาวะของลักษณะประจำที่สามารถทำการเข้าถึงได้ทั้งแบบอ่านและเขียน ส่วน READONLY คือภาวะของลักษณะประจำที่สามารถเข้าถึงแบบอ่านได้อย่างเดียว (ค่าโดยปริยายคือ NORMAL) • Derived เป็นส่วนแสดงว่าลักษณะประจำนี้เป็นลักษณะประจำที่ได้รับการสืบทอดมาจากส่วนต่อประสานอื่นหรือไม่ โดยถ้าเป็นลักษณะประจำที่ได้รับการสืบทอดมาจะมีค่าเป็น YES และถ้าเป็นลักษณะประจำที่ไม่ได้รับการสืบทอดมาจะมีค่าเป็น NO (ค่าโดยปริยายคือ NO)
27 – 33	ส่วนย่อย Operation จะแสดงถึงข้อมูลในการกำหนดตัวกระทำการของส่วนต่อประสาน โดยมีส่วนย่อยภายในสามส่วนคือ • Parameter คือพารามิเตอร์ของตัวกระทำการนี้ โดยอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ • Exception คือข้อยกเว้นที่อาจเกิดขึ้นจากตัวกระทำการนี้ โดยอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ • Context คือคอนเท็กซ์ของตัวกระทำการนี้ โดยอาจมีมากกว่าหนึ่งหรือไม่มีก็ได้ และประกอบด้วยรายละเอียดลักษณะประจำของตัวกระทำการดังนี้คือ • Id เป็นตัวระบุภายในคลังส่วนต่อประสานสำหรับตัวกระทำการนี้ • Name เป็นชื่อของตัวกระทำการ • Version คือรุ่นของตัวกระทำการ • Type เป็นชนิดของผลลัพธ์ของตัวกระทำการ • Mode เป็นภาวะของตัวกระทำการซึ่งจะมีค่าเป็น NORMAL หรือ ONEWAY โดย

บรรทัดที่	คำอธิบาย
	NORMAL คือภาวะของตัวกระทำการที่เมื่อเรียกแล้วจะรอผลลัพธ์จากตัวกระทำการ ส่วน ONEWAY คือภาวะของตัวกระทำการเมื่อเรียกแล้วจะไม่รอผลลัพธ์จากตัวกระทำการ ซึ่งตัวกระทำการจะพยายามทำให้ดีที่สุด (Best Effort) โดยไม่รับรองว่าจะสำเร็จหรือไม่ (ค่าโดยปริยายคือ NORMAL) • Derived เป็นส่วนแสดงว่าตัวกระทำการนี้เป็นตัวกระทำการที่ได้รับการสืบทอดมาจากส่วนต่อประสานอื่นหรือไม่ โดยถ้าเป็นตัวกระทำการที่ได้รับการสืบทอดมาจะมีค่าเป็น YES และถ้าเป็นตัวกระทำการที่ไม่ได้รับการสืบทอดมาจะมีค่าเป็น NO (ค่าโดยปริยายคือ NO)
34 – 35	ส่วนย่อย Parameter เป็นพารามิเตอร์ของตัวกระทำการมีรายละเอียดลักษณะประจำดังนี้ • Name เป็นชื่อของพารามิเตอร์ • Type เป็นชนิดของพารามิเตอร์ • Mode เป็นภาวะของพารามิเตอร์ซึ่งจะมีค่าเป็น IN, OUT หรือ INOUT โดย IN คือพารามิเตอร์ที่ถูกส่งค่าจากผู้เรียกตัวกระทำการไปยังตัวกระทำการ OUT คือพารามิเตอร์จะถูกส่งค่าจากตัวกระทำการกลับมาให้ผู้เรียกตัวกระทำการเมื่อทำงานเสร็จแล้ว และ INOUT คือพารามิเตอร์ที่ถูกกำหนดค่าเริ่มต้น (Initialize) จากผู้เรียกตัวกระทำการ และตัวกระทำการสามารถทำการเปลี่ยนแปลงค่าแล้วคืนกลับมาให้กับผู้เรียกได้เมื่อทำงานเสร็จแล้ว (ค่าโดยปริยายคือ NORMAL)
38 – 42	ส่วนย่อย Exception เป็นข้อยกเว้นที่อาจเกิดขึ้นจากตัวกระทำการ มีส่วนย่อยภายในคือ • Member ซึ่งเป็นสมาชิกภายในข้อยกเว้น โดยจะเป็นตัวอธิบายข้อมูลเพิ่มเติมในข้อยกเว้น และมีรายละเอียดลักษณะประจำดังนี้ • Id เป็นตัวระบุภายในคลังส่วนต่อประสานสำหรับข้อยกเว้นนี้ • Name เป็นชื่อของข้อยกเว้น • Version เป็นรุ่นของข้อยกเว้น • Derived เป็นส่วนแสดงว่าข้อยกเว้นนี้เป็นข้อยกเว้นที่ได้รับการสืบทอดมาจากส่วนต่อประสานอื่นหรือไม่ โดยถ้าเป็นข้อยกเว้นที่ได้รับการสืบทอดมาจะมีค่าเป็น YES และถ้าเป็นข้อยกเว้นที่ไม่ได้รับการสืบทอดมาจะมีค่าเป็น NO (ค่าโดยปริยายคือ NO)
43 – 45	ส่วนย่อย Member เป็นส่วนอธิบายสมาชิกของข้อยกเว้นที่ประกอบไปด้วยลักษณะประจำดังนี้ • Name เป็นชื่อของสมาชิก • Type เป็นชนิดของสมาชิก
46	ส่วนย่อย Context เป็นตัวระบุคอนเท็กซ์ของตัวกระทำการมีลักษณะเป็นข้อมูลอักขระ

บรรทัดที่	คำอธิบาย
47 – 50	ส่วนย่อย TraderServiceType เป็นส่วนแสดงรายละเอียดของคำอธิบายชนิดของบริการที่ได้มาจากเทรดเดอร์ ประกอบไปด้วยส่วนย่อยภายในดังนี้คือ • BaseServiceTypes แสดงถึงโครงสร้างชนิดของบริการที่สืบทอดมา ซึ่งจะมีหรือไม่ก็ได้ • Property เป็นข้อมูลคุณสมบัติที่กำหนดขึ้นเพื่ออธิบายชนิดบริการเพิ่มเติมจากในส่วนต่อประสาน ในส่วนย่อย TraderServiceType ยังมีรายละเอียดเกี่ยวกับตัวชนิดของบริการเองคือ • Id เป็นตัวระบุของส่วนต่อประสานของชนิดของบริการนี้ โดยมีลักษณะเช่นเดียวกับตัวระบุส่วนต่อประสานในคลังส่วนต่อประสาน • Name เป็นชื่อของชนิดของบริการ • Masked เป็นสถานะของชนิดของบริการนี้ว่าจะสามารถให้ข้อเสนอบริการสามารถทำการโฆษณาได้หรือไม่ โดยถ้าไม่สามารถทำการโฆษณาได้จะมีค่าคือ YES และถ้าสามารถโฆษณาได้จะมีค่าเป็น NO (ค่าโดยปริยายคือ NO)

51 – 53	ส่วนย่อย BaseServiceTypes เป็นส่วนแสดงโครงสร้างของชนิดของบริการที่สืบทอดมาซึ่งมีลักษณะโครงสร้างอธิบายด้วยส่วนย่อย Link เช่นเดียวกันส่วนย่อย BaseInterface โดยประกอบไปด้วยส่วนย่อยที่แสดงชื่อของชนิดของบริการที่สืบทอด และส่วนเชื่อมแสดงโครงสร้างดังนี้
	• BaseServiceType เป็นส่วนย่อยที่แสดงชื่อของชนิดของบริการที่สืบทอดมามีรายละเอียดคือ
	  • Name เป็นชื่อของชนิดของบริการที่สืบทอดมา
	• Link เป็นส่วนย่อยที่แสดงแสดงความสัมพันธ์ของลำดับชั้นของชนิดของบริการที่ทำการสืบทอดมาทั้งหมด โดยหนึ่งส่วนย่อยจะแสดงถึงความสัมพันธ์ของชนิดของบริการหนึ่งคู่ที่มีความสัมพันธ์แบบแม่และลูก ประกอบไปด้วย
	  • Source จะมีข้อมูลเป็นชื่อของชนิดของบริการ โดยเป็นชื่อของชนิดของบริการย่อย (Sub Type) ที่สืบทอดจากชนิดของบริการแม่ (Parent Type)
	  • Dest จะมีข้อมูลเป็นชื่อของชนิดของบริการ โดยเป็นชื่อของชนิดของบริการแม่

54 – 59	ส่วนย่อย Property จะแสดงรายละเอียดของคุณสมบัติที่ทำการประกาศไว้สำหรับชนิดของบริการเพื่อเป็นข้อมูลสำหรับให้ผู้ทำการโฆษณาได้ใช้ในการโฆษณาบริการดังนี้คือ • Name เป็นชื่อของคุณสมบัติที่อยู่ในรูปของอักขระ • Type เป็นชนิดของคุณสมบัติ โดยมีลักษณะเช่นเดียวกับชนิดข้อมูลพื้นฐานของไอดีแอล • Mode เป็นภาวะของคุณสมบัติซึ่งจะมีค่าเป็น NORMAL, READONLY, MANDATORY หรือ MANDATORY_READONLY โดย NORMAL คือคุณสมบัติที่ไม่

บรรทัดที่	คำอธิบาย
	จำเป็นต้องระบุเมื่อมีการโฆษณา และสามารถทำการเปลี่ยนแปลงได้ READONLY หมายถึงเมื่อมีการระบุค่าให้กับคุณสมบัตินี้แล้วจะไม่สามารถแก้ไขได้ MANDATORY หมายถึงข้อเสนอบริการเมื่อทำการโฆษณาจะต้องระบุค่าของคุณสมบัตินี้ และ MANDATORY_READONLY คือเมื่อข้อเสนอบริการทำการโฆษณาจะต้องระบุค่าของคุณสมบัตินี้ รวมถึงไม่สามารถทำการแก้ไขค่าได้อีก (ค่าโดยปริยายคือ NORMAL) • Derived เป็นส่วนแสดงว่าคุณสมบัตินี้เป็นคุณสมบัติที่ได้รับการสืบทอดมาจากชนิดของบริการอื่นหรือไม่ โดยถ้าเป็นคุณสมบัติที่ได้รับการสืบทอดมาจะมีค่าเป็น YES และถ้าเป็นคุณสมบัติที่ไม่ได้รับการสืบทอดมาจะมีค่าเป็น NO (ค่าโดยปริยายคือ NO)

3.1.2 การแปลงข้อมูลชนิดของบริการจากคลังส่วนต่อประสาน

ในการแปลงข้อมูลชนิดของบริการจากคลังส่วนต่อประสานจะทำได้โดยการเรียกใช้ตัวกระทำการต่างๆ ที่กำหนดไว้ในข้อกำหนดของคลังส่วนต่อประสานเพื่อดึงข้อมูลส่วนต่อประสานมาใช้ในการแปลง ทั้งนี้จะมีกฎในการนำคำอธิบายส่วนต่อประสานมาสร้างเป็นวัตถุดีโอเอ็มดังนี้คือ

Interface - ในส่วนย่อย Interface สามารถนำข้อมูล CORBA::InterfaceDef จากการเรียก CORBA::Repository::lookup_id(if_name) มาใช้โดย if_name คือตัวระบุส่วนต่อประสานที่ได้จากคลังชนิดของบริการ จากนั้นรายละเอียดของส่วนต่อประสานที่ต้องใช้คือ Id, Name และ Version สามารถนำมาจาก CORBA::InterfaceDef::FullInterfaceDescription ซึ่งได้จากการเรียกใช้ CORBA::InterfaceDef::describe_interface() โดยในแต่ละลักษณะประจำสามารถหาได้ดังนี้

- Id = CORBA::InterfaceDef::FullInterfaceDescription::id
- Name = CORBA::InterfaceDef::FullInterfaceDescription::name
- Version = CORBA::InterfaceDef::FullInterfaceDescription::version

BaseInterfaces - การนำนิยามส่วนต่อประสานที่ได้รับการสืบทอดมาโดยตรง (Explicit Inheritance) สามารถทำได้จากการเรียก CORBA::InterfaceDef::base_interfaces() โดยผลลัพธ์ที่ได้จะเป็นชุดของนิยามส่วนต่อประสานที่ได้รับการสืบทอดมา สำหรับนิยามส่วนต่อประสานที่สืบทอดมาทางอ้อม (Implicit Inheritance) และโครงสร้างสามารถหาได้โดยการวนซ้ำเพื่อเรียก CORBA::InterfaceDef::base_interfaces() ไปยังรายการนิยามส่วนต่อประสานที่สืบทอดมาโดยตรง

Constant - จากนิยามส่วนต่อประสาน สามารถหารายการของนิยามค่าคงที่ที่ไม่ได้สืบทอดมาได้จาก CORBA::InterfaceDef::contents(CORBA::DefinitionKind::dk_Constant, true) โดยผลลัพธ์ที่ได้จะอยู่ในรูปของชุดของ CORBA::Contained ที่สามารถแปลงให้เป็นนิยามค่าคงที่ได้ นิยามค่าคงที่ทั้งหมดทั้งที่ไม่ได้รับสืบทอดมาและที่ได้รับสืบทอดมาสามารถหาได้จาก CORBA::InterfaceDef::contents (CORBA::DefinitionKind::dk_Constant, false) โดยผลลัพธ์ที่ได้จะอยู่ในรูปของชุดของ CORBA::Contained เช่นเดียวกัน ผลลัพธ์ของนิยามค่าคงที่นี้ที่ไม่ได้อยู่ในผลลัพธ์ของชุดแรกจะเป็นส่วนย่อยที่มีลักษณะประจำ Derived เป็น YES ส่วนลักษณะประจำอื่นๆ สามารถหาได้ดังนี้

- Id = CORBA::ConstantDef::id()
- Name = CORBA::ConstantDef::name()
- Version = CORBA::ConstantDef::version()
- Type = CORBA::ConstantDef::type_def()
- Value = CORBA::ConstantDef::version()

Attribute - นิยามลักษณะประจำ CORBA::AttributeDef สามารถหาได้จากการค้นหาโดยเรียกใช้ CORBA::Repository::lookup_id(RepositoryId) โดยที่ RepositoryId คือ CORBA::InterfaceDef::FullInterfaceDescription::attributes[i]::id ที่ได้จากการหาส่วนย่อย Interface ข้างต้น โดยที่ i คือดัชนีของลักษณะประจำทั้งหมด ทั้งที่ได้รับสืบทอดมา และไม่ได้รับสืบทอดมาของส่วนต่อประสานนี้ ลักษณะประจำอื่นๆ สามารถหาได้ดังนี้

- Id = CORBA::InterfaceDef::FullInterfaceDescription::attribute[i]::id
- Name = CORBA::InterfaceDef::FullInterfaceDescription::attribute[i]::name
- Version = CORBA::InterfaceDef::FullInterfaceDescription::attribute[i]::version
- Type = CORBA::AttributeDef::type_def()
- Mode = CORBA::InterfaceDef::FullInterfaceDescription::attribute[i]::mode

ในส่วนลักษณะประจำ Derived จะต้องมีการตรวจสอบลักษณะประจำของส่วนต่อประสานเป็นกรณีไปว่าเป็นลักษณะประจำที่ได้รับการสืบทอดมาหรือไม่

Operation - นิยามตัวกระทำการ CORBA::OperationDef สามารถหาได้จากการค้นหาโดยเรียกใช้ CORBA::Repository::lookup_id(RepositoryId) โดยที่ RepositoryId คือ CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::id ที่ได้จากการหาส่วนย่อย Interface ข้างต้น โดยที่ i คือดัชนีของตัวกระทำการทั้งหมดทั้งที่ได้รับการสืบทอดมา และไม่ได้รับการสืบทอดมา ลักษณะประจำของส่วนย่อย Operation สามารถหาได้ดังนี้

- Id = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::id
- Name = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::name
- Version = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::version
- Type = CORBA::OperationDef::result_def()
- Mode = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::mode

ในส่วนลักษณะประจำ Derived จะต้องมีการตรวจสอบว่าเป็นตัวกระทำการที่ได้รับการสืบทอดมาหรือไม่

Parameter - ส่วนย่อย Parameter ซึ่งเป็นส่วนย่อยภายใน Operation นั้นสามารถนำข้อมูลมาจาก CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::parameters[j] เมื่อ i เป็นดัชนีของตัวกระทำการทั้งที่ได้รับการสืบทอดมา และไม่ได้รับการสืบทอดมาของส่วนต่อประสาน และ j คือดัชนีของพารามิเตอร์ทั้งหมดของตัวกระทำการนี้ โดยหาลักษณะประจำของส่วนย่อย Parameter จาก

- Name = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::parameters [j]::name
- Type = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::parameters [j]::type_def
- Mode = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::parameters [j]::mode

Exception – ส่วนย่อยข้อยกเว้นเป็นส่วนย่อยภายในของส่วนย่อย Operation โดยสามารถนำข้อมูลมาจาก CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::exceptions[j] เมื่อ i คือดัชนีของตัวกระทำการ และ j คือดัชนีของข้อยกเว้น โดยหาลักษณะประจำของส่วนย่อย Exception ได้ดังนี้

- Id = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::exceptions[j]::id
- Name = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::exceptions [j]::name
- Version = CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::exceptions [j]::version

ในส่วนลักษณะประจำ Derived จะต้องมีการตรวจสอบว่าเป็นข้อยกเว้นที่ได้รับการสืบทอดมาหรือไม่

Member – สมาชิกของข้อยกเว้นสามารถนำมาจากนิยามข้อยกเว้น โดยการเรียกใช้ CORBA::ExceptionDef::members() โดยผลที่ได้จะอยู่ในรูปของ CORBA::StructMemberSeq ในขณะที่การหา ExceptionDef สามารถทำได้โดยการค้นหาตัวระบุภายในคลังส่วนต่อประสานโดยการเรียก CORBA::Repository::lookup_id(RepositoryId) เมื่อ RepositoryId คือ CORBA::InterfaceDef::FullInterfaceDescription::operations[i]::exceptions[j]::id ที่ได้จากการหาส่วนย่อย Interface ข้างต้น โดยที่ i คือดัชนีของตัวกระทำการทั้งหมดทั้งที่ได้รับการสืบทอดมา และไม่ได้รับการสืบทอดมา และ j คือดัชนีของข้อยกเว้นทั้งหมดของตัวกระทำการหนึ่งๆ สำหรับลักษณะประจำในแต่ละส่วนย่อย Member จะเป็นดังนี้

- Name = CORBA::StructMember::name
- Type = CORBA::StructMember::type_def

Context – ตัวระบุคอนเท็กซ์ของตัวกระทำการที่อยู่ในรูปของข้อมูลอักขระสามารถหาได้จากการเรียกใช้ CORBA::OperationsDef::contexts() โดยผลลัพธ์จะอยู่ในรูปของชุดของตัวระบุคอนเท็กซ์

จากไอดีแอลที่บันทึกอยู่ในคลังส่วนต่อประสาน เมื่อผ่านกฎในการแปลงที่กำหนดแล้วผลลัพธ์ที่ได้จะมีลักษณะดังตัวอย่างต่อไปนี้

ไอดีแอล	เอ็กซ์เอ็มแอล
interface BankService::CommonService {	<Interface Id = "IDL:BankService:1.0" Name = "BankService" Version = "1.0" > <BaseInterfaces> <BaseInterface Id="IDL:CommonService:1.0 Name="CommonService" /> <Link Source="IDL:BankService:1.0" Dest="IDL:CommonService:1.0" /> </BaseInterfaces>
const float fixed_interest = 10;	<Constant Id = "IDL:BankService/fixed_interest:1.0" Name = "fixed_interest" Version = "1.0" Type = "float" Value = "10" > </Constant>

ไอดีแอล	เอ็กซ์เอ็มแอล
attribute string account_name; ⇨	<Attribute Id = "IDL:BankService/account_name:1.0"
attribute float balance;	Name = "account_name"
	Version = "1.0"
exception Overdrawn {	Type = "string"
string reason; float amount;	Mode = "NORMAL"
};	</Attribute>
void deposit(in float amount);	.....
void withdraw(in float amount) ⇨	<Operation Id = "IDL:BankService/withdraw:1.0"
raises(Overdrawn);	Name = "withdraw"
	Version = "1.0"
float balance();	Type = "void"
};	Mode = "NORMAL" >
	<Parameter Name = "amount"
	Type = "float"
	Mode = "IN" />
	<Exception Id = "IDL:BankService/Overdrawn:1.0"
	Name = "Overdrawn"
	Version = "1.0" >
	<Member Name = "reason" Type = "string" />
	<Member Name = "amount" Type = "float" />
	</Exception>
	</Operation>
	</Interface>

3.1.3 การแปลงข้อมูลชนิดของบริการจากคลังชนิดของบริการ

ดังที่กล่าวมาในหัวข้อ 2.2.1 เราสามารถนำข้อมูลคำอธิบายชนิดของบริการจากคลังชนิดของบริการมาใช้โดยการเรียกใช้ตัวกระทำการต่างๆ ที่กำหนดไว้ในข้อกำหนดของบริการเทรดเดอร์ จากนั้นนำข้อมูลที่ได้มาสร้างเป็นวัตถุดีโอเอ็มดังนี้

TraderServiceType - ส่วนย่อย TraderServiceType จะประกอบไปด้วยตัวระบุ ชื่อ และสภาวะว่าเป็น Masked หรือไม่ โดยสามารถนำข้อมูลของชนิดของบริการเหล่านี้ที่อยู่ในรูปของ CosTradingRepos::TypeStruct มาจากการเรียกใช้

CosTradingRepos::ServiceTypeRepository::describe_type(

in CosTrading::ServiceTypeName name)

โดยแต่ละลักษณะประจำจะเป็นดังนี้

- Id = CosTradingRepos::TypeStruct::if_name
- Name = CosTrading::ServiceTypeName name
- Masked = CosTradingRepos::masked

BaseServiceTypes - ชื่อชนิดของบริการที่ชนิดของบริการนี้ได้รับการสืบทอดมาโดยตรง จะได้จาก CosTradingRepos::TypeStruct::super_types โดยจะมีลักษณะเป็นลำดับของสายอักขระที่เป็นชื่อของชนิดของบริการ สำหรับชนิดของบริการที่สืบทอดมาทางอ้อม และโครงสร้างสามารถหาได้โดยการวนซ้ำเรียก describe_type() ไปยังรายการชนิดของบริการที่สืบทอดมาโดยตรง

Property - คุณสมบัติของชนิดของบริการจะอยู่ในรูปของ CosTradingRepos::PropStructSeq โดยสามารถนำมาจาก CosTradingRepos::TypeStruct::props และคุณสมบัติที่ได้รับการสืบทอดมาสามารถหาได้จากการวนซ้ำเพื่อเรียกใช้ describe_type() กับชนิดของบริการที่สืบทอดมาทั้งหมด โดยรายละเอียดภายในส่วนย่อย Property สามารถนำมาจาก

- Name = CosTradingRepos::PropStruct::name
- Type = CosTradingRepos::PropStruct::value_type
- Mode = CosTradingRepos::PropStruct::mode

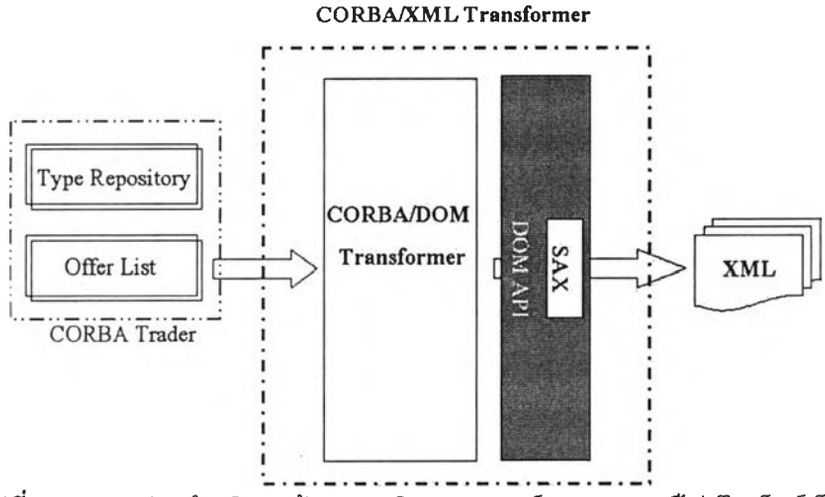
จากคำอธิบายชนิดของบริการในเทรดเดอร์ เมื่อผ่านกฎในการแปลงที่กำหนดแล้วผลลัพธ์ที่ได้จะมีลักษณะดังตัวอย่างต่อไปนี้

คำอธิบายชนิดของบริการ	เอ็กซ์เอ็มแอล

service BankService {
interface IDL:BankService:1.0
 	
property string BankName;
readonly property long ReservedFund;
};
	<TraderServiceType Id = "IDL:BankService:1.0"
	Name = "BankService"
	Masked = "NO" >
	<Property Name = "BankName"
	Type = "string"
	Mode = "NORMAL" />
	<Property Name = "ReservedFund"
	Type = "long"
	Mode = "READONLY" />
	</TraderServiceType>

3.2 การแปลงคำอธิบายข้อเสนอบริการจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล

การแปลงข้อมูลจากเทรดเดอร์เพื่อสร้างคำอธิบายข้อเสนอบริการนั้นจะเป็นการสร้างวัตถุดีโอเอ็มขึ้นมาจากคำอธิบายข้อเสนอบริการที่จัดเก็บคุณสมบัติของตัวบริการ โดยคำอธิบายข้อเสนอบริการที่เป็นเอกสารเอ็กซ์เอ็มแอลนี้จะต้องเป็นไปตามดีทีดีของคำอธิบายข้อเสนอบริการในหัวข้อ 3.2.1 จากนั้นจึงสร้างเอกสารเอ็กซ์เอ็มแอลผ่านวัตถุดีโอเอ็มดังแสดงในรูปที่ 3.2



รูปที่ 3.2 การแปลงคำอธิบายข้อเสนอบริการจากคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอล

3.2.1 ดีทีดีของคำอธิบายข้อเสนอบริการ

ดีทีดีของคำอธิบายข้อเสนอบริการเป็นตัวกำหนดไวยากรณ์ของเอกสารเอ็กซ์เอ็มแอลที่อธิบายข้อเสนอบริการ โดยดีทีดีของคำอธิบายข้อเสนอบริการจะเป็นไปดังตารางที่ 3.2

ตารางที่ 3.2 ดีทีดีของคำอธิบายข้อเสนอบริการ

```
1  <!ELEMENT ServiceOfferDescription (OfferType, Property*, ObjectReference)>
2  <!ELEMENT OfferType EMPTY>
3  <!ATTLIST  OfferType    Name      CDATA      #REQUIRED>
4  <!ELEMENT Property (DynamicPropEval, ExtraInfo)?>
5  <!ATTLIST  Property     Name      CDATA      #REQUIRED
6                          Value     CDATA      #IMPLIED>
7  <!ELEMENT DynamicPropEval (#PCDATA)>
8  <!ATTLIST DynamicPropEval ReturnType CDATA    #REQUIRED>
9  <!ELEMENT ExtraInfo EMPTY>
10 <!ATTLIST ExtraInfo     Type      CDATA      #REQUIRED
11                         Value     CDATA      #REQUIRED>
12 <!ELEMENT ObjectReference (#PCDATA)>
```

จากตารางที่ 3.2 สามารถสรุปข้อกำหนดในการสร้างเอกสารเอ็กซ์เอ็มแอลของคำอธิบายข้อเสนอบริการได้ดังนี้

บรรทัดที่	คำอธิบาย
1	รากของเอกสารคือ ServiceOfferDescription ที่ประกอบไปด้วยส่วนย่อยภายในอีกสามส่วนคือ • OfferType เป็นตัวระบุชื่อของชนิดของบริการของข้อเสนอบริการนี้ โดยส่วนย่อยนี้จำเป็นต้องมีในทุกเอกสารเอ็กซ์เอ็มแอลที่อธิบายข้อเสนอบริการ • Property เป็นส่วนอธิบายคุณสมบัติของข้อเสนอบริการ • ObjectReference เป็นตัวแสดงที่อยู่ของข้อเสนอบริการที่เรียกว่าข้อมูลในการอ้างถึงวัตถุ ในรูปแบบของไอโออาร์ (Interoperable Object References (IOR)) โดยส่วนย่อยนี้จำเป็นต้องมีในทุกเอกสารเอ็กซ์เอ็มแอลที่อธิบายข้อเสนอบริการ
2 – 3	ส่วนย่อย OfferType เป็นส่วนแสดงชนิดของบริการของข้อเสนอบริการนี้ โดยมีลักษณะประจำเป็นชื่อของชนิดของบริการที่อยู่ภายในเทรดเดอร์ • Name ชื่อของชนิดของบริการ
4 – 6	ส่วนย่อย Property เป็นส่วนอธิบายคุณสมบัติของข้อเสนอบริการนี้ อาจมีส่วนย่อยภายในเป็น DynamicPropEval หรือ ExtraInfo โดยประกอบไปด้วยลักษณะประจำคือ • Name ชื่อของคุณสมบัติ โดยจะต้องสอดคล้องกับที่ระบุในคำอธิบายชนิดของบริการ • Value ค่าของคุณสมบัติที่มีหรือไม่ก็ได้ โดยถ้ามีค่าหมายถึงเป็นคุณสมบัติแบบสถิตย์ แต่ถ้าไม่มีค่าหมายถึงเป็นคุณสมบัติแบบพลวัต ซึ่งส่วนย่อย Property จะต้องมีส่วนย่อยภายในคือ DynamicPropEval และ ExtraInfo ด้วย
7 – 8	ส่วนย่อย DynamicPropEval เป็นส่วนย่อยภายใน Property เมื่อคุณสมบัติดังกล่าวเป็นแบบพลวัต โดยจะแสดงข้อมูลที่อยู่ของตัวประมวลคุณสมบัติในรูปของไอโออาร์ที่เป็นโหนดข้อความ และมีลักษณะประจำคือ • ReturnType เป็นชนิดของค่าของคุณสมบัติแบบพลวัตที่จะทำการส่งกลับเมื่อมีการเรียกการประมวลผลคุณสมบัติแบบพลวัต โดยมีขอบเขตชนิดของค่าคุณสมบัติเช่นเดียวกับในคำอธิบายชนิดของบริการ
9 – 11	ส่วนย่อย ExtraInfo เป็นส่วนย่อยภายใน Property ที่เป็นคุณสมบัติแบบพลวัต โดยจะแสดงข้อมูลเพิ่มเติมให้กับคุณสมบัติ มีลักษณะประจำสองอย่างสำหรับการสร้างตัวแปรแบบ any คือ • Type เป็นชนิดของข้อมูลเพิ่มเติมที่มีขอบเขตเช่นเดียวกับคำอธิบายชนิดของบริการ • Value เป็นค่าของข้อมูลเพิ่มเติม
12	ส่วนย่อย ObjectReference เป็นตัวแสดงที่อยู่ของข้อเสนอบริการในรูปไอโออาร์ โดยที่อยู่นี้จะเก็บอยู่ในรูปของโหนดข้อความภายในส่วนย่อย

3.2.2 การแปลงคำอธิบายข้อเสนอบริการจากคอร์บาเทรดเดอร์

ในการแปลงคำอธิบายข้อเสนอบริการจากคอร์บาเทรดเดอร์ จะทำได้โดยการเรียกใช้ตัวกระทำการต่างๆ ที่กำหนดไว้ในข้อกำหนดของบริการเทรดเดอร์เพื่อดึงข้อมูลข้อเสนอบริการมาใช้ในการแปลง ทั้งนี้จะมีกฎในการนำคำอธิบายข้อเสนอบริการมาสร้างเป็นวัตถุดีโอเอ็มดังนี้คือ

OfferType - ส่วนย่อย OfferType ซึ่งจะประกอบไปด้วยชื่อของชนิดของบริการ โดยลักษณะประจำ Name นี้สามารถหาได้จากสามรูปแบบ (ตามวิธีการร้องขอการแปลงดังจะกล่าวใน 3.5) ดังนี้คือ

- การแปลงโดยระบุชนิดของบริการ - ชื่อชนิดของบริการจะเป็นไปตามที่ระบุ
- การแปลงโดยระบุตัวระบุของข้อเสนอบริการ (Offer Identifier) - ชื่อชนิดของบริการสามารถหาได้จากการเรียก CORBA::CosTrading::Register::describe(OfferId) โดยผลลัพธ์ที่ได้จะอยู่ในรูปของ CORBA::CosTrading::Register::OfferInfo ซึ่งชื่อของชนิดของบริการคือ CORBA::CosTrading::Register::OfferInfo::type
- การแปลงข้อเสนอบริการทั้งหมด - ชื่อชนิดของบริการทั้งหมดสามารถหาได้จากการเรียก CORBA::CosTradingRepos::ServiceTypeRepository::list_types(SpecifiedServiceTypes)
  เมื่อ SpecifiedServiceTypes คือชนิดของยูเนียน (union) ที่สามารถระบุได้ว่าจะนำชนิดของบริการทั้งหมด (all) หรือชนิดของบริการตั้งแต่หมายเลขอินคาร์เนชัน (Incarnation number) ที่ระบุเป็นต้นไป โดยที่ SpecifiedServiceTypes จะต้องระบุว่าเป็นชนิดของบริการทั้งหมด โดยผลลัพธ์ที่ได้จะอยู่ในรูปของ ServiceTypeNameSeq ซึ่งชื่อในลำดับคือชื่อชนิดของบริการทั้งหมดที่มีอยู่ในเทรดเดอร์

Property - ส่วนย่อย Property นี้สามารถหาได้จาก CORBA::CosTrading::Register::OfferInfo ที่ได้จากการหาในส่วนย่อย OfferType โดยจะได้ CORBA::CosTrading::Property[i] ที่เป็นชุดของคุณสมบัติ เมื่อ i เป็นดัชนีของคุณสมบัติ โดยที่มีลักษณะประจำดังนี้

- Name = CosTrading::Property[i]::name
- Value = CosTrading::Property[i]::value โดยถ้า value เป็นคุณสมบัติแบบพลวัต จะไม่มีลักษณะประจำนี้

DynamicPropEval – หากส่วนย่อย Property ไม่มีลักษณะประจำ Value ส่วนย่อย DynamicPropEval นี้จะเป็นส่วนย่อยภายในที่ระบุข้อมูลเกี่ยวกับคุณสมบัติแบบพลวัต ซึ่งจะสามารถหาได้จาก CosTrading::Property[i]::value โดยสามารถเปลี่ยนรูปให้เป็น CosTradingDynamic::DynamicProp ได้ ลักษณะประจำแต่ละตัวจะต้องเป็นดังนี้

- ReturnType = CosTradingDynamic::DynamicProp::returned_type

และมีโหนดข้อความซึ่งเป็นที่อยู่ของตัวประมวลคุณสมบัติในรูปของไอออาร์ที่ได้จากการแปลง CosTradingDynamic::DynamicProp::eval_if ให้เป็นชุดอักขระ

ExtraInfo - ส่วนย่อยนี้จะเป็นข้อมูลเพิ่มเติมสำหรับคุณสมบัติแบบพลวัต โดยจะเป็นข้อมูลที่หาได้จาก CosTradingDynamic::DynamicProp::extra_info โดยแต่ละลักษณะประจำจะเป็นดังนี้

- Type = CosTradingDynamic::DynamicProp::extra_info::type()
- Value = CosTradingDynamic::DynamicProp::extra_info::type()

ObjectReference – ตัวระบุที่อยู่ของข้อเสนอบริการสามารถแปลงเป็นชุดอักขระได้จาก CORBA::CosTrading::Register::OfferInfo::reference ซึ่งได้จากการหาส่วนย่อย OfferType

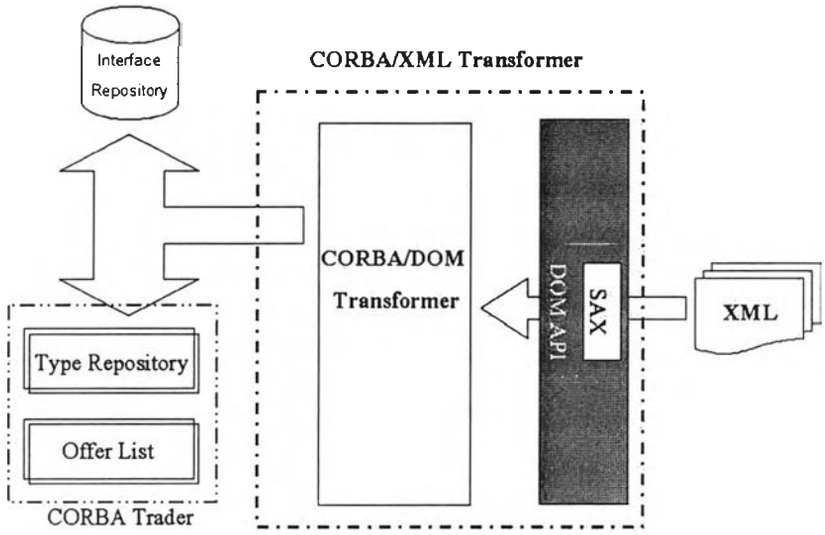
จากคำอธิบายข้อเสนอบริการในเทรดเดอร์ เมื่อผ่านกฎในการแปลงที่กำหนดแล้วผลลัพธ์ที่ได้จะมีลักษณะดังตัวอย่างต่อไปนี้

คำอธิบายข้อเสนอบริการ	เอ็กซ์เอ็มแอล
Type: BankService	<OfferType Name = "BankService" />
Properties:	
BankName = 'Chula'	<Property Name = "BankName" Value = "Chula" />
ReservedFund = {Dynamic}	<Property Name = "ReservedFund">
eval_if = IOR:0001211…	<DynamicPropEval ReturnType = "long" >
returned_type = long	IOR:0001211…
extra_info = 'Reserved fund	</DynamicPropEval>
changes everyday'	<ExtraInfo Type = "string"
	Value = "Reserved fund changes everyday" /<
Reference:	</Property>
IOR:115944C21……	<ObjectReference>IOR:115944C21……</ObjectReference>

3.3 การแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์

ในการแปลงคำอธิบายชนิดของบริการจากรูปแบบเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์จะมีส่วนย่อยสองส่วนเช่นเดียวกับการแปลงจากรูปแบบของคอร์บาเทรดเดอร์ไปเป็นเอ็กซ์เอ็มแอลนั่นคือการแปลงเอกสารบางส่วนให้อยู่ในรูปแบบของคลังส่วนต่อประสาน และบางส่วนให้อยู่ในรูปแบบของคลังชนิดของบริการ สำหรับการแปลงจากรูปแบบเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์นี้จะเป็นการอ่านเอกสารเอ็กซ์เอ็มแอลเพื่อสร้างเป็นวัตถุดีโดเอ็ม จากนั้นจึงทำการดึงข้อมูลภายในวัตถุดีโอเอ็มมาสร้างให้อยู่ในรูปแบบของคอร์บา ดังแสดงในรูปที่ 3.3 การดึงข้อมูลผ่านวัตถุดีโอเอ็มสามารถทำได้ดังที่กล่าวไว้ในบทนำของบทที่ 3

และการแปลงให้อยู่ในรูปแบบของคลังส่วนต่อประสาน และคลังจัดเก็บชนิดของบริการจะกล่าวถึงในหัวข้อ 3.3.1 และ 3.3.2 ตามลำดับ



รูปที่ 3.3 การแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์

ในการแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์นั้นจะทำได้ก็ต่อเมื่อส่วนต่อประสาน และชนิดของบริการที่คำอธิบายชนิดของบริการที่สืบทอดมาจะต้องมีอยู่แล้วภายในคลังส่วนต่อประสาน และคลังชนิดของบริการเสียก่อน

3.3.1 การแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคลังส่วนต่อประสาน

สำหรับการแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคลังส่วนต่อประสานจะทำได้โดยการเรียกใช้ตัวกระทำการต่างๆ ที่กำหนดไว้ในข้อกำหนดของคลังส่วนต่อประสานเพื่อสร้างวัตถุที่แสดงนิยามของส่วนต่อประสาน โดยในแต่ละส่วนย่อยจะกำหนดการแปลงดังนี้

BaseInterfaces - ส่วนย่อย BaseInterface ภายในส่วนย่อย BaseInterfaces นั้นจะใช้ในการตรวจสอบว่าส่วนต่อประสานแม่ซึ่งส่วนต่อประสานที่ต้องการทำการแปลงนี้สืบทอดมามีรายละเอียดอยู่ภายในคลังส่วนต่อประสานแล้วหรือไม่ จึงจะทำการแปลงและเพิ่มนิยามส่วนต่อประสานนี้ลงในคลังส่วนต่อประสาน การตรวจสอบทำได้โดยการเรียก CORBA::Repository::lookup_id(RepositoryId) เมื่อ RepositoryId คือตัวระบุของส่วนต่อประสานแม่ ซึ่งก็คือลักษณะเฉพาะ Dest ของส่วนย่อย Link ที่มีลักษณะเฉพาะ Source เป็นตัวระบุของส่วนต่อประสานนี้ ผลลัพธ์ของการค้นหาจะต้องหา RepositoryId นี้พบ จึงจะสามารถทำการแปลงและเพิ่มนิยามส่วนต่อประสานลูกลงในคลังส่วนต่อประสานได้

Interface - จะเป็นการสร้าง InterfaceDef โดยใช้ข้อมูลจากลักษณะประจำโดยผ่านไอดีแอลดังนี้คือ

- สร้าง ModuleDef จาก Id ให้เป็นไปตามลักษณะการบรรจุเชิงตรรกะโดยการเรียก CORBA::Container::create_module(id, name, version) เมื่อ Container คือตัวบรรจุล่าสุดที่ต้องการสร้าง ModuleDef
- สร้าง InterfaceDef โดยการเรียก CORBA::Container::create_interface(id, name, version, baseInterfaces) เมื่อ Container คือตัวบรรจุล่าสุดที่ต้องการสร้าง InterfaceDef และ id, name และ version คือลักษณะประจำของส่วนย่อย Interface Id, Name และ Version ตามลำดับ ส่วน baseInterfaces คือ InterfaceDef ของส่วนต่อประสานที่ได้จากการเรียก CORBA::Repository::lookup_id(RepositoryId) ในส่วนย่อย BaseInterfaces ที่กล่าวมาข้างต้น

Constant - จะต้องสร้าง ConstantDef ขึ้นภายใน InterfaceDef ที่ได้จากส่วนย่อย Interface โดยสร้าง ConstantDef จากการเรียก CORBA::InterfaceDef::create_constant(id, name, version, type, value) โดย id, name, version, type และ value คือ ลักษณะประจำของส่วนย่อย Constant Id, Name, Version, Type และ Value ตามลำดับ ซึ่ง ConstantDef ที่จะสร้างขึ้นจะพิจารณาเฉพาะจากส่วนย่อย Constant ที่ไม่มีค่าลักษณะประจำ Derived หรือมีค่าเป็น NO

Attribute - เช่นเดียวกับ Constant คือจะต้องสร้าง AttributeDef ขึ้นภายใน InterfaceDef ที่ได้จากส่วนย่อย Interface โดยสร้าง AttributeDef จากการเรียก CORBA::InterfaceDef::create_attribute(id, name, version, type, mode) โดย id, name, version, type และ mode คือลักษณะประจำของส่วนย่อย Attribute Id, Name, Version, Type และ Mode ตามลำดับ ซึ่ง AttributeDef ที่จะสร้างขึ้นจะพิจารณาเฉพาะจากส่วนย่อย Attribute ที่ไม่มีค่าลักษณะประจำ Derived หรือมีค่าเป็น NO

Operation - การสร้าง OperationDef นั้นจะต้องสร้างขึ้นภายใน InterfaceDef ที่ได้จากส่วนย่อย Interface เช่นเดียวกัน แต่การสร้าง OperationDef จะต้องสร้างรายการของ ParameterDescription, ExceptionDef และ Context ขึ้นมาก่อน โดยจะต้องสร้างขึ้นเป็นลำดับดังนี้

- ParameterDescriptionSeq - โดยการสร้างชุดของ CORBA::ParameterDescription ซึ่งมีสมาชิกเป็น name, type, type_def และ mode โดย name จะได้จากลักษณะประจำ Name ส่วน type และ type_def จะได้จากลักษณะประจำ Type และ mode จะได้จากลักษณะประจำ Mode ของส่วนย่อย Parameter
- ExceptionDefSeq - สร้างขึ้นจากการเรียก CORBA::Repository::create_exception(id, name, version, members) เมื่อ id, name และ version คือลักษณะประจำ Id, Name และ

Version ในส่วนย่อย Exception ตามลำดับ และ members คือชุดของ StructMember ที่จะกล่าวถัดไป

- StructMemberSeq - เป็นชุดของ CORBA::StructMember ที่สร้างขึ้นจากส่วนย่อย Member โดย StructMember มีสมาชิกคือ name, type และ type_def ทั้งนี้ name ได้จากลักษณะประจำ Name ส่วน type และ type_def ได้จากลักษณะประจำ Type ในส่วนย่อย Member
- ContextIdSeq - เป็นชุดของคอนเท็กซ์ของตัวกระทำการนี้ ซึ่งสามารถหาได้จากโหนดข้อความของส่วนย่อย Context

จากนั้นจึงสร้าง OperationDef โดยการเรียก CORBA::InterfaceDef::create_operation(id, name, version, result, mode, params, exceptions, contexts) โดยที่

- id คือลักษณะเฉพาะ Id จากส่วนย่อย Operation
- name คือลักษณะเฉพาะ Name จากส่วนย่อย Operation
- version คือลักษณะเฉพาะ Version จากส่วนย่อย Operation
- result คือลักษณะเฉพาะ Type จากส่วนย่อย Operation
- mode คือลักษณะเฉพาะ Mode จากส่วนย่อย Operation
- params คือ ParameterDescriptionSeq ที่สร้างขึ้นจากที่กล่าวมาแล้ว
- exceptions คือ ExceptionDefSeq ที่สร้างขึ้นจากที่กล่าวมาแล้ว
- context คือ ContextIdSeq ที่สร้างขึ้นจากที่กล่าวมาแล้ว

เช่นเดียวกับส่วนย่อย Constant และ Attribute ส่วนย่อย Operation ที่จะสร้าง OperationDef จะพิจารณาเฉพาะกรณีที่ไม่มีลักษณะประจำ Derived หรือมีค่าเป็น NO

3.3.2 การแปลงคำอธิบายชนิดของบริการจากเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคลังชนิดของบริการ

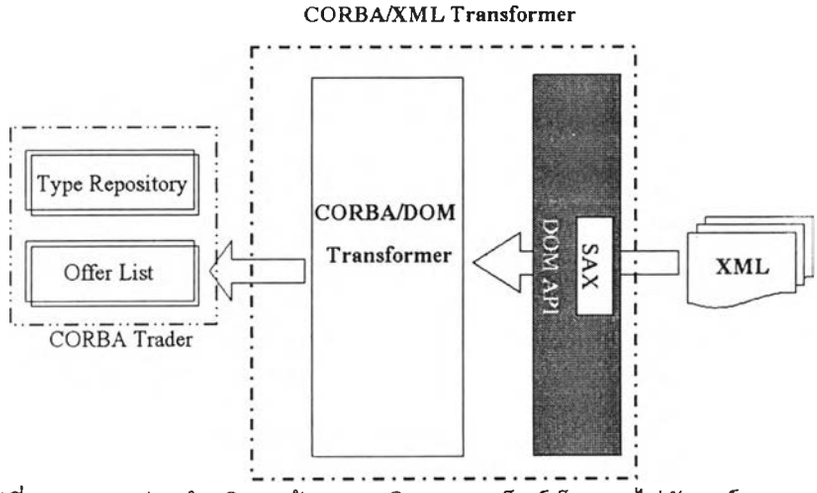
คำอธิบายชนิดของบริการที่มาจากคลังชนิดของบริการคือส่วนย่อย TraderServiceType โดยส่วนย่อยนี้จะสามารถนำมาสร้างชนิดของบริการในคลังชนิดของบริการโดยการเรียกตัวกระทำการภายในส่วนต่อประสานหลักที่อยู่ในรูปของไอดีแอลที่กำหนดโดยโอเอ็มจีคือ CORBA::CosTradingRepos::ServiceTypeRepository::add_type(name, if_name, props, super_types) โดยแต่ละพารามิเตอร์จะได้จากข้อมูลในเอ็กซ์เอ็มแอลดังนี้

- name คือชื่อของชนิดของบริการจะได้จาก ลักษณะประจำ Name ของส่วนย่อย TraderServiceType
- if_name คือชื่อตัวระบุของส่วนต่อประสานของชนิดของบริการนี้ในคลังจัดเก็บส่วนต่อประสานจะได้จากลักษณะประจำ Id ของส่วนย่อย TraderServiceType

- props คือลำดับของคุณสมบัติของชนิดของบริการ (CORBA::CosTradingRepos::PropStructSeq) โดยในแต่ละคุณสมบัติจะสร้างขึ้นด้วยสมาชิกดังนี้
  - name คือชื่อของคุณสมบัติ นำมาจากลักษณะประจำ Name ของส่วนย่อย Property
  - value_type คือชนิดของคุณสมบัติ นำมาจากลักษณะประจำ Type ของส่วนย่อย Property
  - mode คือสภาวะของคุณสมบัติ นำมาจากลักษณะประจำ Mode ของส่วนย่อย Property ภายในเอกสารเอ็กซ์เอ็มแอลของคำอธิบายชนิดของบริการจะประกอบไปด้วยคุณสมบัติที่ได้รับการสืบทอดมา และคุณสมบัติโดยตรง แต่การเพิ่มชนิดของบริการเข้าไปในคลังชนิดของบริการนั้นคุณสมบัติที่จะระบุภายใต้ชนิดของบริการนี้คือคุณสมบัติโดยตรงเท่านั้น ดังนั้นส่วนย่อย Property ที่จะพิจารณาคือส่วนย่อย Property ที่ไม่มีลักษณะประจำ Derived หรือมีค่าเป็น NO
- super_types คือชุดของชนิดของบริการที่ได้รับการสืบทอดมาโดยตรง นั่นคือสามารถนำมาจากส่วนย่อยภายใน BaseServiceTypes อันได้แก่ส่วนย่อย Link ที่มีลักษณะประจำ Source เป็นชื่อของชนิดของบริการนี้ และลักษณะประจำ Dest เป็นชนิดของบริการที่ได้รับการสืบทอดมาโดยตรง

3.4 การแปลงคำอธิบายข้อเสนอบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์

คำอธิบายที่อยู่ในรูปแบบของเอกสารเอ็กซ์เอ็มแอลจะถูกแปลงให้อยู่ในรูปแบบของคอร์บาเทรดเดอร์โดยการโฆษณาข้อเสนอบริการเพิ่มเข้าไปตามชนิดของบริการที่มีอยู่แล้วดังรูปที่ 3.4



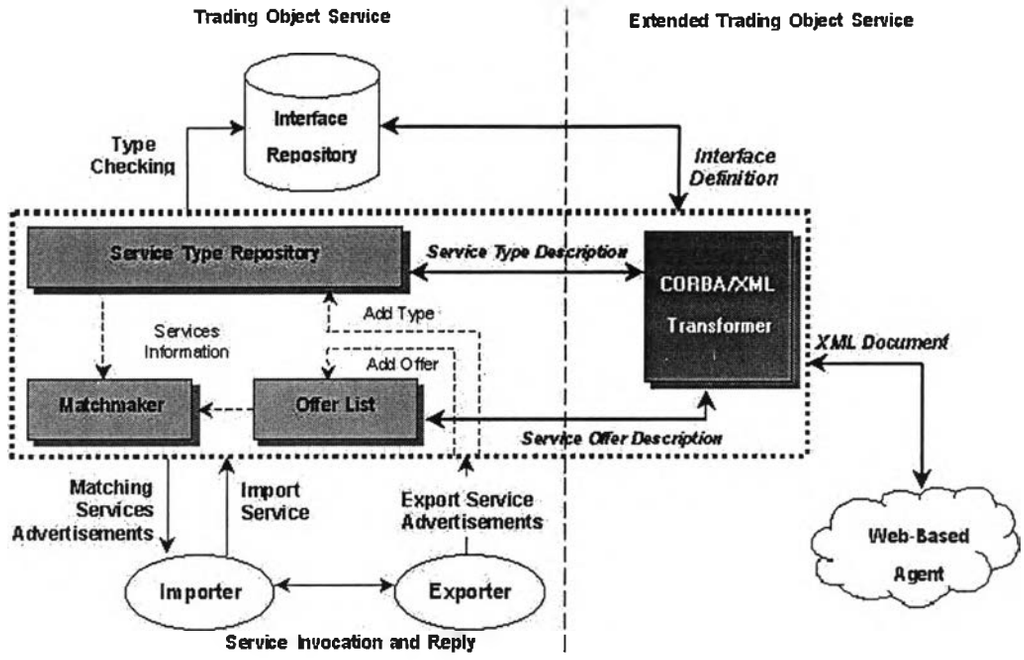
รูปที่ 3.4 การแปลงคำอธิบายข้อเสนอบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์

การแปลงคำอธิบายข้อเสนอบริการจากเอ็กซ์เอ็มแอลไปยังคอร์บาเทรดเดอร์จะผ่านตัวกระทำการภายในส่วนต่อประสานของโอเอ็มจีคือ CORBA::CosTrading::Register::export(reference, type, properties) โดยในแต่ละพารามิเตอร์คือ

- reference คือที่อยู่ของข้อเสนอบริการในรูปแบบของข้อมูลในการอ้างถึงวัตถุ จะได้จากการแปลงโหนดข้อความของส่วนย่อย ObjectReference
- type คือชนิดของบริการที่จะทำการโฆษณา จะได้จากลักษณะประจำ Name ในส่วนย่อย OfferType
- properties คือชุดของคุณสมบัติที่ประกอบไปด้วยชื่อ และค่า นั่นคือ CORBA::CosTrading::Property ที่มีสมาชิกคือ name และ value โดย name และ value คือ ลักษณะประจำ Name และ Value ในส่วนย่อย Property ตามลำดับ สำหรับคุณสมบัติแบบพลวัต ซึ่งในเอกสารเอ็กซ์เอ็มแอลจะไม่มีลักษณะประจำ Value ในส่วนย่อย Property ดังนั้นสมาชิก value ใน CORBA::CosTrading::Property จะมีค่าของ CORBA::CosTradingDynamic::DynamicProp ซึ่งมีสมาชิกคือ
  - eval_if คือที่อยู่ของตัวประมวลผลคุณสมบัติแบบพลวัต CORBA::CosTradingDynamic::DynamicPropEval ที่ได้มาจากโหนดข้อความของส่วนย่อย DynamicPropEval
  - returned_type คือชนิดของค่าของคุณสมบัติที่ได้มาจากลักษณะประจำ ReturnType ของส่วนย่อย DynamicPropEval
  - extra_info คือข้อมูลเพิ่มเติมในรูปของ any ที่มีชนิดคือลักษณะประจำ Type และค่าคือลักษณะประจำ Value ในส่วนย่อย ExtraInfo ตามลำดับ

3.5 การเพิ่มขยายบริการคอร์บาเทรดเดอร์ให้มีความสามารถในการแปลงคำอธิบายบริการระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล

บริการเทรดเดอร์ตามมาตรฐานของโอเอ็มจีที่อธิบายอยู่ในรูปของไอดีแอลสามารถเพิ่มขยายเพื่อให้สามารถรองรับการแปลงคำอธิบายบริการระหว่างคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอลได้โดยการเพิ่มขยายในส่วนของ CosTrading ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 บริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย

จากรูปที่ 3.5 บริการเทรดเดอร์ตามมาตรฐานของโอเอ็มจีซึ่งอยู่ทางซ้ายของเส้นประ จะถูกเพิ่มขยายด้วยส่วนการแปลงคำอธิบายบริการ (CORBA/XML Transformer) ซึ่งอยู่ทางด้านขวาเพื่อรองรับการแปลงคำอธิบายบริการระหว่างคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอล ส่วนการแปลงคำอธิบายจะทำหน้าที่ในการนำข้อมูลจากคอร์บาเทรดเดอร์ดังที่กล่าวมาแล้วในหัวข้อ 3.3 และ 3.4 มาแปลงเป็นเอกสารเอ็กซ์เอ็มแอล โดยเอกสารที่ได้สามารถนำไปใช้กับตัวแทน (Agent) อื่นๆ บนเว็บเพื่อใช้ประโยชน์ในการอธิบายบริการ หรือการค้นหาบริการต่อไป และนำเอกสารเอ็กซ์เอ็มแอลมาแปลงให้อยู่ในรูปแบบของคอร์บาเทรดเดอร์ ในส่วนของการเพิ่มขยายบริการเทรดเดอร์ตามมาตรฐานของโอเอ็มจีต้องมีการเพิ่มเติมในส่วนของไอดีแอลที่อธิบายบริการเทรดเดอร์ (CosTrading.idl) ดังนี้

1. การเพิ่มเติมการประกาศส่วนต่อประสานดังนี้

   interface CorbaXmlTransformer;

2. การเพิ่มเติมลักษณะประจำในส่วนต่อประสาน TraderComponents เพื่อระบุจุดเข้าถึง (Access Point) ไปยังส่วนต่อประสาน CorbaXmlTransformer ดังนี้

   readonly attribute CorbaXmlTransformer transform_if;

3. การเพิ่มเติมส่วนต่อประสาน CorbaXmlTransformer เพื่อรองรับการแปลงคำอธิบายดังนี้

```
interface CorbaXmlTransformer {
    typedef Istring Identifier;
    exception InvalidXmlFileLocation { string location; };
    exception InvalidXmlDocument { };
    exception UnknownInterface { Identifier if_name; };
    exception ServiceTypeExists { ServiceTypeName name; };
```

```
        long transform_all_type();
        boolean transform_type(in ServiceTypeName type)
                raises (UnknownServiceType, IllegalServiceType);
        long transform_all_offer(out OfferIdSeq ids);
        long transform_offer(in ServiceTypeName type, out OfferIdSeq ids)
                raises (UnknownServiceType, IllegalServiceType, NotImplemented);
        boolean transform_offer_id(in OfferId id)
                raises (UnknownOfferId, IllegalOfferId);
        OfferId transform_xml(in string file)
                raises (NotImplemented, InvalidXmlFileLocation, InvalidXmlDocument,
                        UnknownInterface, UnknownServiceType, ServiceTypeExists);
    };
```

โดยในแต่ละส่วนของตัวกระทำการสามารถอธิบายได้ดังนี้คือ

transform_all_type() - ตัวกระทำการนี้ใช้เพื่อการแปลงคำอธิบายชนิดของบริการทั้งหมดที่อยู่ภายในเทรดเดอร์ให้เป็นเอ็กซ์เอ็มแอล โดยผลลัพธ์ที่ได้จะเป็นเอกสารเอ็กซ์เอ็มแอลของคำอธิบายชนิดของบริการทั้งหมดที่อยู่ในรูปแฟ้มข้อมูลภายในไดเร็กทอรี่จัดเก็บฐานข้อมูลที่ถูกกำหนดโดยผู้เริ่มใช้งานเทรดเดอร์ และมีการคืนกลับค่าเป็น long ที่เป็นจำนวนของชนิดของบริการที่ทำการแปลงสำเร็จทั้งหมด ชนิดของบริการที่มีข้อผิดพลาด หรือข้อยกเว้นจะไม่ถูกแปลง และไม่ถูกนับรวมในค่าที่คืนกลับมา

transform_type() - ตัวกระทำการนี้ใช้เพื่อการแปลงคำอธิบายชนิดของบริการให้เป็นเอ็กซ์เอ็มแอลโดยกำหนดชื่อชนิดของบริการที่ต้องการแปลงเป็นพารามิเตอร์ ผลลัพธ์ที่ได้คือเอกสารเอ็กซ์เอ็มแอลของคำอธิบายชนิดของบริการตามชื่อของชนิดของบริการที่กำหนดในรูปแฟ้มข้อมูลภายในไดเร็กทอรี่จัดเก็บฐานข้อมูลที่ถูกกำหนดโดยผู้เริ่มใช้งานเทรดเดอร์ โดยมีข้อยกเว้นดังนี้คือ

- เมื่อชื่อของชนิดของบริการที่กำหนดในพารามิเตอร์ไม่มีอยู่ในคลังชนิดของบริการ จะเกิดข้อยกเว้น CORBA::CosTrading::UnknownServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่ไม่มีในคลังชนิดของบริการ
- เมื่อชื่อของชนิดของบริการที่กำหนดในพารามิเตอร์ไม่เป็นไปตามข้อกำหนดใน [2] สำหรับ Service Type Model จะเกิดข้อยกเว้น CORBA::CosTrading::IllegalServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่ไม่เป็นไปตามข้อกำหนด

ค่าที่มีการคืนกลับจะมีชนิดเป็น boolean ที่บอกสถานะเป็นจริงเมื่อการแปลงสำเร็จ หรือเป็นเท็จเมื่อเกิดข้อผิดพลาดภายในที่นอกจากข้อยกเว้น

transform_all_offer() - ตัวกระทำการนี้ใช้เพื่อการแปลงคำอธิบายข้อเสนอบริการทั้งหมดที่มีอยู่ภายในเทรดเดอร์ให้เป็นเอ็กซ์เอ็มแอล ผลลัพธ์ที่ได้คือเอกสารเอ็กซ์เอ็มแอลของคำอธิบายข้อเสนอบริการทั้งหมดในรูปแฟ้มข้อมูลภายในไดเร็กทอรี่จัดเก็บฐานข้อมูลที่ถูกกำหนดโดยผู้เริ่มใช้งานเทรดเดอร์ และมีการคืนกลับค่าเป็น long ที่เป็นจำนวนเอกสารเอ็กซ์เอ็มแอลของข้อเสนอบริการที่แปลงสำเร็จ และคืนค่าพารามิเตอร์ ids ที่มีสภาวะเป็น out โดยจะเป็นชุดของ CORBA::CosTrading::OfferId ที่ทำการแปลงสำเร็จทั้งหมด ข้อเสนอบริการที่มีข้อผิดพลาด หรือข้อยกเว้นในการแปลงจะไม่ถูกนับรวมในค่าที่คืนกลับมา และพารามิเตอร์ ids

transform_offer() - ตัวกระทำการนี้ใช้เพื่อการแปลงคำอธิบายข้อเสนอบริการตามชนิดของบริการ โดยสามารถระบุชนิดของบริการที่ต้องการเป็นพารามิเตอร์ ผลลัพธ์ที่ได้คือเอกสารเอ็กซ์เอ็มแอลของข้อเสนอบริการตามชนิดของบริการที่กำหนด รวมถึงข้อเสนอบริการของชนิดของบริการที่เป็นลูกของชนิดของบริการที่ระบุ ที่อยู่ในรูปแฟ้มข้อมูลภายในไดเร็กทอรี่จัดเก็บฐานข้อมูลที่ถูกกำหนดโดยผู้เริ่มใช้งานเทรดเดอร์ โดยมีข้อยกเว้นดังนี้คือ

- เมื่อชื่อของชนิดของบริการที่กำหนดในพารามิเตอร์ไม่มีอยู่ในคลังชนิดของบริการ จะเกิดข้อยกเว้น CORBA::CosTrading::UnknownServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่ไม่มีในคลังชนิดของบริการ
- เมื่อชื่อของชนิดของบริการที่กำหนดในพารามิเตอร์ไม่เป็นไปตามข้อกำหนดใน [2] สำหรับ Service Type Model จะเกิดข้อยกเว้น CORBA::CosTrading::IllegalServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่ไม่เป็นไปตามข้อกำหนด
- เมื่อเทรดเดอร์ไม่สนับสนุนส่วนต่อประสาน CORBA::CosTrading::Register จะเกิดข้อยกเว้น CORBA::CosTrading::NotImplemented ขึ้น

ค่าที่มีการคืนกลับมีชนิดเป็น long ที่แสดงถึงจำนวนข้อเสนอบริการทั้งหมดที่มีการแปลงเป็นเอ็กซ์เอ็มแอลได้สำเร็จ โดยมีพารามิเตอร์ชนิด out คือ ids เป็นชุดของ CORBA::CosTrading::OfferId ของข้อเสนอบริการที่แปลงสำเร็จ ข้อเสนอบริการที่มีข้อผิดพลาด และข้อยกเว้นในการแปลงจะไม่ถูกนับรวมในค่าที่คืนกลับมา และค่าพารามิเตอร์ ids

transform_offer_id() - ตัวกระทำการนี้ใช้เพื่อการแปลงคำอธิบายข้อเสนอบริการให้เป็นเอกสารเอ็กซ์เอ็มแอลโดยกำหนดตัวระบุข้อเสนอบริการเป็นพารามิเตอร์ ผลลัพธ์ที่ได้คือเอกสารเอ็กซ์เอ็มแอลของข้อเสนอบริการหนึ่งตัวตามที่ระบุในรูปแฟ้มข้อมูลภายในไดเร็กทอรี่จัดเก็บฐานข้อมูลที่ถูกกำหนดโดยผู้เริ่มใช้งานเทรดเดอร์ โดยมีข้อยกเว้นดังนี้คือ

- เมื่อตัวระบุของข้อเสนอบริการที่กำหนดในพารามิเตอร์ไม่มีอยู่ในเทรดเดอร์ จะเกิดข้อยกเว้น CORBA::CosTrading::UnknownOfferId ขึ้น โดยมีสมาชิกคือชื่อของตัวระบุที่ไม่มีในเทรดเดอร์

- เมื่อตัวระบุของข้อเสนอบริการที่กำหนดในพารามิเตอร์ไม่เป็นไปตามข้อกำหนดใน [2] สำหรับ Offer Identifier จะเกิดข้อยกเว้น CORBA::CosTrading::IllegalOfferId ขึ้น โดยมีสมาชิกคือตัวระบุของข้อเสนอบริการที่ไม่เป็นไปตามข้อกำหนด

ค่าที่มีการคืนกลับมามีชนิดเป็น boolean โดยเมื่อมีค่าเป็นจริงแสดงว่าแปลงสำเร็จ หรือเป็นเท็จเมื่อมีข้อผิดพลาดนอกเหนือจากข้อยกเว้นเกิดขึ้น

transform_xml() - ตัวกระทำการนี้ใช้เพื่อการแปลงเอกสารเอ็กซ์เอ็มแอลให้อยู่ในรูปแบบของคอร์บาเทรดเดอร์ โดยพารามิเตอร์คือชื่อและที่อยู่ของเอกสารเอ็กซ์เอ็มแอลที่อาจอยู่ในรูปแบบของยูอาร์แอล หรือแฟ้มข้อมูลก็ได้ โดยผลลัพธ์ที่ได้จะเป็นการแปลงเอกสารเอ็กซ์เอ็มแอลให้เป็นชนิดของบริการ หรือข้อเสนอบริการซึ่ง (ขึ้นอยู่กับรากของเอกสารเอ็กซ์เอ็มแอล) ให้อยู่ในคอร์บาเทรดเดอร์ที่มีการเรียกใช้ตัวกระทำการนี้ ข้อยกเว้นที่อาจเกิดขึ้นจากตัวกระทำการนี้คือ

- เมื่อเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงประกอบไปด้วยชนิดที่ระบุโดยผู้ใช้ (User Defined Type) จะเกิดข้อยกเว้น CORBA::CosTrading::NotImplemented เนื่องจากยังไม่สามารถทำการแปลงเอกสารที่มีชนิดที่ระบุโดยผู้ใช้ได้
- เมื่อชื่อและที่อยู่ของเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงไม่อยู่ในรูปแฟ้มข้อมูล หรือยูอาร์แอลที่ถูกต้อง หรือไม่สามารถค้นหาและอ่านเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงนี้ได้ จะเกิดข้อยกเว้น CORBA::CosTrading::CorbaXmlTransformer::InvalidXmlFileLocation ขึ้น โดยมีสมาชิกคือชื่อและที่อยู่ของเอกสาร
- เมื่อเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงไม่เป็นเอกสารที่ถูกต้อง (Valid Document) ตามดีทีดีของคำอธิบายชนิดของบริการ หรือดีทีดีของคำอธิบายข้อเสนอบริการ จะทำให้เกิดข้อยกเว้น CORBA::CosTrading::CorbaXmlTransformer::InvalidXmlDocument ขึ้น
- ในการแปลงคำอธิบายชนิดของบริการจากเอกสารเอ็กซ์เอ็มแอล หากส่วนต่อประสานของชนิดของบริการนี้สืบทอดมาจากส่วนต่อประสานอื่น โดยที่ส่วนต่อประสานแม่นี้ยังไม่ได้รับการบันทึกลงในคลังส่วนต่อประสาน จะทำให้เกิดข้อยกเว้น CORBA::CosTrading::CorbaXmlTransformer::UnknownInterface ขึ้น โดยมีสมาชิกคือตัวระบุส่วนต่อประสานที่สืบทอดมาที่ไม่มีในคลังส่วนต่อประสาน
- ในการแปลงคำอธิบายชนิดของบริการจากเอกสารเอ็กซ์เอ็มแอล หากชนิดของบริการนี้สืบทอดมาจากชนิดของบริการอื่น โดยชนิดของบริการแม่นี้ยังไม่ได้รับการบันทึกลงในคลังชนิดของบริการ จะทำให้เกิดข้อยกเว้น CORBA::CosTrading::UnknownServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่สืบทอดมาที่ไม่มีในคลังชนิดของบริการ
- เมื่อเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงเป็นคำอธิบายข้อเสนอบริการ โดยชนิดของบริการของข้อเสนอบริการนี้ไม่มีอยู่ในคลังชนิดของบริการ จะทำให้เกิดข้อยกเว้น

CORBA::CosTrading::UnknownServiceType ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการของข้อเสนอบริการที่ไม่มีในคลังชนิดของบริการ

- เมื่อเอกสารเอ็กซ์เอ็มแอลที่จะทำการแปลงเป็นคำอธิบายชนิดของบริการ ซึ่งมีอยู่แล้วในคลังชนิดของบริการ จะทำให้เกิดข้อยกเว้น CORBA::CosTrading::CorbaXmlTransformer::ServiceTypeExists ขึ้น โดยมีสมาชิกคือชื่อชนิดของบริการที่มีอยู่แล้วในคลังชนิดของบริการ

ในส่วนการแปลงเอกสารเอ็กซ์เอ็มแอลชนิดของบริการโดยมีส่วนต่อประสานที่ได้รับการบันทึกไว้ในคลังส่วนต่อประสานแล้ว จะสามารถกระทำการแปลงชนิดของบริการได้โดยไม่เกิดข้อยกเว้นใดๆ แต่ผู้เรียกใช้งานต้องทำการตรวจสอบในคลังส่วนต่อประสานว่าส่วนต่อประสานที่ทำการแปลงนั้นเป็นส่วนประสานเดียวกันกับในเอกสารเอ็กซ์เอ็มแอลหรือไม่

ในบทนี้ที่ได้กล่าวมาแล้วเป็นการออกแบบข้อกำหนดลักษณะของเอกสารเอ็กซ์เอ็มแอลในรูปของดีทีดี กฎในการแปลงคำอธิบายชนิดของบริการและข้อเสนอบริการ รวมทั้งออกแบบการเพิ่มขยายเทรดเดอร์ โดยการออกแบบจะไม่ขึ้นกับภาษาโปรแกรม ระบบปฏิบัติการ และเทรดเดอร์ที่สร้างขึ้นต่างๆ กัน ดังนั้นแนวทางนี้สามารถนำไปพัฒนาเพื่อเพิ่มขยายความสามารถของเทรดเดอร์ใดๆ ให้สามารถทำการแปลงคำอธิบายบริการระหว่างรูปแบบของคอร์บาเทรดเดอร์กับเอ็กซ์เอ็มแอลได้

สำหรับในบทต่อไปจะได้กล่าวถึงการพัฒนาต้นแบบของบริการเทรดเดอร์ที่มีส่วนเพิ่มขยาย และโปรแกรมทดสอบการใช้งานของเทรดเดอร์ที่มีส่วนเพิ่มขยายเพื่อแปลงข้อมูลระหว่างรูปแบบของเทรดเดอร์กับเอกสารเอ็กซ์เอ็มแอล