

CORBA Inter-domain management

Paller Gábor

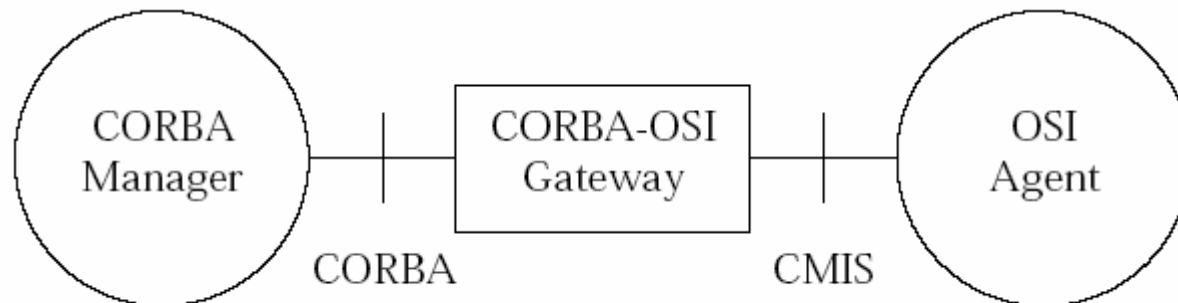
2004.11.28

JDIM

- Joint Inter-domain management
- OMG (korábbi nevén Open Group) és Network Management Forum (NMF) közös specifikáció
- Célja: különböző menedzsment technológiákat (legfontosabbak: OSI és SNMP) közös, CORBA alapú rendszerbe foglalni
- Motiváció: ekkor menedzsment rendszerek és ágensek egyaránt CORBA model szerint írhatók, nem kell a menedzsmenttechnológia változása szerint újraírni őket
- Motiváció2: TMN rendszereket CORBA alapon is lehet programozni

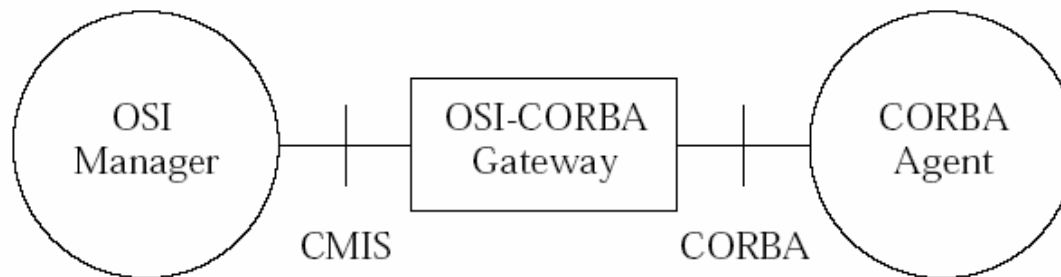
Felhasználási példák: CORBA menedzser, OSI ágens

- Menedzsment objektumok GDMO-ban definiálva
- Ezt CORBA IDL reprezentációvá fordítják, hogy a menedzser megértse az ágens objektumait
- A CORBA-alapú menedzser a fordított IDL alapján CORBA objektumokat ér el és azokon keresztül menedzseli az ágenszt (a menedzselt rendszert)
- A CORBA objektumok mögött levő gateway a CORBA kéréseket OSI kérésekké (CMIP protokollüzenetekké) fordítja
- Az ágens csak CMIP forgalmat lát



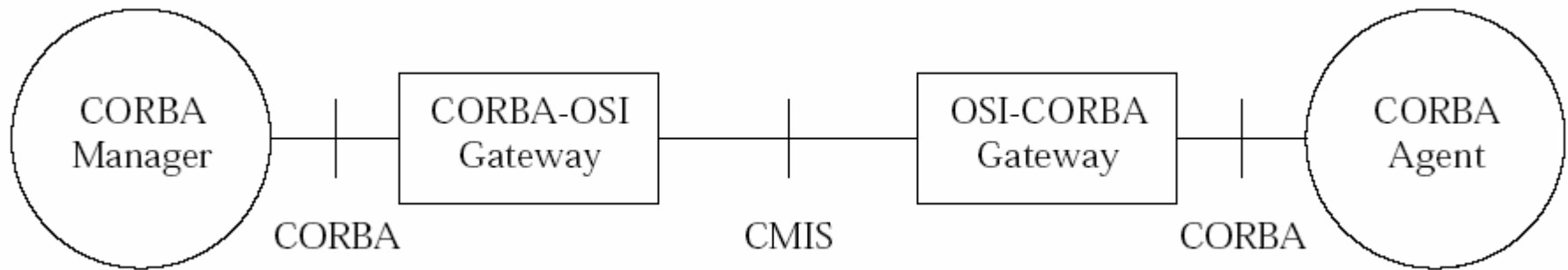
Felhasználási példák: OSI menedzser, CORBA ágens

- Menedzsment objektumok GDMO-ban definiálva
- Ebből IDL leírást fordítanak az ágens részére
- Az ágens menedzsmentfunkcionalitását az IDL alapján készített CORBA objektumok segítségével implementálják
- A menedzser az ágenst a GDMO leírás alapján menedzseli
- A menedzser CMIP kéréseit a gateway CORBA hívásokká fordítja, így jutnak el az ágens menedzsmentkódjához
- A menedzser és gateway között CMIP, a gateway és az ágens között IIOP forgalom van



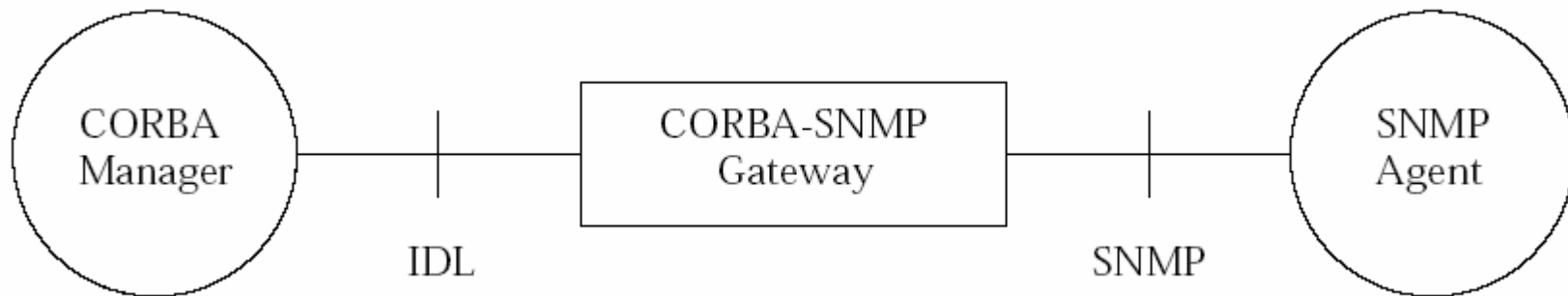
Felhasználási példák: CORBA menedzser, CORBA ágens, CMIP forgalom

- Mind a menedzsert, mind az ágenszt CORBA-ban írják
- A menedzsment objektumokat GDMO-ban írjuk le. Ezt IDL-lé fordítjuk a menedzsernek és ágensnek is
- Mind a menedzser, mint az ágens CORBA modellben írt, a GDMO leírásból generált CORBA objektumokat használják a menedzselésre ill. a menedzsmentműveletek kezelésére
- A menedzser és az ágens CORBA objektumai CMIP segítségével kommunikálnak, a menedzser és az ágens között CMIP forgalom van



Felhasználási példa: CORBA menedzser, SNMP ágens

- Az SNMP MIB-ből IDL-t fordítanak
- Ennek az IDL-nek objektumait használja a menedzser, hogy CORBA hívásokkal menedzselje az ágenst
- Az generált IDL mögött levő CORBA objektumok SNMP üzeneteket generálnak és fogadnak
- A gateway és az ágens között SNMP forgalom van

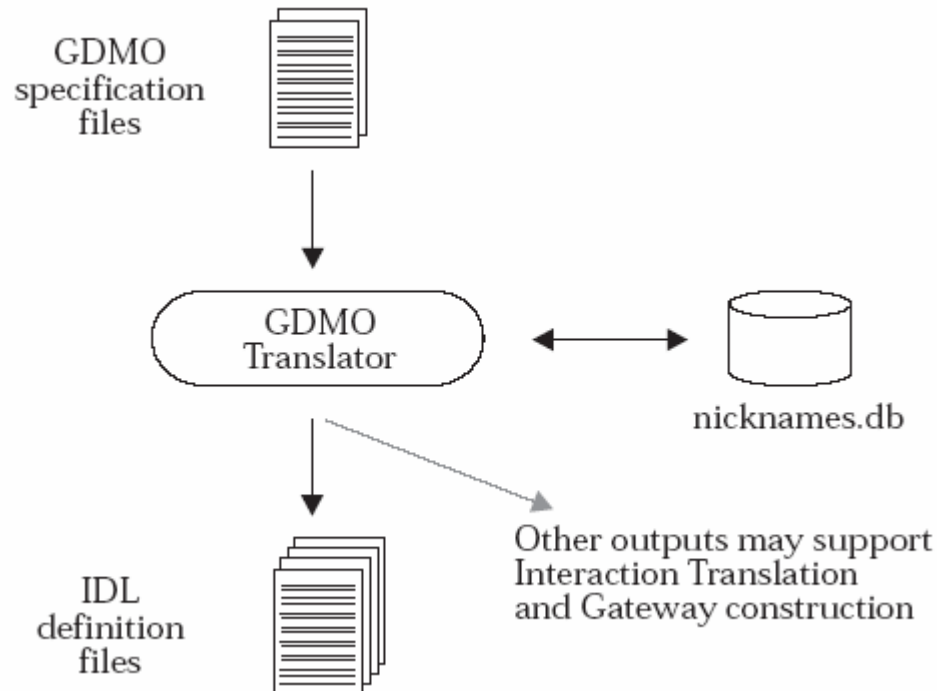


JIDM specifikációk

- Specification Translation – Azzal foglalkozik, hogyan lehet menedzsment objektum leírások között fordítani (pl. GDMO->IDL)
- Interaction Translation – Azzal foglalkozik, hogyan működik a protokollfordító gateway. Függ a Specification Translation-

GDMO->IDL fordítás

- A GDMO gazdagabb, mint az IDL, a fordítás során információ vesz el
- A GDMO fájlt több IDL fájlra fordítják
- Egy GDMO menedzsment objektum osztályt egy készletnyi fájlra fordít. A nevekkel probléma van, ezért egy "nickname" adatbázist használ a GDMO objektumnév és a megfelelő készletnyi IDL fájl között. Pl. az X721 Attribútum modult az X721Att névvel kezdődő IDL fájlkészletre fordítja



GDMO->IDL fordítás

- Minden GDMO MANAGED OBJECT CLASS templétet egy IDL interfészre fordít, ami tartalmazza a menedzsmet objektum metódusait és a ManagedObject interfész leszarmazottja
- Ezen felül az IDL interfész leszarmazottja az összes generált IDL interfésznek, ami a GDMO templétben a DERIVED FROM kulcsszóval van megjelölve (a GDMO templétből IDL interfész keletkezik és az IDL interfészek között ugyanolyan leszarmazási reláció van, mint a GDMO objektumok között)
- Feltételesen jelen levő csomagokat mindig befordítja – az implementációnak kell hibát jeleznie akkor, ha a csomagnak nem szabad jelen lennie

GDMO attribútumok fordítása

- Minthogy a GDMO attribútumok nagyon bonyolultak lehetnek (halmaz, default értékek), ezért minden attribútumhoz egy saját IDL interfészt generál
- `<type> <attribute-label>Get() raises (ATTRIBUTE_ERRORS);`
`void <attribute-label>Set(in <type> value) raises (ATTRIBUTE_ERRORS);`
`void <attribute-label>Add(in <type> value) raises (ATTRIBUTE_ERRORS);`
`void <attribute-label>Remove(in <type> value) raises (ATTRIBUTE_ERRORS);`
`<type> <attribute-label>SetDefault() raises (ATTRIBUTE_ERRORS);`
- `<type>` az ASN.1 típus IDL megfelelője (a JIDM specifikációnak van egy ASN.1->IDL szekciója is)

GDMO attribútumok fordítása

CMIS Művelet	IDL művelet
GET	<attribute-label>Get
REPLACE	<attribute-label>Set
GET-REPLACE	<attribute-label>Get és <attribute-label>Set
ADD	<attribute-label>Add
REMOVE	<attribute-label>Remove
ADD-REMOVE	<attribute-label>Add és <attribute-label>Remove
REPLACE-WITH-DEFAULT	<attribute-label>SetDefault

Akciók

- Egy vagy több IDL metódushívássá fordítódnak (egy a konfirmált, egy a nem konfirmált hívásnak)
- Példa:
suspend ACTION MODE CONFIRMED
PARAMETERS not-supported, not-running;
WITH INFORMATION SYNTAX ActionModule.Suspend;
WITH REPLY SYNTAX ActionModule.ServerState;
REGISTERED AS { 1 2 3 4 9 };
megfelel
ActionModule::ServerStateType suspend(in ActionModule::SuspendType
actionInfo) raises (ACTION_ERRORS, UsingMR);
// PARAMETERS not-supported, not-running;
- UsingMR kivétel: ha több válasz van, így jelez a metódus, hogy további válaszüzenetek jönnek

Akciók (2)

- suspend ACTION PARAMETERS not-supported, not-running, time-delay;
WITH INFORMATION SYNTAX ActionModule.Suspend;
REGISTERED AS { 1 2 3 4 9 };

megfelel

```
void suspend (in ActionModule::SuspendType actionInfo) raises  
(ACTION_ERRORS);
```

```
// PARAMETERS not-supported, not-running, time-delay;
```

```
oneway void suspendUnconfirmed (in ActionModule::SuspendType actionInfo);
```

- Minthogy ezt a suspend-et meg lehet hívni konfirmálva és nem konfirmálva, ezért két metódus van a kétféle híváshoz (oneway – CORBA kulcsszó, hogy a hívásnak csak a kérés részét kell küldeni, nincs válasz)

Események

- Push vagy pull formátum
- CORBA eseménynek nincs visszatérési értéke (a GDMO-ban van, de ritkán használt) – nem teljes a kompatibilitás push esetén
- Pull esetén viszont a menedzsernek meg kell hívnia az eseményforrást
- objectCreation NOTIFICATION BEHAVIOUR objectCreationBehaviour;
WITH INFORMATION SYNTAX Notification-ASN1Module.ObjectInfo AND
ATTRIBUTE IDS
sourceIndicator SourceIndicator,
attributeList AttributeList,
notificationIdentifier NotificationIdentifier,
correlatedNotifications CorrelatedNotifications,
additionalText AdditionalText,
additionalInformation AdditionalInformation;
REGISTERED AS { joint-iso-ccitt ms(9) smi(3) part2(2) notification(10) 6 }

Események (2)

- // in X721_N module (push)
oneway void objectCreationUnconfirmed (
 in ASN1_ObjectIdentifier sourceObjectClass,
 in X711CMI::ObjectInstanceType sourceObjectInstance,
 in X711CMI::ASN1_GeneralizedTimeOpt eventTime,
 in X711Not::ObjectInfoType notifInfo);
 - // in X721_NP module (pull)
void pull_objectCreation (
 out ASN1_ObjectIdentifier sourceObjectClass,
 out X711CMI::ObjectInstanceType sourceObjectInstance,
 out X711CMI::ASN1_GeneralizedTimeOpt eventTime,
 out X711Not::ObjectInfoType notifInfo);
-
- Aki az üzenetet küldte
- Az üzenet maga
- ASN.1 struktúrából fordított objektum (WITH INFORMATION SYNTAX)

SNMP leképezés átugorva ...

Interaction Translation

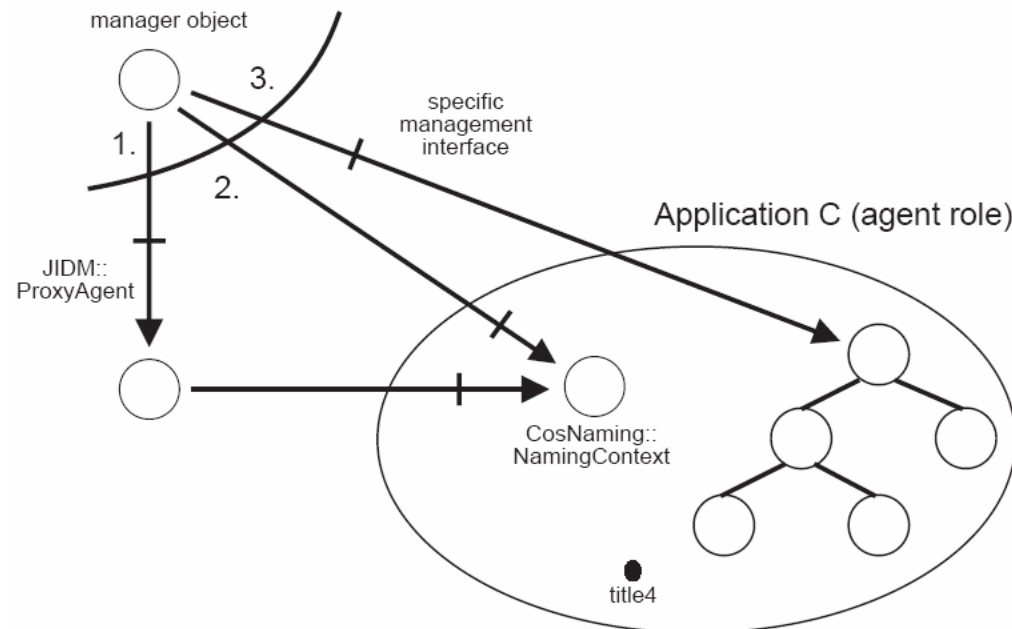
- Pontos definiálja, hogyan változnak CORBA metódushívások CMIP/SNMP protokollüzenetké (ill. fordítva: hogyan lesznek CORBA metódushívások CMIP/SNMP üzenetkből)
- Modell:
 - Menedzsmet objektumok összetartozó együttese a *domain*
 - Minden domain-nek címkéje van (legalább egy, vagy több)
 - Egy címke csak egy domain-hez tartozhat
 - A menedzser és az ágens között egy kapcsolat neve *port*. Egyszerre több port is lehet nyitva egy menedzser és egy ágens között
 - Domain port a neve a kapcsolatnak, ha a menedzser metódusok meghívására használja
 - Event portnak hívják, ha az ágens üzeneteket küld a menedzsernek rajta keresztül

Interaction Translation (2)

- A következő interfészeket használhatják a JIDM menedzserek és ágensek
- Menedzserek:
 - ProxyAgent interface: létrejön, amikor egy menedzser kapcsolatot létesít egy domain-nel
 - ProxyAgentController interface: vezérli a ProxyAgent-ek elpusztítását
 - ProxyAgentFinder interface: egy kulcsnak és kritériumnak megfelelő ProxyAgent-et keres meg vagy hoz létre az adott domain-en
 - EventPort interface: az ágens használja, ha eseményt akar küldeni a menedzsernek
 - EventPortFactory interface: EventPort-okat gyárt
- Egy ágens ezen felül a következő interfészeket használja
 - DomainPort interface: az ágensben jön létre a ProxyAgent objektum létrehozásának hatására
 - DomainPortFactory interface: DomainPort-okat gyárt

Menedzsment objektumok elérése CORBA hívásokkal

- ProxyAgent megszerzése az adott domain-hoz
- A ProxyAgent-ből a domain CosNaming elnevezési kontextusának megszerzése (get_domain_naming_context() a ProxyAgent interfészen)
- A menedzsment objektum referenciájának megszerzése a menedzsment objektum neve alapján (resolve() az elnevezési kontextuson)
- A menedzsment objektum elérése



Menedzsment objektumok elérése nem-CORBA menedzsment protokollon keresztül

- Az előbb említett objektumok a gateway-ben léteznek

