Extending CORBA For Multi-Domain Management

Bela Ban
IBM Research Division
Zurich Research Laboratory
email: bba@zurich.ibm.com
www: http://www.zurich.ibm.com/~bba/
Postscript: http://www.zurich.ibm.com/~bba/doct96.ps

October 2, 1996

Outline

- Motivation
- Related Work
- Proposal for a dynamic, CORBA-based (management) API
  - Object Model
  - Extending CORBA's Interface Repository
  - Adapters
- Conclusion

Motivation

- CORBA is becoming more important in NM.
- Other objects model prevalent in NM: SNMP, CMIP
- Goal: Use of CORBA for NM
  - Integration of existing models using a dynamic approach
- Benefits:
  - Transparent management of all models using only CORBA
  - Use of a large existing base of management functionality
  - Protection of existing investment (OSI / SNMP)

Related Work: XoJIDM

- Objectives:

  1. *Manage OSI agents using CORBA clients*
  2. Manage SNMP agents using CORBA clients
  3. Manage CORBA agents using OSI managers
  4. Manage CORBA agents using SNMP clients

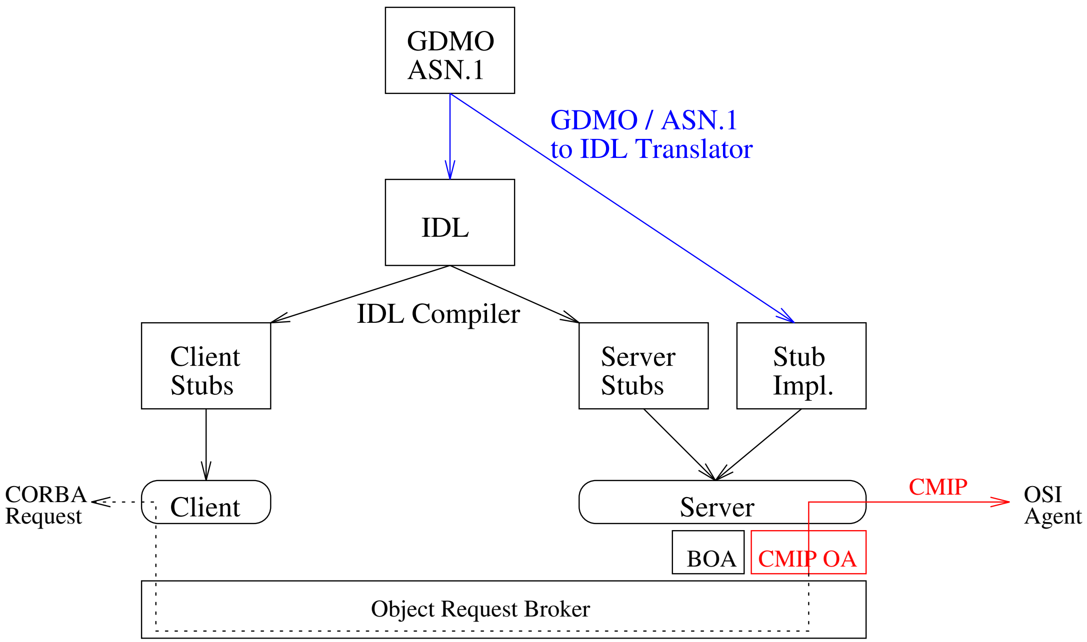
- Approach to 1:

  – Translate GDMO / ASN.1 specification to IDL (*Specification Translation*)
  – Translate CORBA operations to CMIP requests and reverse (*Interaction Translation*)

XoJIDM: Specification Translation Mapping

GDMO Templates	IDL Interfaces (up to 3: class, notifications and multiple replies)
GDMO Packages	IDL Interfaces
GDMO Attributes	IDL Types / Interfaces
GDMO Attribute Access	IDL Operations (get_X(), set_X())
GDMO Actions	IDL Operations
GDMO Notifications	IDL interfaces
GDMO Parameters	IDL Interfaces
ASN.1 Types	IDL Types

XoJIDM: Specification and Interaction Translation



XoJIDM: Features of the Static Approach

- Static code generation: Modification of specification requires recompilation
- Tight binding between clients and servers: inclusion of generated language bindings at client compile time
- Mass of generated code (through transitive closure over include files): A lot of unneeded information is included in clients
- Problems of mapping certain GDMO / ASN.1 idiosyncracies statically:
  - Conditional packages
  - `ANY DEFINED BY` ASN.1 type
  - Attribute groups

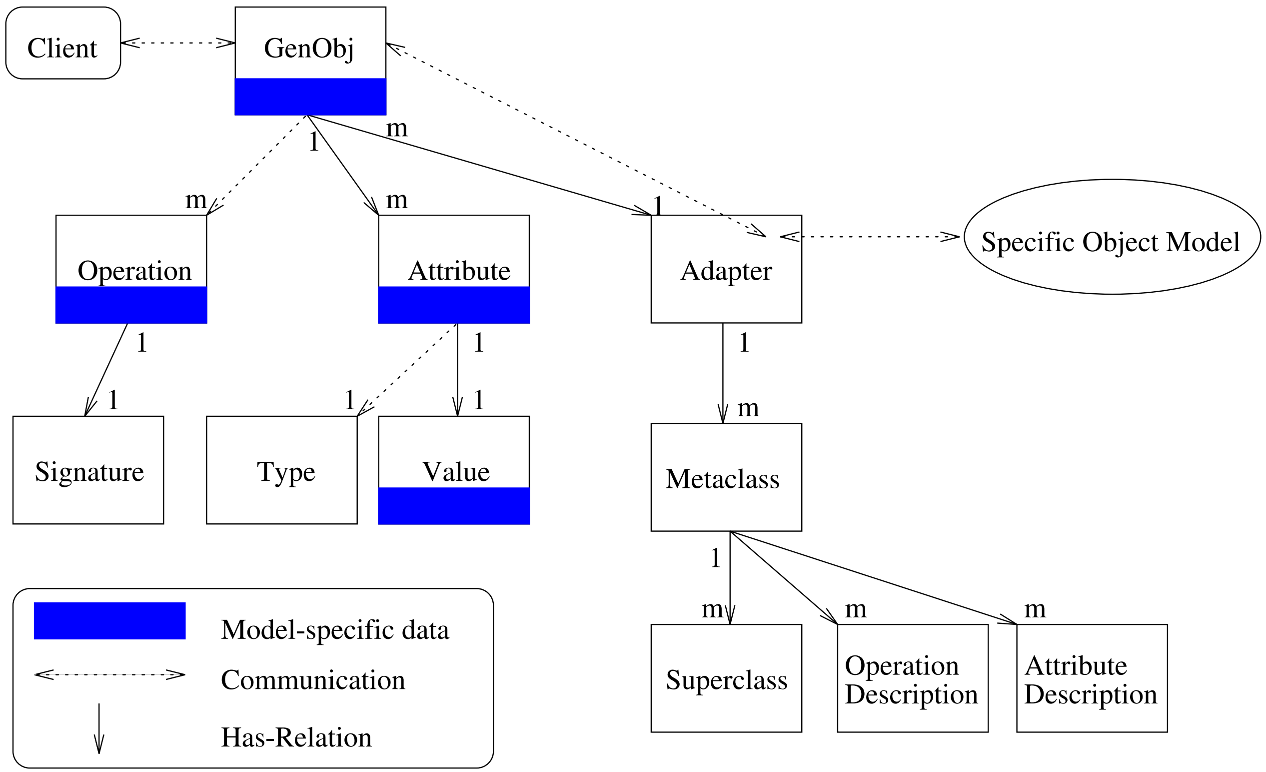
Requirements for a Dynamic Approach

- Creation of instances of classes that were not known *at compile time*: breaks client dependency on generated IDL interfaces

- Dynamic addition of attributes, operations *at runtime*: support for GDMO conditional packages, attribute groups etc.

- Approach:
  - Modelling of classes, attributes, operations, values etc. as *CORBA interfaces* (reification). Generic set of value interfaces: `Sequence`, `Integer`, `Struct`, `String`, `Boolean` etc. (14)
  - This allows to represent instances of all classes in the system through instances of a single interface (`GenObj`)

Object Model

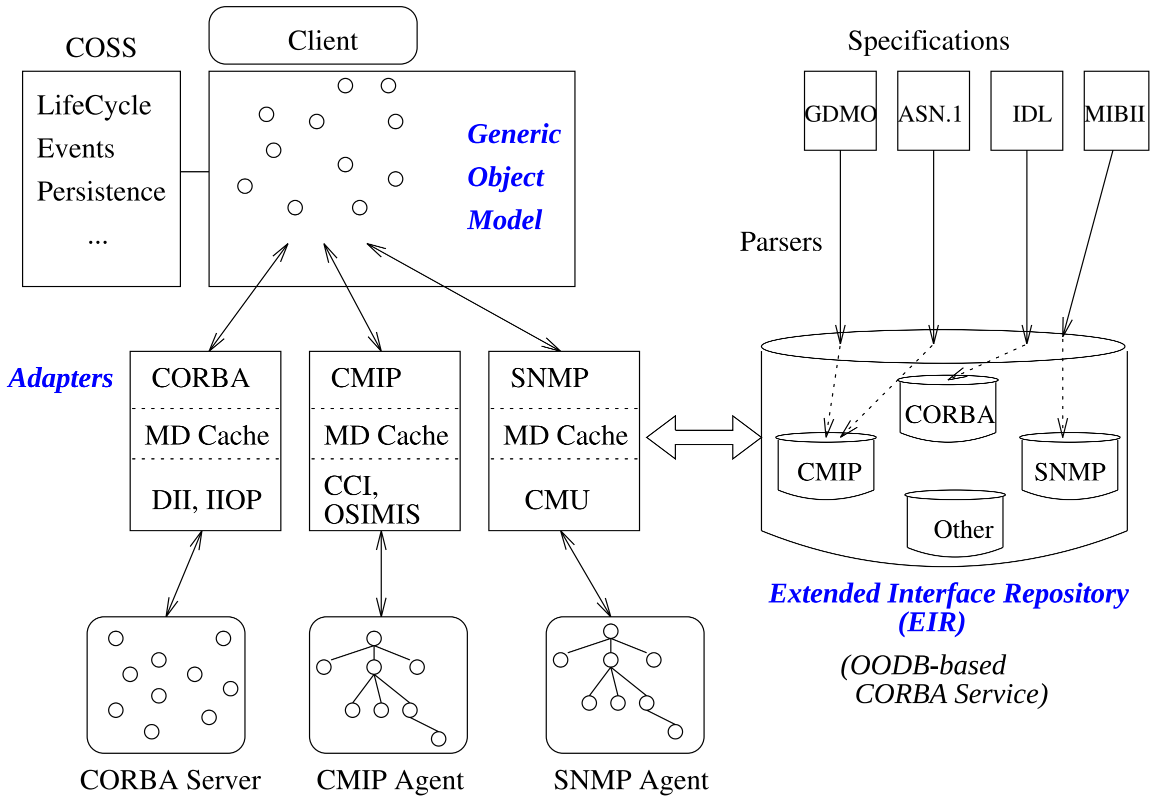


GenObj IDL Interface

```
interface Adapter; // subclass of GenObj
interface ArgList; interface Attribute; interface Val;
interface ExecMode;  // synchronous, asynchronous and sync mechanisms
typedef sequence<string>    StrList;
typedef sequence<Attribute> AttrList;

interface GenObj {
  readonly attribute Adapter   ad;
  readonly attribute string    class_name;
  readonly attribute AttrList  attributes;
           attribute any       specific_data;
  Val      Get(in string attrname);
  boolean  Set(in string attrname, in Val new_val);
  Val      Exec(in string opname, in ArgList args, in ExecMode mode);
};
```

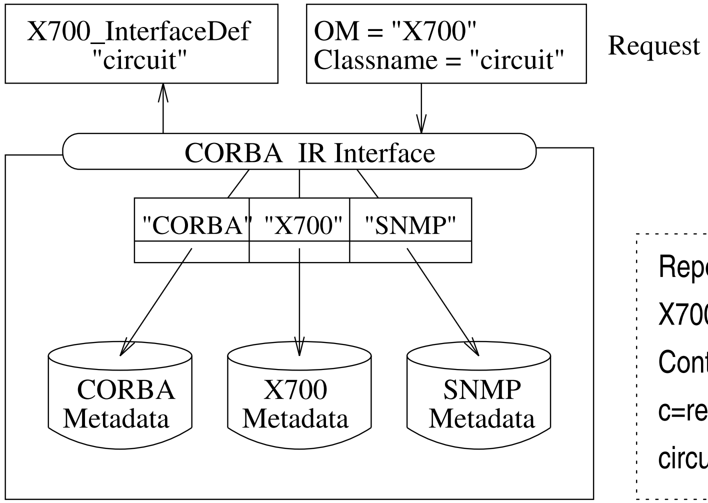
Architecture



Adapters

- All of the conversion / adaptation work is done here
- Translation of requests between the dynamic and (exactly one) specific object model
- Association between class that `GenObj` represents and adapter to use
  - is in EIR (determined by administrator when parsing specifications using a special compiler flag)
  - is determined by user when creating an instance: `Create("X700", "system", "netId=TelcoNet;systemID=(name IBM)")`

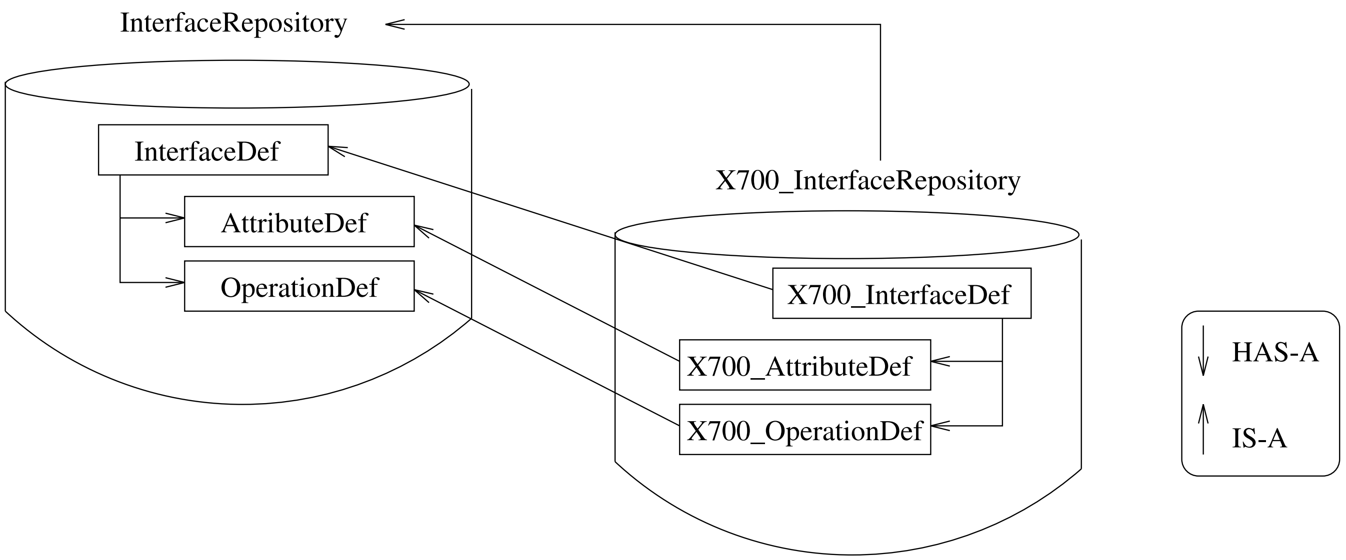
Extending CORBA's IR




```
Repository* rep;
X700_InterfaceDef* circuit_def;
Contained* c;
c=rep->lookup_id("X700:/op1vol4/circuit:1.1");
circuit_def= ( /* Castdown */ )c;
```

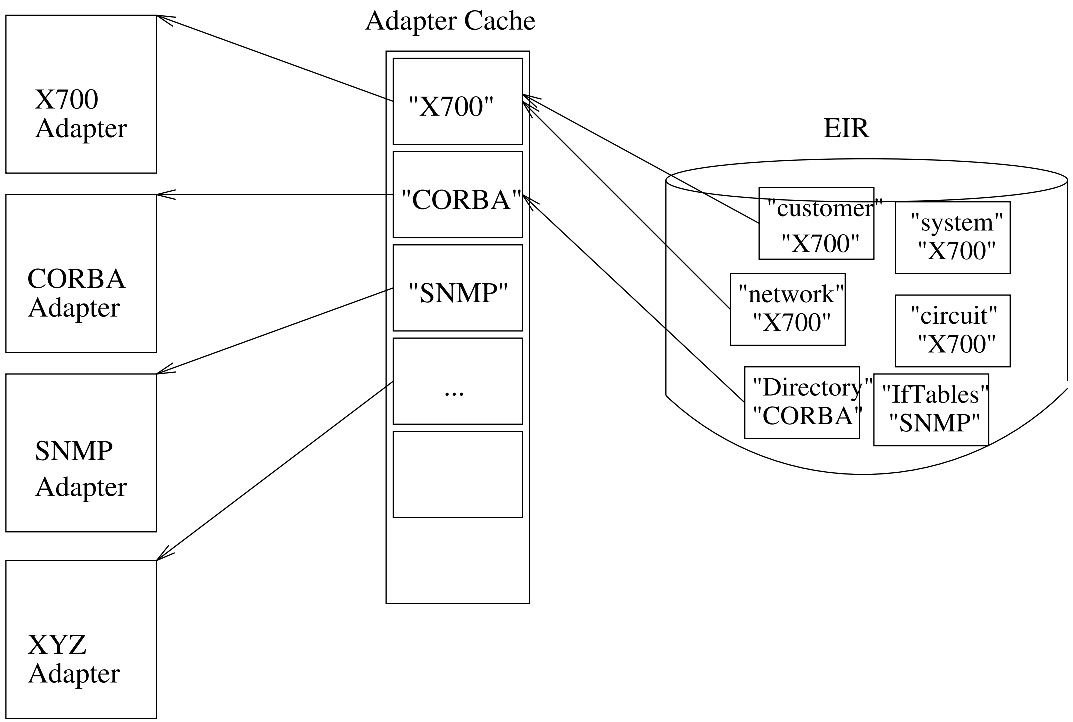
- IR is extended to allow clients to use it for metadata of other object models; *interface of IR remains the same*, implementation needs modification

Maintaining Specific Metadata



- Object model specific metadata is derived from normal IR interfaces
- New object model specific metadata interfaces are only introduced if specific object model element (e.g an attribute) cannot be described using interfaces available in IR

Adapters and the EIR



Advantages

- Uniform CORBA-based client (management) API to other object models: small learning curve
- Flexible binding between clients and servers based on metadata: no recompilation of clients when specification is modified
- Small memory footprint: only classes that are really needed are (dynamically) included

Conclusion: Where can this model be used ? (1)

- CORBA management applications: instances of several (object) models can be managed transparently
- Applications where the classes used are not fixed / known at compile time: topology browsers, interpreters
- Applications where memory conservation is important

Conclusion: Where can this model be used ? (2)

- CORBA-centric: companies that already use CORBA and need to access other ('legacy') object models
- Roaming agents:
  - Roaming agents encounter objects whose services they want to use, but don't have type information about them
  - It is virtually impossible for roaming agents to know the types of all instances they will ever encounter on their trips
  - Implementation of interpreter on top of the dynamic object model (GOMscript): `http://www.zurich.ibm.com/~bba/gomscript.html`
  - Simple roaming agent execution platform using GOMscript: `http://www.zurich.ibm.com/~bba/agent.html`

References

[1] Postscript version of presentation:
http://www.zurich.ibm.com/~bba/doct96.ps

[2] Bela Ban: Extending CORBA For Multi-Domain Management.
http://www.zurich.ibm.com/~bba/corbagom.ps

[3] Bela Ban: GOMscript: An Interpreter For GOM.
Versions for AIX 3.2.5, 4.1.X, Linux 2.0.0
http://www.zurich.ibm.com/~bba/gomscript.html

[4] Bela Ban: Using GOMscript As An Agent Execution Platform.
http://www.zurich.ibm.com/~bba/agent.html

[5] Bela Ban: The Generic Object Model: Towards An Object-Oriented Framework For Multi-Domain Management. IBM Research Report RZ 2789. http://www.zurich.ibm.com/~bba/GOM.ps

[6] Papers, Presentations:
http://www.zurich.ibm.com/~bba/papers_and_presentations.html