**CCM: The CORBA Component Model**

- Portable Object Adapter (POA) revisited
- CORBA Objects
- CORBA Component Model

**Configuring a POA**

<table>
<thead>
<tr>
<th>POA policy type</th>
<th>Allowed values</th>
<th>Root POA policy</th>
<th>Child POA default policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Policy</td>
<td>ORB_CTRL_MODEL (multi-thr.), SINGLE_THREAD_MODEL</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Life Span Policy</td>
<td>TRANSIENT, PERSISTENT</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>ID Assignment Policy</td>
<td>SYSTEM_ID, USER_ID</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>ID Uniqueness Policy</td>
<td>UNIQUE_ID, MULTIPLE_ID</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Request Processing Policy</td>
<td>USE_ACTIVE_OBJECT_MAP, ONLY USE_DEFAULT_SERVANT, USE_SERVANT_MANAGER</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Servant Retention Policy</td>
<td>RETAIN, NON_RETAIN</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Implicit Activation Policy</td>
<td>NO_IMPLICIT_ACTIVATION, IMPLICIT_ACTIVATION</td>
<td>✅</td>
<td>✅</td>
</tr>
</tbody>
</table>

**Configuring a POA, continued**

- **Remark:** Not all theoretically possible combinations make sense / are permitted.
- **RootPOA:** The default POA when the server comes up
- **Customized child POA’s can be created with different policies**

**Life span of CORBA Objects**

- **Transient objects**
  - Lifetime bounded by
    - lifetime of server process that created it
    - and its POA
  - Usually, short-lived
- **Persistent objects**
  - Lifetime unbounded
  - Its creating server can be stopped and restarted multiple times
  - Dormant when its POA is destroyed;
    reactivated when its server is restarted and POA is reinstantiated
  - State stored in a database

**Types of CORBA Objects (1)**

- **Session objects**
  - usually short-lived, non-persistent
Types of CORBA Objects (2)

- **Entity objects**
  - Usually long-lived, persistent

Client

Server

POA - "EntityPOA"

`Fred`

Servant

IO

Create servant

Create DB record

Activate servant

Create IOR

Types of CORBA Objects (3)

- **Factory Object**
  - Create, find and destroy other objects
  - Perform steps 1, 2, 3, (4) as above for session / entity objects
  - Can itself be a session or entity object

CORBA Component Model (CCM)

- Abstraction of CORBA object with its environment
- **Container**
  - Abstraction of POA functionality
  - Life cycle management
  - Provision of CORBA Services for enterprise applications
    - Security, transaction, event service
    - Simplified interfaces covering most commonly used features
  - Support for ready-made components
  - Configuration and assembly
  - Separate component logic and CORBA functionality
    - E.g., hide house-keeping code (POA object creation, life cycle management for CORBA objects etc.)
- **EJB look-and-feel**
- **Basic components**: Fully EJB compatible

CORBA Component Model Architecture

- **Home**: Factory object to create, find, remove component instances
- **Component**: Basic building block for server application
- **Container**: Environment that supports components
- **Callback interfaces for container to notify component about events etc.**
- **Receptacle**: Registry for component references - "Uses" relation made explicit

CORBA Component Categories

- **Session components** (cf. EJB stateful session beans)
  - Transient state, transient identity
- **Service components** (cf. EJB stateless session beans)
  - No state, no identity
- **Entity components** (cf. EJB entity beans)
  - Persistent state
  - Persistent identity, automatically visible to clients
- **Process components**
  - Persistent state
  - Persistent identity, not automatically visible to clients

CORBA Component Model

- **Basic components**: Fully EJB compatible and interoperable
- **Extended components**: Beyond EJB, e.g. ports
  - Ports: Special sorts of interfaces, grouped together
    - Facets
    - Receptacles
    - Event sources
    - Event sinks
  - Configuration interfaces with setters and getters
- **Extended IDL** (Component IDL, CIDL)
  - New elements: component, interface, home, finder, factory