Fine Grained CORBA Services to Build Scalable DRT&E Architectures

Victor Giddings
Objective Interface Systems, Inc.
victor.giddings@ois.com
Object Model Architecture

- Application Objects
- Vertical CORBA Facilities
- Horizontal CORBA Facilities

Common Object Request Broker

CORBA services
“The CORBA Services component standardizes the life cycle management of objects.”

Defined in a set of specifications:

- Additional Structuring Mechanisms for the OTS
- Collection Service
- Concurrency Service
- Enhanced View of Time
- Event Service
- Externalization Service
- Licensing Service
- Life Cycle Service
- Lightweight Log Service
- Management of Event Domains
- Naming Service
- Notification Service
- Persistent State Service
- Property Service
- Query Service
- Relationship Service
- Security Service
- Telecoms Log Service
- Time Service
- Trading Object Service
- Transaction Service
Most CORBA services implementations have been monolithic servers

- Single standalone process or program
- On first examination, makes some sense
  - Interface, implementation, and execution environment delivered in one package
  - Persistence, recovery, etc. features bundled into server
- Reinforced by resolve_initial_references mechanism

```c
The_Event_Channel = CosEventChannelAdmin::EventChannel::_narrow(
  orb->resolve_initial_references("EventService"));
```

- Only one
- Statically configured by Object URL or proprietary means
But, services are specified in “fine-grained” manner

- Service interfaces are (fine-grained) CORBA objects
  - Location-transparent
  - Could be in same process as client of service
- Services include features for collaboration or composition
  - Name Service – federation
  - Event/Notification
    - Separation of administrative interfaces from producer and consumer interfaces
    - Setup deliberately more complicated than needed – in order to support “joining” of event channels
Embedded systems may not have
- Resources for separate processes
- Processes (single address space O/Ses)
- Disks for persistent storage

Fine-Grained services can be delivered as library-based implementations
- Can be integrated with ORB-provided and user-selected
  - Alternate transports
  - Real-time CORBA feature usage
    - Banded connections
    - Priority propagation
  - Security
- “Collocation” optimizations
Fine-Grained CORBA Services Example Implementation

- ORBexpress Names: Embedded Objects & ORBexpress Events: Embedded Objects
  - Library-based implementations
  - Relatively small footprint

```
<table>
<thead>
<tr>
<th>ID</th>
<th>NAME</th>
<th>.text</th>
<th>.data</th>
<th>.bss</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>lib0</td>
<td>291088</td>
<td>3655</td>
<td>120</td>
</tr>
<tr>
<td>0</td>
<td>lib0</td>
<td>10568</td>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>lib0</td>
<td>79120</td>
<td>156</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>lib0</td>
<td>114400</td>
<td>200</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>lib0</td>
<td>22592</td>
<td>76</td>
<td>1</td>
</tr>
<tr>
<td>Total:</td>
<td>1413028</td>
<td>114352</td>
<td>36698</td>
<td></td>
</tr>
</tbody>
</table>
```

Names Library: 80K
Events Library: 115K
Prototype Application – Two Physical Configurations

Board 1

- CSPI node 1
  - Part 1
- CSPI node 2
  - Part 2

Health and Status Monitor

- CSPI node 2
  - Part 0

- CSPI node 3
  - Part 1
- CSPI node 4
  - Part 2

- CSPI node 5
  - Part 1
- CSPI node 6
  - Part 2

Health and Status Monitor

- CSPI node 4
  - Part 0
- CSPI node 6
  - Part 0
Prototype Application
Using “Chained” Event Channels

Part 1
Part 2

Board Channel

Part 1
Part 2

Board Channel

Part 1
Part 2

Board Channel

Health and Status Monitor

System Channel
Using Fine-Grained Name Service to Reflect Components

Stage 1

Stage 2

Stage 3

“Stage 1”

“Stage 2”

“Stage 3”

“H&S”

Health and Status Monitor
Using Fine-Grained Name Service for Sub-Components

Stage 1

Request Distribution Channel

Name Binding

Part 1

Part 2

Board Channel
“Architectural Glue” in Prototype Application

- **Fine-grained event service**
  - “Chained” event channels – on-board channel feeds system channel
  - “Local” event channels – single-typed channel provides data transfer for collective invocation

- **Fine-grained name service**
  - “Federated” fine-grained name services reflect
    - Application structure
    - Application deployment
    - Sub-component structure and implementation
  - Simple naming convention allows reflection and dynamic query
Fine-Grained CORBA Services

What is Missing?

- Library-based Implementations
- Factory interfaces
  - Name Service - can’t create a NamingContext without a reference to a NamingContext
  - Event Service – no factory for EventChannel (addressed in Notification Service)
- Location controls
  - new_context creates NamingContext at same location as “parent” NamingContext
  - ConsumerAdmin, SupplierAdmin and Proxies will be created collocated with EventChannel
  - Deployment and Configuration Specification??