Local Components and Reuse of Legacy Code in the CORBA Component Model

Egon Teiniker, Stefan Mitterdorfer, Christian Kreiner, Zsolt Kovacs, Reinhold Weiss

Institute for Technical Informatics
Graz University of Technology
AUSTRIA
Overview

• CORBA Component Model
• Development Requirements
• Related Work
• Local Component Adapter Concept
• Remote vs. Local Components
• Reuse of Legacy Code
• Component Development Process
• Experimental Results
• Conclusion and Future Work
CORBA Component Model

**CORBA Component**

- Home Interface
- Equivalent Interface
- Facets
- Receptacles
- Attributes
- Event Sinks
- Event Sources

**Component Container**

- Home
- Component
- Callback Interface
- Internal Interface
- POA
- CORBA Interface
- External Interface

---

Institute for Technical Informatics
Graz University of Technology

http://www.iti.tu-graz.ac.at
CORBA Component Model

CORBA Component Assembly
Development Requirements

Problems with CORBA Component Model:

• CORBA is complicated
  – Business logic
  – Legacy code

• Components in the same address space
  – Performance
Related Work

• Local Enterprise JavaBeans
  – since EJB 2.0

• CORBA Collocation
  – Standard Collocation
  – Direct Collocation

• CCM Language Mapping
  – Local CORBA interfaces
Local Component Adapter Concept

Diagram showing the concept of a Local Component Adapter Concept with Home Interface, Equivalent Interface, Facets, Attributes, and Home Component as key components.
Local Component Adapter Concept

Component 1

C++ Business Logic

A_R

Receptacle Adapter

C++

IDL3

A_F

Facet Adapter

C++

Component 2

C++ Business Logic

A_L

local path

remote path

I_1

I_3

I_2
Remote vs. Local Components

IDL3 / C++

local CCM

remote CCM

CORBA

local CCM

remote CCM
Remote vs. Local Components

Process A
- Home
- Component
  - local

Process B
- Home
- Component
  - remote
Reuse of Legacy Code

CCM Component Wrapper

Legacy Code
Component Development Process

Institute for Technical Informatics
Graz University of Technology
http://www.iti.tu-graz.ac.at
Experimental Results

Example Assembly:
Experimental Results

typedef sequence<long> LongSeq;

interface I2 {
    long f1(in long a);
    long f2(in string a);
    long f3(in LongSeq s);
}

<table>
<thead>
<tr>
<th>Function</th>
<th>Remote [ms]</th>
<th>Local [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1()</td>
<td>2824</td>
<td>36</td>
</tr>
<tr>
<td>f2()</td>
<td>4478</td>
<td>32</td>
</tr>
<tr>
<td>f3()</td>
<td>11758</td>
<td>32</td>
</tr>
</tbody>
</table>

1,000,000 Requests on a Pentium III 700MHz
Conclusion and Future Work

• **Current status:**
  - Local component adapter concept
    - no CORBA in business code
    - easy reuse of legacy code
  - Assemblies of local and remote components
    - local components

• **Next steps:**
  - Embedding scripting languages in CCM
  - Integration of entity concept
  - Integration of event handling