Architecture of the CORBA Component Model

Course Overview

Course Topics

- CORBA (Common Object Request Broker Architecture)
- CCM (CORBA Component Model)
- IDL (Interface Definition Language)
- IDL Language Mappings: Java, C++, (Python)
- .NET SSCLI (Shared Source Common Language Runtime)

Course Topics (2)

- CCM Architecture:
  - IDLv3: Component Definitions
  - CIDL (Component Implementation Definition Language)
  - Container Architecture
  - (Supporting Services: Security, Transactions)

Course Topics (3)

- Standardization and Standards Conformance:
  - OMG (Object Management Group)
  - ECMA (European Computer Manufacturers Association)
  - Role of Conformance, Portability, Interoperability
  - Comparing Specification and Implementation

Course Topics (4)

- .NET SSCLI:
  - Architecture of .NET
  - Architecture of the implementation
  - Conformance to specification

Practical Work

- Experiments with CORBA and CCM implementations
- Focus on C++ as an implementation language, Java and others as an option
- CORBA testbed (C-E.4)
- Studying specifications

What Are Components?

- Szyperski (in „Component Software“):
  - A component is a unit of independent deployment
  - A component is a unit of third-party composition
  - (A component has no persistent state)
- Terminology: Objects, Components, Instances
  - Component Instances?

CORBA Components

- CORBA v2.x:
  - **Objects** have **interfaces** which are defined in **IDL**
  - Objects are implemented through **programming language mappings**
  - Objects access occurs through **object references**, every access is a procedure/method call
  - Interoperability
  - Object Services
  - Programming Paradigms, Design Patterns

CORBA replaces ad-hoc special protocols with a open, standardized, and portable platform

CORBA Components (2)

- Transparencies of CORBA:
  - Language transparency
  - Location transparency
  - Service transparency
  - Implementation transparency
  - Architecture transparency
  - Operating system transparency
  - (Protocol transparency)
  - (Transport transparency)

CORBA Components (3)

- Issues with CORBA v2:
  - Objects are restricted to a single interface
    - Need for separate service and management interfaces
    - Asynchronous communications is clumsy
  - Dealing with large numbers of objects requires significant book-keeping in application
  - Deploying CORBA applications is difficult
    - Multiple systems
    - Multiple platforms
    - Infrastructure prerequisites

CORBA Components (4)

- CORBA v3.x: Support for Components
- IDL extensions for components:
  - Multiple interfaces (faces)
  - Event-oriented communication
  - Configuration of a component
- Implementation framework and container framework to simplify state management and persistance
- Packaging and Deployment specification

Literatur

- M. Henning and S. Vinoski. Advanced CORBA Programming with C++. Addison-Wesley, 1999.
- OMG. CORBA 3.0.2.
- OMG. CORBA Component Model, v.3.0
- OMG. C++ Language Mapping.
- ECMA. Standard ECMA-335. Common Language Infrastructure.

1. Übungsaufgabe

- Java-Language-Mapping
- http://www.dcl.hpi.uni-potsdam.de/cms/teaching/ccm03/
- Abgabetermin: 21. April