Component Based Software Engineering (3C05/D22)

Unit 12: Component-based SE

Objectives:

- Provide an Overview of Component-Based Development (CBD)
- Review the Changes that have to be done to the development process to adopt CBD
- Provide an overview of infrastructures that enable CBD

What are Components?

- „Components are software units that are context independent, both in the conceptual and the technical domain“ (Ciupke/Schmidt, ECOOP Workshop 96)
- „A component denotes a self-contained entity (black-box) that exports functionality to its environment and may also import functionality from its environment using well-defined and open interfaces. In this context, an interface defines the syntax and semantics of the functionality it comprises. Compnents may support their integration into the surrounding environment by providing mechanics such as introspection or configuration functionality.“ (Stal, Concepts&Tools 19(1998)).

Motivation Components

- Speed of application development
- Reuse beyond lists
- Configuration
- Integration and stepwise migration
- Get application developer closer to domain
- Heterogeneity of platforms
- Separation between interface and implementation

What is CBD?

- Construction and deployment of software systems that have been assembled from components
- Discovery, engineering and purchase of these components
- Re-engineering of legacy software for component assembly

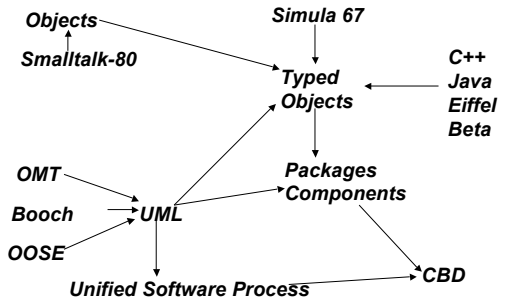
Overview of CBD

- OO Modelling leads to large number of fine-grained classes, objects and relationships
- Difficult to find reuseable parts in these small units
- Idea: Integrate related parts and reuse them together
- These integrated parts are called components

Benefits of CBD

- Reduced Delivery
  - Lookup in Component Catalogues
  - Reuse of pre-fabricated components
- Reduced Costs
  - Buy cheaper than make
- Increased Productivity
  - Developers focus on application development
- Increased Quality
  - Component Producers can allow more time for quality assurance

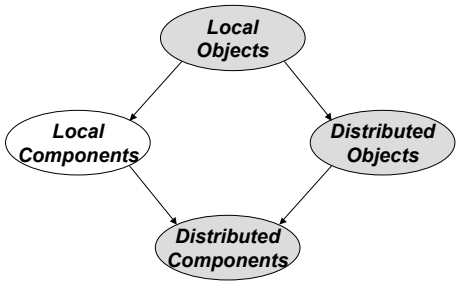
The Route to CBD



Activities that are specific to CBD

- Development of components
  - Using some component infrastructure
- Component publishing
  - (e.g. www.componentsource.com)
- Component search and retrieval
- Component evaluation
- Component assembly
  - Using visual component assembly tools
  - E.g. Java BeanBox

Overview of Component Infrastructures



Local Component Models

- Microsoft Object Linking and Embedding (OLE)
- Javasoft's JavaBeans
- Microsoft's Component Object Model (COM) as of Windows 3.5 (incorporates and replaces OLE)

JavaBeans

- Introduced in Java 1.1
- Supports CBD in Java
- Inspired by Delphi of Inprise (formerly Borland)
- JavaBeans are Java classes that follow certain conventions
- Mostly used for GUI Development

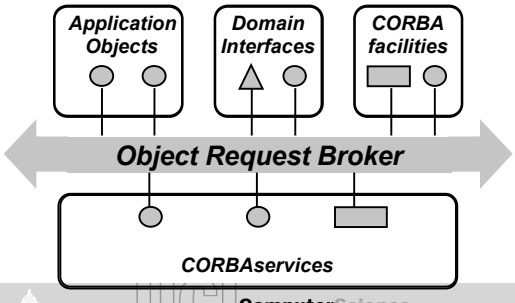
What is a Bean

- „A Java Bean is a reusable Software Component that can be manipulated visually in a builder tool“ (JavaBeans-Whitepaper)
- A Bean is a Java Class, that follows certain conventions:
  - Properties
  - Events
  - Contained in Bean Container
  - Manifest-File, that declares Bean
  - Packaged into jar File with all Java Classes that it uses

Distributed Object Models

- OMG/CORBA
- Microsoft COM
- Java Remote Method Invocation (RMI)

Object Management Architecture / CORBA

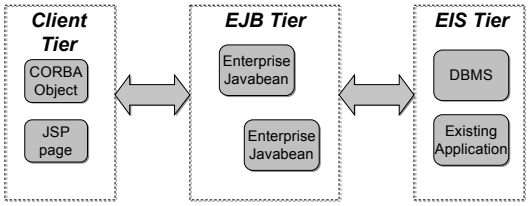


Distributed Component Models

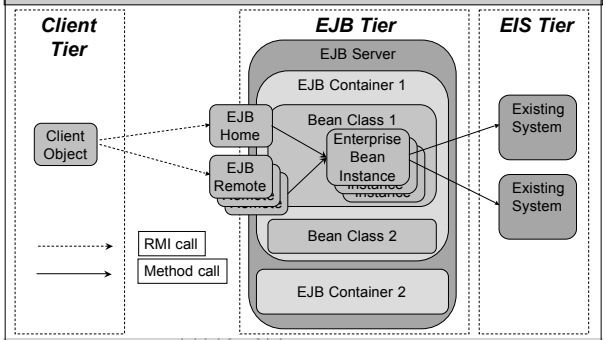
- Microsoft Transaction Server
- .NET
- Enterprise Java Beans
- CORBA Component Model

J2EE Architecture

- Java2 Enterprise Edition
  - Multi-tier development
  - Introduce a middle tier between clients and Enterprise Information Systems (EIS)



EJB Architecture (2)



Key Points

- Component-based Development aims to provide better productivity and quality by systematically re-using coarse grained components
- Component-based Development processes include non-traditional development activities, such as component-evaluation and component retrieval.
- Component-based Development needs to take place within a supporting middleware infrastructure, such as Enterprise Java Beans.