

Component-based Software Engineering Processes

TOOLS'99
Santa Barbara

Moderators

Kingsley C. Nwosu
Lucent Technologies

Robert C. Seacord
Software Engineering Institute
Carnegie Mellon University

Agenda

- Opening remarks - R. Seacord
- Component-based Software Development - R. Seacord
- Mediators for Component Retrieval - R. Braga
- A Generic Process and Terminology for evaluating COTS Software - F. Hansen
- A Component Selection Methodology - I. Kuruganti
- Open Discussion

Workshop Goals

- Identify processes that are *unique* to - Component-based Software Engineering
- Identify *existing* processes that need to be *modified* to accommodate CBSE
- Sit by the pool sipping umbrella drinks

Starting Point....

- What is Component-based software development?

Component-based Software Development

Robert C. Seacord
Software Engineering Institute
Carnegie Mellon University

Etymology

- The whole comprises its parts, and the parts compose the whole. To compose, from the Latin *com* “together” and *pose* “to put.” Obviously, the parts that we compose are, etymologically speaking, components.

A Component Is...

- Binary implementation of functionality
- Contractually specified
- Unit of independent deployment
- Subject to 3rd party composition
- Conformant with a component model

Binary Implementation

- Implementation details of components are hidden from integrators.
 - ensure components are analyzable based on contractually specified properties.
- Components are only modifiable using component-provided mechanisms.
 - ensure no changes to component implementations that would violate interface contracts.

Binary Implementation - 2

- Composition is independent of the programming language or tools used to implement the component.
 - ensure commerce in components does not depend upon universal acceptance of a particular implementation language or programming environment.

What is in a Contract?

- Contracts include essentials for component composition including
 - application programming interface (API)
 - pre- and post-conditions on interfaces
 - specification of quality attribute
 - security
 - performance
 - resource consumption

Benefits of Contractual Specification

- Contracts provide a basis for reasoning about the properties of component assemblies
- Contracts provide a basis for certification of components

Unit of Independent Deployment

- Components are deployed from their development environment to an assembly and runtime environment
- A component is an independent unit of deployment if
 - it is bundled with everything needed for deployment
 - its dependencies on external resources are explicitly documented

Component Composition

- Composition enables communication between components to
 - provide a service
 - request a service

Forms of Composition

- **Component--Component:** composition enabling interaction among components.
- **Component--Framework:** composition enabling interactions between components and a component framework.
- **Framework--Framework:** composition enabling interactions between frameworks, or between components within different frameworks.

Composition Questions

- When two components are composed, is the result necessarily a new component?
- Does a component model need to support runtime composition of components?

Component Model

- Architectural specifications
 - plays the role of an architectural specification for component-based systems
 - component models convey greater specificity
- Define resource management policies
 - load balancing, failover, resource allocation

Component Models & Types

- A component model specifies the types of components in a system and the interactions that can occur among components.
- A component type defines the role a component will play in a system apart from the specific functionality of the component.

Component Frameworks

- Implement the infrastructure of component-based systems.
- A component framework provides
 - runtime services needed to support the component model
 - the run-time environment into which components are deployed and in which components execute and interact with other components

Component-based Software Development Vision

- Component-based Software Development is the assembly of systems from components where
 - the components are commercially available commodities having certified properties
 - the properties of assembled systems can be predicted from the properties of the components

Component Marketplace

- Provides component improvement through
 - competition
 - amortization of development costs
 - built-in upgrade paths

Component Marketplace - 2

- A commodity market in components implies components are commensurable and substitutable, and implies a market in which price and quality become dominant factors in component selection.

Component Certification

- Provides engineers with confidence in component properties that have been certified as having specified values.
 - can be used for engineering analysis in building component assemblies.
- Behavioral properties and quality attributes can both be certified

Component-based System Properties

- System properties can be determined from analysis of component properties
 - specified in contract
 - independently certified