Technical Concepts of Component-Based Software Engineering
2nd Edition
By
Bachmann, Felix; Bass, Len; Buhman, Charles; Comella-Dorda, Santiago; Long, Fred; Robert, John; Seacord, Robert; Wallnau, Kurt
May 2000

Presented by: Manish Mehta

What is this document about?

- A part of feasibility study of Component-Based Software Architecture (CBSE) at Software Engineering Institute (SEI), CMU.
- Volume I contains market assessment.
- Volume III outlines proposed course of action for the SEI.
- Volume II, this document, establishes technical foundations in CBSE.

Manish Mehta, UMKC, Jan 2004
Component-Based system

- Not a whole new concept
- Nothing but decomposing a problem and try to solve/tackle each sub-problem separately.
- *Inherent* meaning is “part-based whole”.

Component-Based Software Engineering (CBSE)

- Study of how the software component technology *should be used* in the service of an overall engineering discipline based in software components.
How to fit them all in a story?

- A software implementation that can be executed on a physical or a logical device is called a **component**. A component implements one or more **interfaces** that are imposed on it. This means that a component satisfies certain obligations, or **contract**. Contracts ensure that components can be **deployed independently**.
A component-Based system is based upon a small number of distinct component types. A component model is the set of component types, their interfaces, and specification of the allowable patterns of interactions among components. A component framework provides a variety of runtime services.

Motivation for CBD

- Independent Extensions
- Component Markets
- Reduced time to market
- Improved predictability
Vision

- CBSE is concerned with the rapid assembly of systems from components where
  - Components and frameworks have certified properties
  - These certified properties provide the basis for predicting the properties of systems built from components

Components

- Many definitions
  - Anything that can be reused.
  - Can be equated to Commercial off-the-shelf (COTS) products
  - Can be equated to units of project and configuration management
  - Can be equated to design abstraction
Definition from SEI

- A Component is
  - An opaque implementation of functionality
  - Subject to third party composition
  - Conformant with a component model

Interfaces

- Interface Abstraction and APIs
- Extending APIs to extra-functional properties
  - Specifying Behavior
  - Specifying Synchronization
  - Specifying Quality of Service
- Credentials
- Components and multiple interfaces
Contracts

- Needed because the client and the module are co-dependent.
- Contracts are of more importance in CBSE because of the premium placed on component substitutability.

Contracts in general

- Between two or more parties
- Parties often negotiate the details of a contract
- Contracts prescribe normative and measurable behaviors on all parties
- Contracts can not be changed unless agreed to by all parties.
Contracts in CBSE

- Between components
- Components must agree on any “bleed through” properties introduced by a component
- Component Certification
- Stable standards

Two senses of Contracts

- Component-contract
  - Specifies a pattern of interaction on that component.
    The contract specifies the services provided by a component and the obligations of clients and the environment needed by a component to provide these services.
- Interaction-contract
  - Specifies a pattern of interaction among different roles, and the reciprocal obligations of components that fill those roles.
In Summary

- Component-Based design is “Part-Based Whole”
- A component is an opaque implementation, subject to third party composition, and conformant to component model.
- CBSE is concerned with certified components, predicting the properties of systems and rapid assembly.
- New to CBSE: Interfaces and Contracts

Questions ??