CBD를 위한 전략

김정아

2001. 9. 8
clara@mail.kwandong.ac.kr
목차

- Impact of Component
- Why Software Component?
- What is Software Component?
- Component-based Delivery Process
- Component Technology
- Component Tools
- Component Market
- The Future of Software Component
- Success Factors for CBD
Impact of Components

Component-Based Development

- Legacy Integration
- Technology Independence
- Investment Protection
- Adaptable Applications
- Reduced Cost and Time-to-Market
- Business Service-Based Applications
- Evolution Through Plug-and-Play
- Build, Buy and Assemble
- Encapsulated Business Service Components
- Specification and Execution Standards
Why Software Component

1980s and earlier
- Organisation Focus
- Mainframe-centric
- Monolithic
- Internal Use

1980s-90s
- Business Process Focus
- Client/Server
- Monolithic
- Business-to-Business via EDI - file transfer

New Millennium
- Third-Party Service Providers
- Extranet
- Internet
- Customers

- Virtual Organisations
- Distributed
- Componentised
- E-business direct to customers - real-time transactions
Why Software Component

- **Prime Business Objectives for Applications**
  - Time-To-Market
  - Adaptable
  - Supporting Integration
  - Applicable
  - Upgradeable

- **Componentization of business application software**
  - Reusable
  - Replaceable
  - Upgradeable

- **Prime IS Objectives for Application**
  - Productivity
  - Quality
  - Reduce Costs
  - Manageable Process
  - Management of Skill
What is a Software Component

- **Essential Characteristics of Component**
  - Identifiable
  - Traceable through full development life cycle
  - Replaceable by component offering same service
  - Accessed only via interfaces
  - Services offered through interfaces must not change
  - Accurately documented service

- **Desirable Characteristics of Component**
  - Physical implementation is hidden
  - Independent of other components
  - Encapsulated
  - Reuse of services not constrained by physical implementation
  - Can be reused dynamically
  - Offer generic service that can be adapted to specific need
  - Specialized via plug points
Characteristics of Component

- It has an implementation:
  ```java
  for (int i=0; i<limit; i++)
  { list[i] = ... }
  ```

- It has a specification

- It can be packaged up

- It can be deployed

- It conforms to an standard

*Enterprise Component Standard*
Component Granularity

- **Implementation Specific Components**
  - Lowest level of abstraction
  - Class library

- **Business component**
  - Meaningful services focused on a particular business concepts
  - Accessible via the interface

- **Component frameworks**
  - Pre-build assembly of components
  - Together with glue logic to bind the components
  - Offer services through interface

- **Application Component**
  - Completed working application
  - Offer services through interface
Component Granularity

- Implementation Specific Components
  - Models
  - Class Libraries
  - Used to specify or build components

- Business Component
  - Implements a business concept
  - Service-based
  - Replaceable
  - Independent
  - Identifiable
  - Accessed via interfaces

- Component Framework
  - Collections of related components designed to be extended

- Application Component
  - Complete application used as a component
4 Tier Architecture
Component Spectrum of J2EE blueprint

- HTML pages
- Basic JSP pages & Servlets
- JSP pages with Modular Components
- JSP pages with Modular Component & Enterprise Beans

- HTML pages
- JSP pages
- Servlets
- HTML pages
- JSP pages
- Servlets
- JavaBeans Components
- Custom tags
- HTML pages
- JSP pages
- Servlets
- JavaBeans Components
- Custom tags
- Templates
- Enterprise beans
Component-based Delivery Process

 développer

Essential Characteristics of the component-based process
- Architected
- Based on reuse of services
- Enables transition
- Integrated disparate source of components
- Incremental
- Distributed
- Impacts full life cycle of Application delivery
CBD Framework
Incremental Delivery
CBD Maturity Levels

6. CBD Maturity
   Fully implemented in component technologies

5. Components are Enterprise Standard
   Enterprise-wide agreement on components

4. Components are Reused
   Ad hoc reuse of components across projects

3. Components are Business Concept
   Analysts model business concepts as components

2. Components are Implementation Concept
   Programmers/designers structure code implementation as components

1. Components are GUI controls
   ActiveX or JavaBeans components are used to implement client applications
Component Requires Architecture

- **Business Architecture**
  - determines the process component, business components, business interface

- **Application Architecture**
  - actual implementation of the business concept

- **Technology Architecture**
  - technical service by technical components & platform infrastructure
Component Technology

- **Benefit**
  - Language and platform independent component reuse
  - Runtime reuse of services
  - Location Transparency
  - Self Describing Components
  - Dynamic Scalability

- **Component Execution Environments**
  - Transaction Processing
  - Object Request Brokers
  - Message-Oriented Middleware and Message Queuing
  - Message Broker
Component Technology

- Adoption of Component Technologies
  - .NET
  - CORBA
  - EJB
Net
Component Tools
Component Tools

- **Modeling Tools**
  - support for component and interface modeling
  - improve communication between component supplier and consumer
  - facilitate reuse
  - component catalogues can use single notation (UML)
  - Enable tool data exchange

- **Build Tools**
  - developing implementation of Components
  - generation of native components from model-based specification
  - generation of interfaces for multiple component models
  - increasing use of patterns and framework
  - generation of test harness
  - wrapping for legacy environments
Component Tools

- **Repositories and Configuration/Deployment Management**
  - Inventory management for components
  - Microsoft repository
    - Open Information model to share information

- **Assembly Tools**
  - Visual specification and assembly of application
  - at the business semantic level
  - at the component technology level

- **EAI software**
  - platform integration
    - integration of different technologies across execution platforms
  - package integration
    - integrating different packaged applications
Future of Component

- Component Market
Future of Component

- Component Market
Future of Software Component

- Comparison of Maturity

![Diagram showing the future of software component maturity with axes for Modelling, Design and Build, Standards, Deployment and Execution, Client/Server, and Management. The diagram highlights CBD in the center.](image-url)
Future of Software Component

- Build & Buy
Success Factors for CBD

- A project’s success is influenced by many hygiene factors
  - New Roles/Organization
  - Training
  - Technical Infrastructure
  - Standards/Methods
  - Architecture
  - Policy