

Agile Modeling and Design of Service-Oriented Component Architecture

Zoran Stojanovic, Ajantha Dahanayake
Faculty of Technology, Policy and Management
Delft University of Technology, The Netherlands
{Z.Stojanovic, A.Dahanayake} @tbm.tudelft.nl

Related Developments

- ◆ Model-Driven Development (MDD)
- ◆ Agile Development (AD)
- ◆ CBD modeling and design
- ◆ Service-Oriented Architecture (SOA)
- ◆ Business Process Modeling

Technologies and Standards

- ◆ CORBA Components, Enterprise Java Beans, Microsoft COM+/.NET
- ◆ XML, SOAP, WSDL, UDDI, WSCL, BPEL4WS
- ◆ Besides technology we need a methodology
- ◆ For complex solutions we need a modeling and design approach
- ◆ Designing a Service-Oriented Architecture

Foundation for Service Modeling

- ◆ Component and Objects
- ◆ Unified Modeling Language (UML)
- ◆ Component-based methods
- ◆ Components first as implementation artifacts
- ◆ Components as underlying entities
- ◆ Interface-based design, design by contract
- ◆ Action semantics is needed
- ◆ Possible improvements in UML 2.0

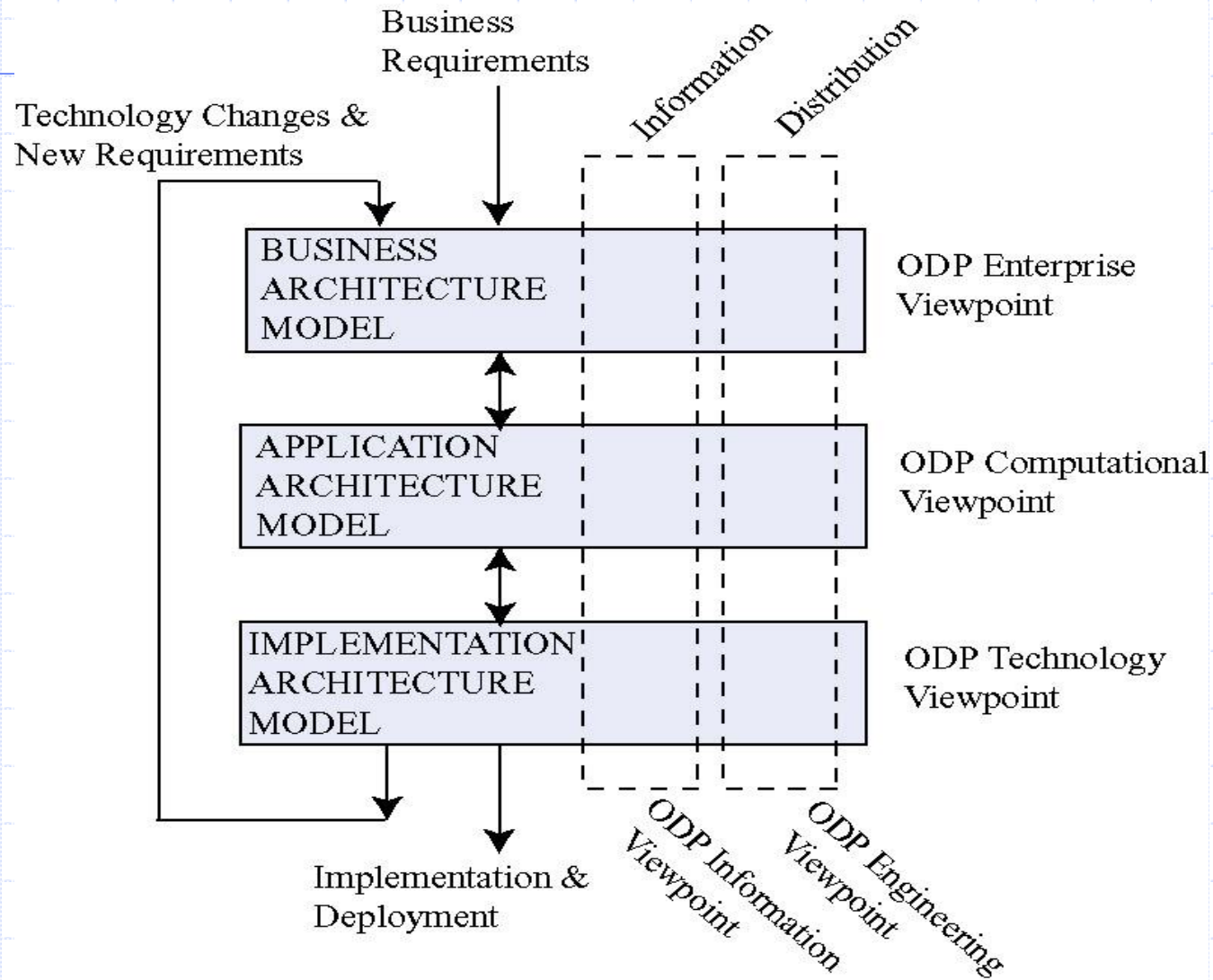
Basic Component Concepts

- ◆ WHAT and not HOW (black-box functionality).
- ◆ Loose coupling, business level granularity
- ◆ Platform-independent interfaces
- ◆ Higher abstraction than objects/classes.
- ◆ Contract, Context and Content.
- ◆ Common ground for business and IT.
- ◆ Communication between stakeholders in terms of services and components.

Contract (Interface) Concepts

- ◆ Identity (Name, Purpose, Goal)
- ◆ Behavior (operations (events) provided and required, coordination among them, pre- and post-conditions).
- ◆ Information (data types used by operations as parameters; invariants).
- ◆ Context-aware configuration parameters.
- ◆ Non-functional (quality) parameters.

SOA Modeling Approach



Business Architecture Model

- ◆ Business Service Component
- ◆ Use case diagram, Activity diagram, Domain object diagram
- ◆ Use cases are business-goal driven
- ◆ Service components are use case driven – (support use-case realization)
- ◆ Clustering services into Service Units
- ◆ SRC and CRCC cards

Application Architecture Model

- ◆ Different types of Application Service Components
- ◆ Interface Component, Logic Component, Entity Component, Data Access Component
- ◆ Coordination Manager
- ◆ Event Manager
- ◆ Business Rule Manager
- ◆ Towards Implementation Model

Conclusion

- ◆ We need both MDD and AD
- ◆ Agile principles and practices important in building SOA (ever-changing business).
- ◆ Concepts of Components and Services provide agility in MDD.
- ◆ Service-Oriented Components – a foundation for SOA design
- ◆ SOA - balancing business and technology.
- ◆ Service & Component Refactoring.