

COMPSCI 732

Software Tools and Techniques



Lecturer:
Dr SANTOKH SINGH

**Lecture 2: Advanced topics in
Aspect Oriented Development**

Lecture 2

- Aspect Oriented Components & Systems

Components, CBSE & Aspects

- Components are already well established and widely used in all other engineering disciplines.
- In recent years, software engineers and developers have started to apply the idea of using components in software design and development.

Software Components

- A component generally has three main features:
 - *(a.) It constitutes an independent and replaceable part of a software system that has a clear function to fulfil.*
 - *(b.)*
 - *(c.)*

Software Components

- A component generally has three main features:
 - *(a.) It constitutes an independent and replaceable part of a software system that has a clear function to fulfil.*
 - *(b.) It works within the context of well defined software architecture.*
 - *(c.)*

Software Components

- A component generally has three main features:
 - (a.) It constitutes an independent and replaceable part of a software system that has a clear function to fulfil.
 - (b.) It works within the context of well defined software architecture.
 - (c.) It communicates with other components through its **interface** definitions.

Why CBSE?

- Software systems have evolved to become even more complex, larger, more difficult to understand and harder to control than ever.
- This has resulted in high development costs, unmanageable software quality, low productivity and higher risks when migrating to newer technologies.
- As a result there is an urgent need for better and more efficient software development methodologies.
- One possible solution - CBSE.

Some CBSEs

- Several component-based software engineering methodologies have also evolved to try to address the development and maintenance issues.
- These CBSEs include the COMO approach, The Select Perspective and The Catalysis approach.
- Others like OMG's Model Driven Architecture have an extensive collection of tools that can generate interface definitions and application code that is vital for components.

The 4 activities involved in CBSE when dealing with COTS

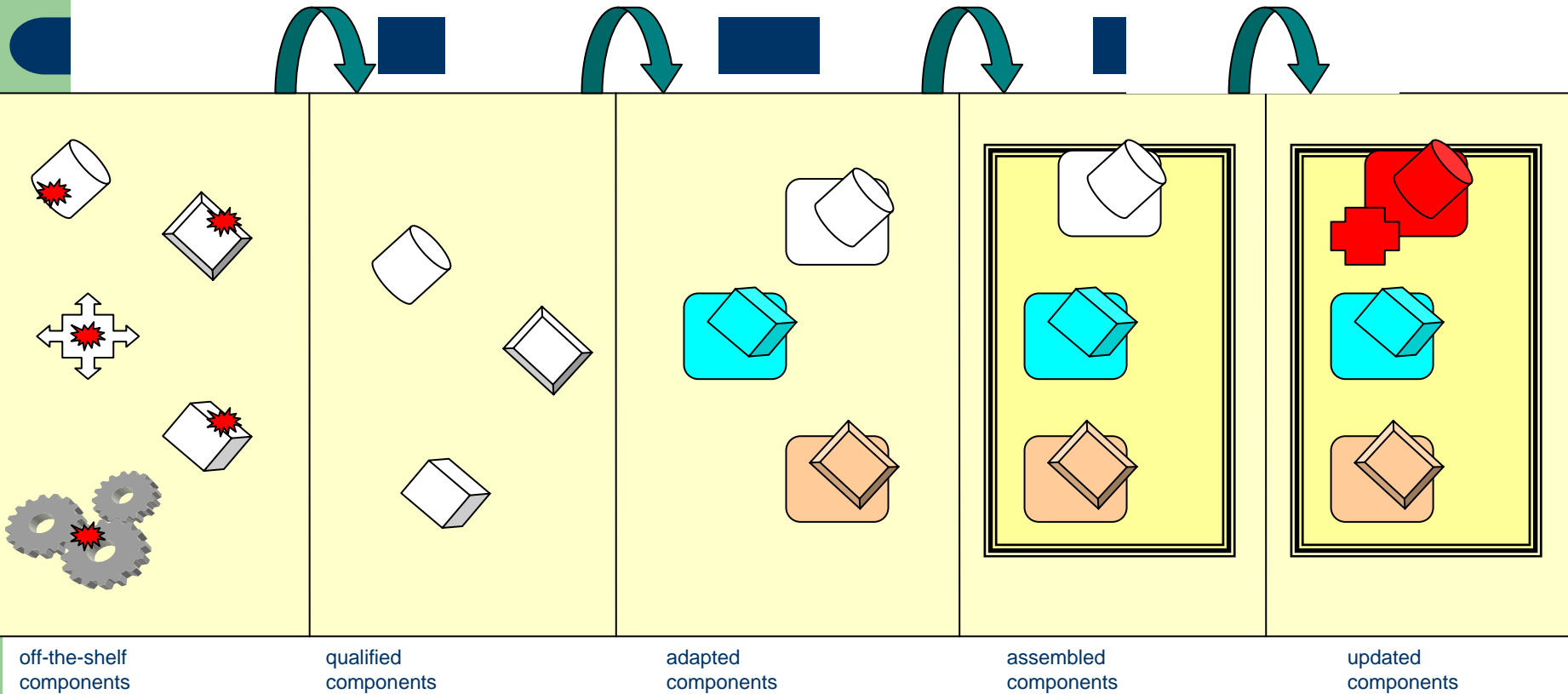
4 activities/transformations:

1. Qualifications to discover interface and fitness for use

2. Adaptation to remove architectural mismatch

3. Composition into selected architectural style

4. Evolution to updated components



CBSE with Aspects?

- Currently, even component-based software systems have evolved to become even more complex, larger, more difficult to understand and harder to control than ever due to cross cutting concerns cluttering their designs and implementations. This has resulted in high development costs, unmanageable software quality, low productivity and higher risks when migrating to newer technologies.
- As a result there is an urgent need for even better and more efficient software development methodologies!
- One possible solution is using Component Based Software Engineering (CBSE) for design and development with aspects taken into consideration.

Types of Aspects that we can consider in Components

- Security
- Transaction Processing
- User Interface
- Performance
- Persistency
- Distribution
- Resource Utilization
- Configuration
- Debugging Aspects
- Etc.

The above aspects list is non-exhaustive.

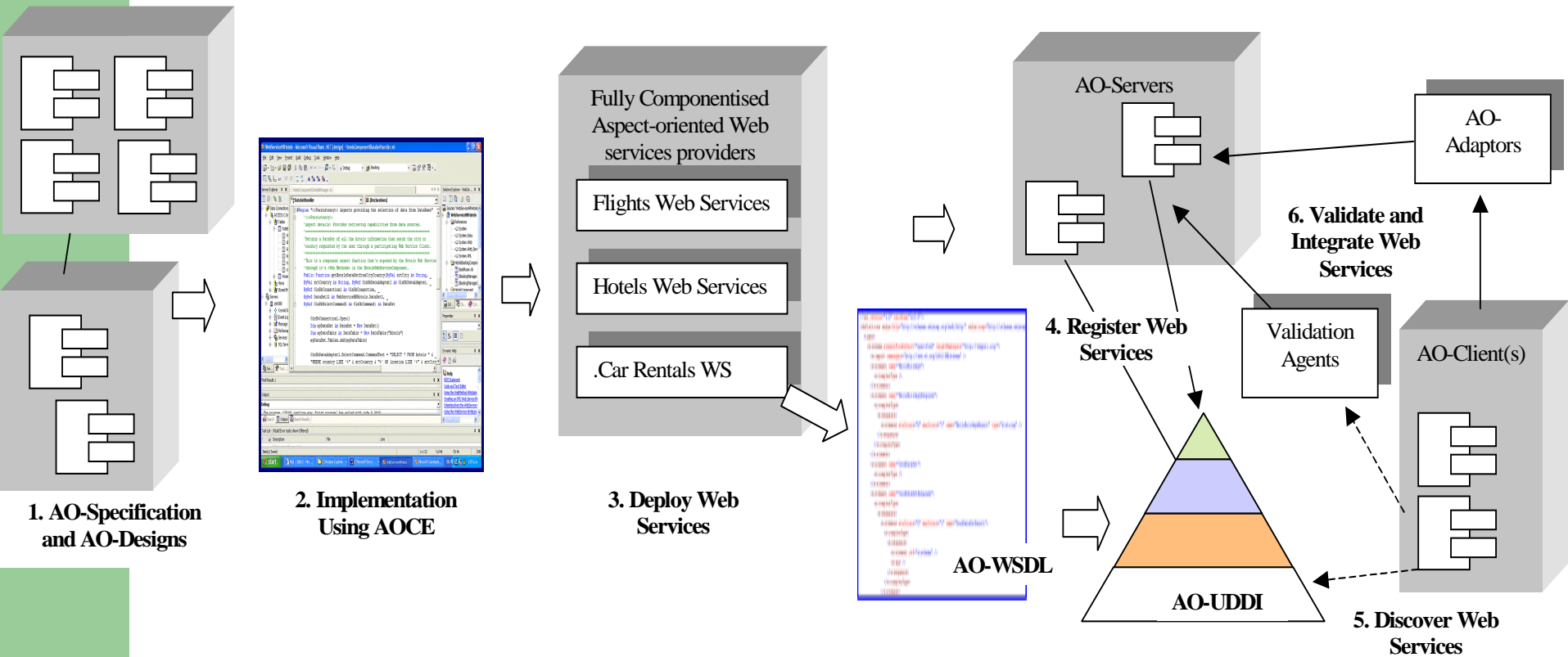
What is an Aspect-Oriented Component?

- It has four main features:
 - (a.) It is composed of one or more types of aspects.
 - (b.) It constitutes an independent and replaceable part of an aspect-oriented software system that has a clear function to fulfil.
 - (c.) It works within the context of well defined aspect-oriented software architecture.
 - (d.) It communicates with other components (both aspect-oriented and non aspect-oriented components) through its **aspect-oriented interface** definitions.

Aspect-Oriented Component Engineering

- The aim of using Aspect-Oriented Component Engineering (AOCE) is to enable software engineers develop better characterised and categorised software components.
- These aspect-oriented software components are also more efficient and reusable as compared to those non aspect-oriented ones.
- The last slides shows the whole process of applying the CBSE methodology called Aspect-Oriented Component Engineering (AOCE) to develop aspect-oriented web based software applications from inception to implementation, delivery and subsequent maintenance if necessary.

Example: Developing large web service-based Aspect-Oriented Systems



Example: Collaborative AO-Travel Planner System based on AOWS

