

# COMPSCI 732

## Software Tools and Techniques



**Lecturer:**  
**Dr SANTOKH SINGH**

**Lecture 4: Advanced topics in  
Aspect Oriented Development**

# Lecture 4

- Aspect Oriented Software Development (AOSD) across the Software Development Lifecycle (SDLC)
- A comparison of development methodologies, support tools and ideas
- Revision/Assignment 3

# SDLCs for CBSEs with Aspects

- Aspect-Oriented Component Engineering (AOCE)
- Total Component Engineering Methodology (TCEM)
- Total Components Aspect-Oriented (TCAO) methodology.
- The Object Management Group's Model Driven Architecture (MDA)
- *Other methodologies, e.g. ABC (Architecture Based Component Composition), TOPCODER etc.*

# OMG's Model Driven Architecture for Software Development

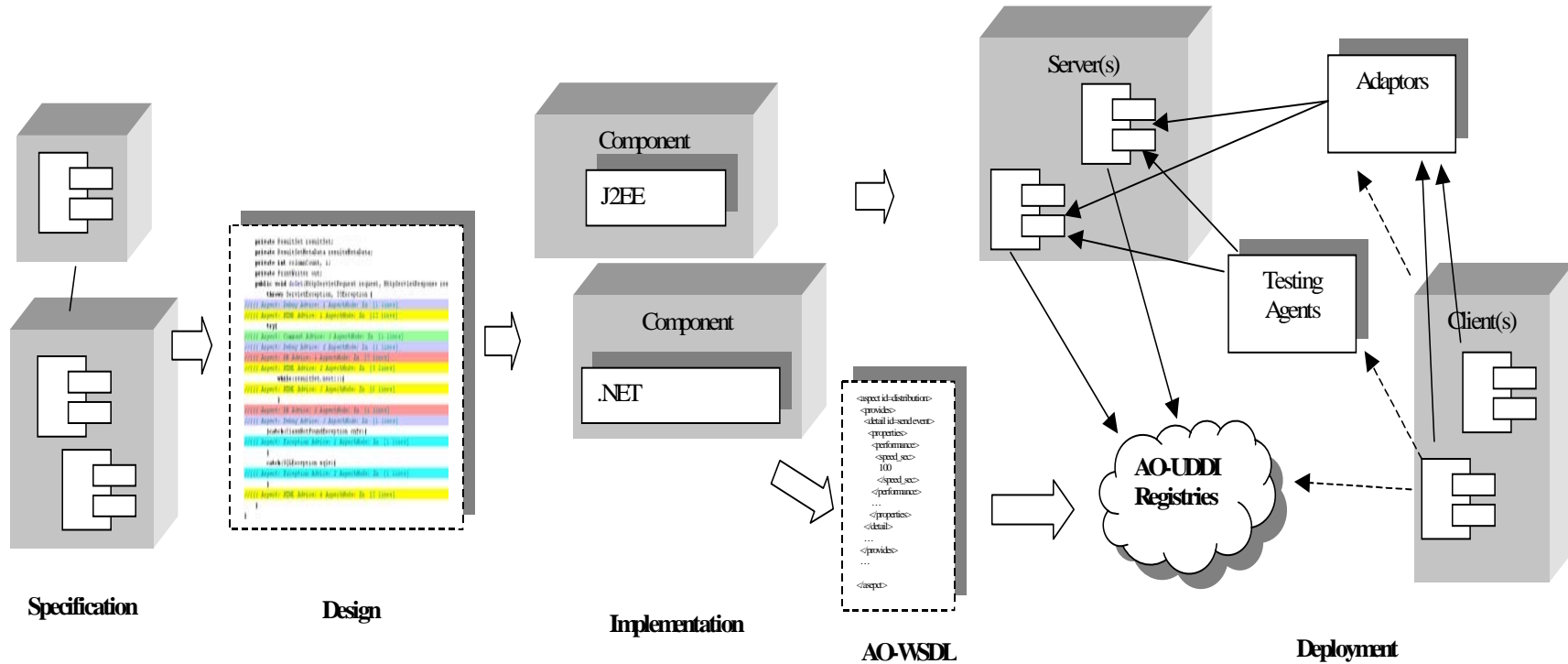
- The Object Management Group (OMG) has its own methodology for developing complex software systems. They call their approach the Model Driven Architecture (MDA).
- It has a comprehensive set of MDA tools that can be used to generate interfaces definitions and application code to construct components.
- MDA was formulated with the objective of simplifying the process of modeling, design, implementation, and integration of applications, including large and complex software systems, by defining software fundamentally at the model level. This is expressed in OMG's standard Unified Modeling Language (UML).

# OMG's Model Driven Architecture for Software Development *cont.*

- The MDA base model of the application specifies every detail of its behavior and business functionality in a technology-neutral way. This is called the application's Platform-Independent Model. Using the Platform-Independent Model, MDA tools follow an OMG-standard mapping to generate an intermediate model. This is tailored to the target middleware implementation platform. This intermediate product produced is called a Platform-Specific Model (PSM).
- The PSM reflects non-business, computing-related details e.g. those affecting performance and resource utilization that are added to the Platform-Independent Model by the web services' architects. The PSM generated from the MDA tool may not be perfect and may require some manual "tune-up" before it can be used for the next stage.
- However, the PSM is extremely detailed and contains the same information as a fully-coded application but in the form of a UML model instead of as code.
- In the final development phase, MDA tools use the PSM to generate interface definitions and application code, makefiles, and configuration files for components and the PSM's middleware platform. The components produced can also be reconfigured and refactored manually to produce the desired results.

# Aspect-Oriented Component Engineering

- Discussion Notes based on AOCE paper:
- Example of the development process for a highly distributed system:



# Total Component Engineering Methodology (TCEM)

- Main\_Component (a design pattern)
- Requirements Engineering: Use case – with components (“Early Components” )
- Multi-dimension taxonomy
  - Continuous modification (Word/text)
  - Visual taxonomy (Pictures)

# Total Component Engineering Methodology *continued*

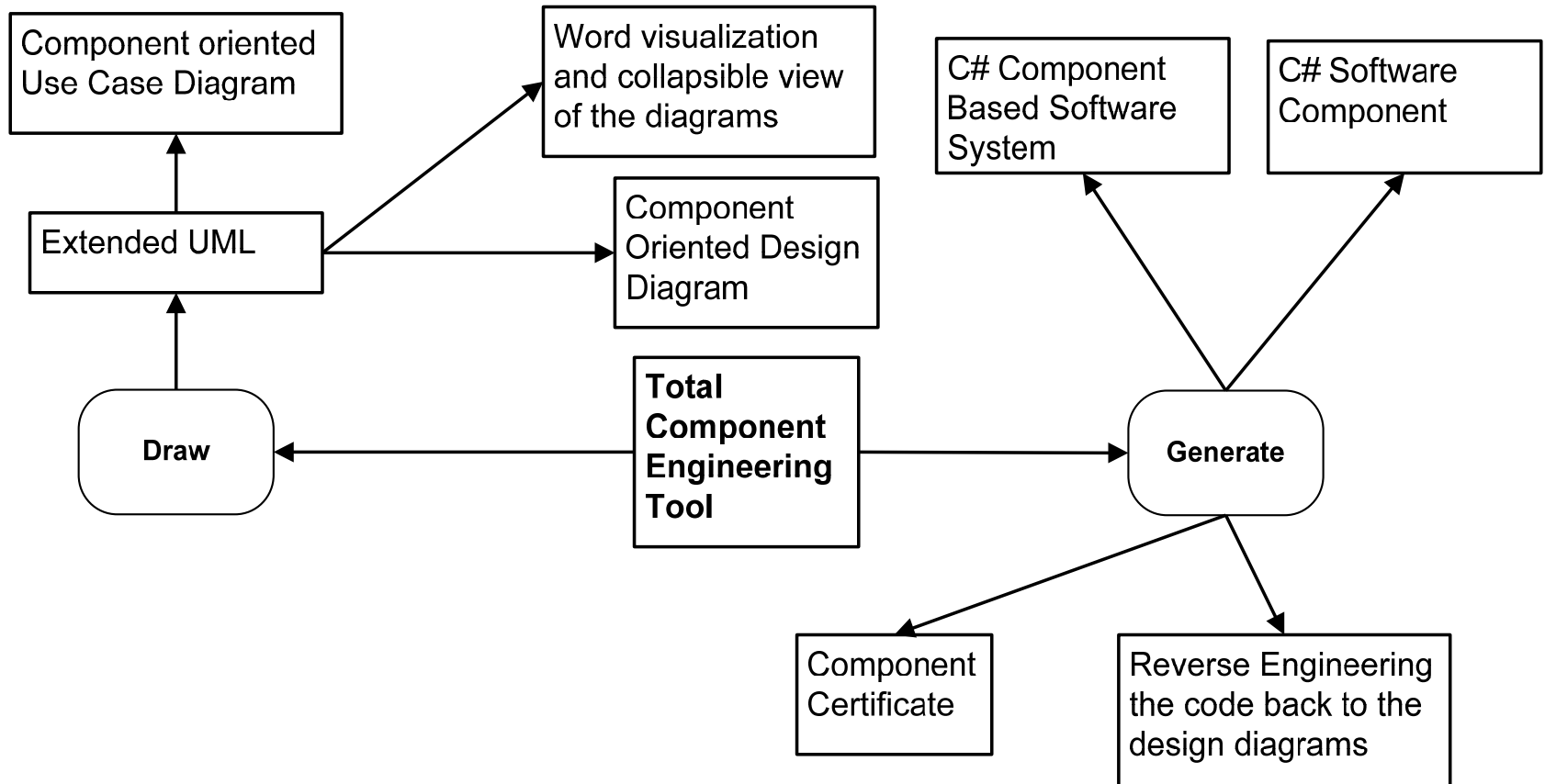
- Identify and Apply software components in every phase:
  - Early-Components
  - Thought-out all the stages of the SDLC
- Component security, including
  - Component Certificate
- Component's reusability
  - Global reuse
  - Easy to search & discover components
  - Documentation – 2 levels (one machine readable, the other for humans)



# Total Component Engineering Tool (TCET)

- Extends UML and has the following diagrams
  - Import List diagrams
  - New Component diagrams
  - Method diagrams
- Auto-Code generation
  - Generates C# software code and components
- Local component database
  - Can download and reuse components from web repository into any project
- Collapsible views
  - Hide/Show software components and entities
- Visual depiction
  - Use different images, patterns and colour to represent specific features

# TCET tool *continued*



# TCAO SDLC

Consider Aspects,  
Components & Software

TCAO start

Consider Early Aspects,  
Components & Software

TCAO  
Analysis &  
Design

TCAO  
Development &  
Implementation

Consider Aspects,  
Components & Software

TCAO Software  
Development Life  
Cycle

TCAO  
Requirement  
Engineering

TCAO Testing  
&  
Deployment

Consider Aspects,  
Components & Software

TCAO Maintenance

Consider Aspects,  
Components & Software

