

# COMPSCI 732

## Software Tools and Techniques



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

**Lecture 3: Advanced topics in  
Aspect Oriented Development**

# Lecture 3 Overview

- Tools and Technologies to support AOSD
  - TCET for TCEM  
(TCEM - Total Component Engineering Methodology TCET - Total Component Engineering Tool)
  - Total Components Aspect-Oriented (TCAO)
  - Pounamu/Marama
  - Early Component and Software Development (eCASD) Tool
  - AspectJ, Spring etc. and Assignment 3

# Resource for extra Reading

## TCET for TCEM

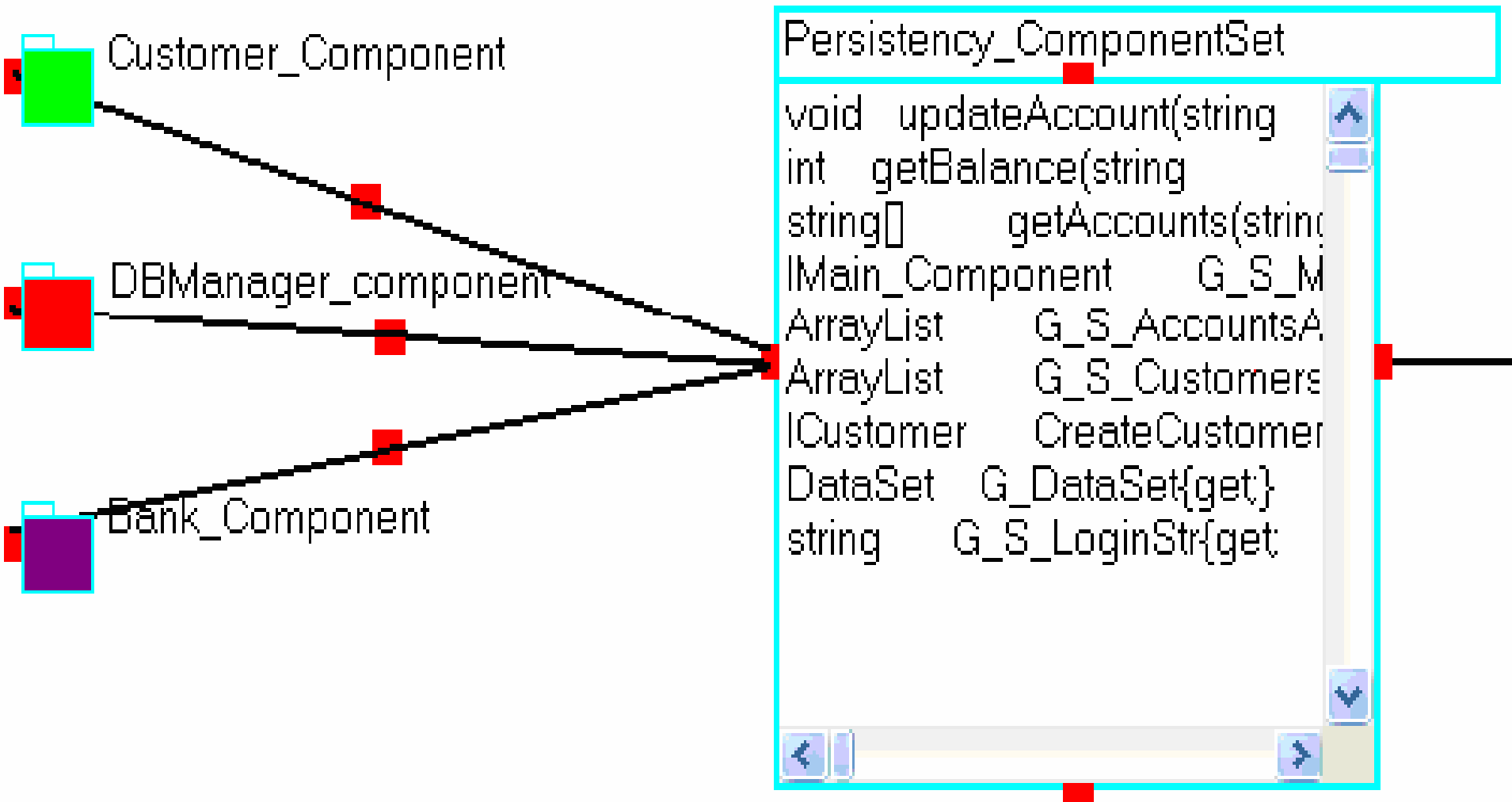
- TCEM: Total Component Engineering Methodology
- TCET: Total Component Engineering Tool
- ***APSEC 2006 Conference Paper***
  - **“Efficient Component Based Software Engineering using the TCEM Methodology and the TCET Tool”**
  - TCEM supports the development of component based software system starting from the early stages using Early Components right through all the development phases.
  - The TCET tool fully supports and complements TCEM in developing comprehensive software systems and can also address the problem of cross-cutting issues called aspects in software components and systems.

# System's Requirement Engineering with TCEM

- For the Total Component (TC) Software Requirement Engineering phase or TCSRE, we introduced a new concept called “Early Components” (EC) in the requirement engineering phase itself.
- Currently the early aspects concept has been used successfully for the aspect-oriented software development. Rather than just decide on the type of aspect early in the development phase, we believe that it is a good idea to identify and determine (if possible) the component during the early phases of TCSRE.
- The “Early Components” concept requires developers to identify any potentially useful software components e.g. authentication component, customer component etc. in the use case diagram of the CBSS during the requirement phase itself.
- By applying the “EC” concept the overall view of the software system become more visible and clearer even in the early development stages.

# TCET tool: Persistency\_ComponentSet and the related components

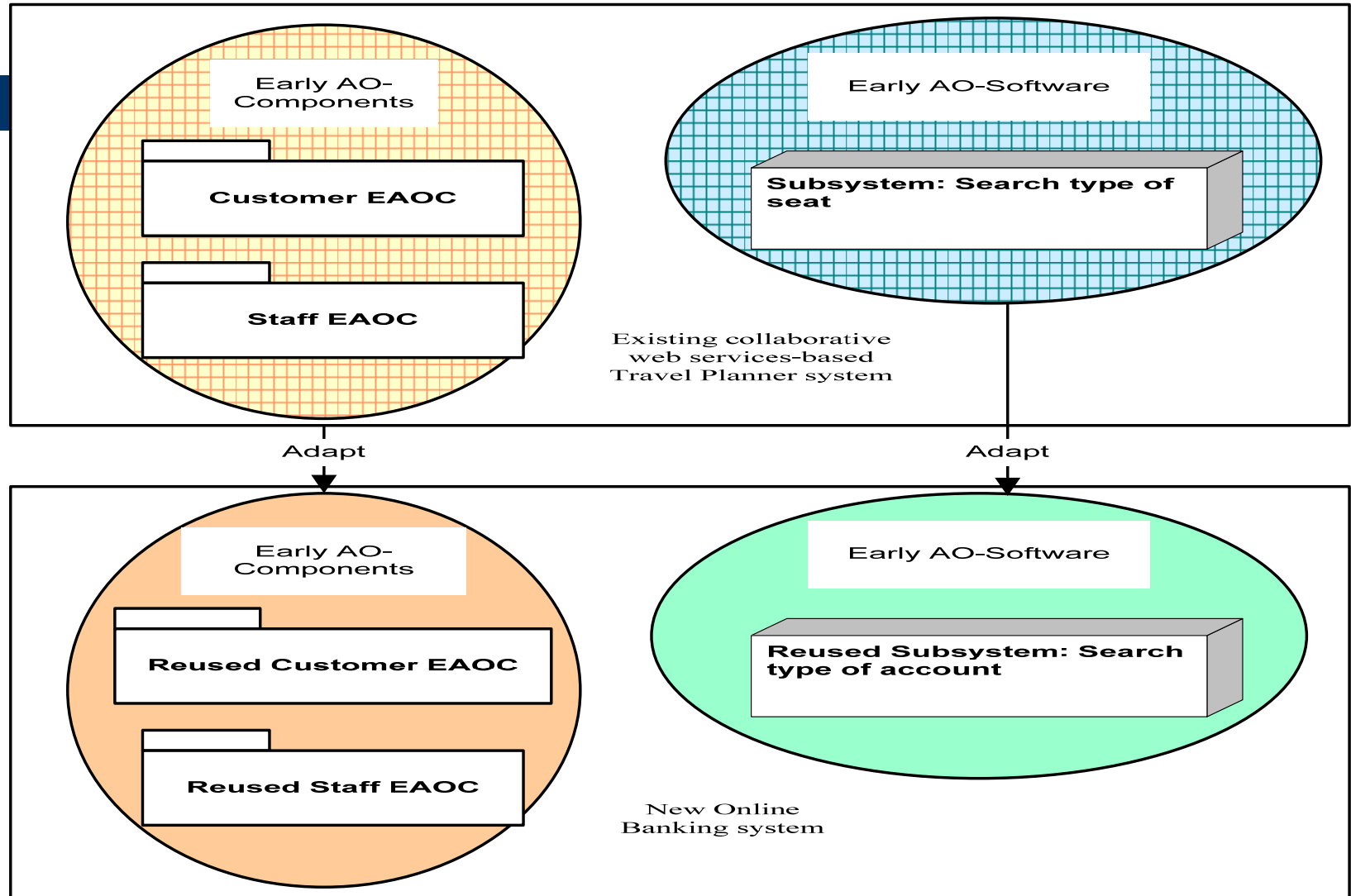
(from APSEC 2006 Conference Paper)



# Total Components Aspect-Oriented (TCAO)

- TCAO – a development methodology to efficiently and rapidly design and develop reusable distributed systems of any size, functionality and complexity.
- Uses Early Aspect-Oriented Components (EAOC) and Early Aspect-Oriented Software (EAOS), collectively known as early systemic-entities, to get an early head-start in the development process starting from the requirements engineering phase itself and spanning throughout all the phases.
- Refer to resource paper provided.

# Early AO-Components and Early AO-Software in TCAO

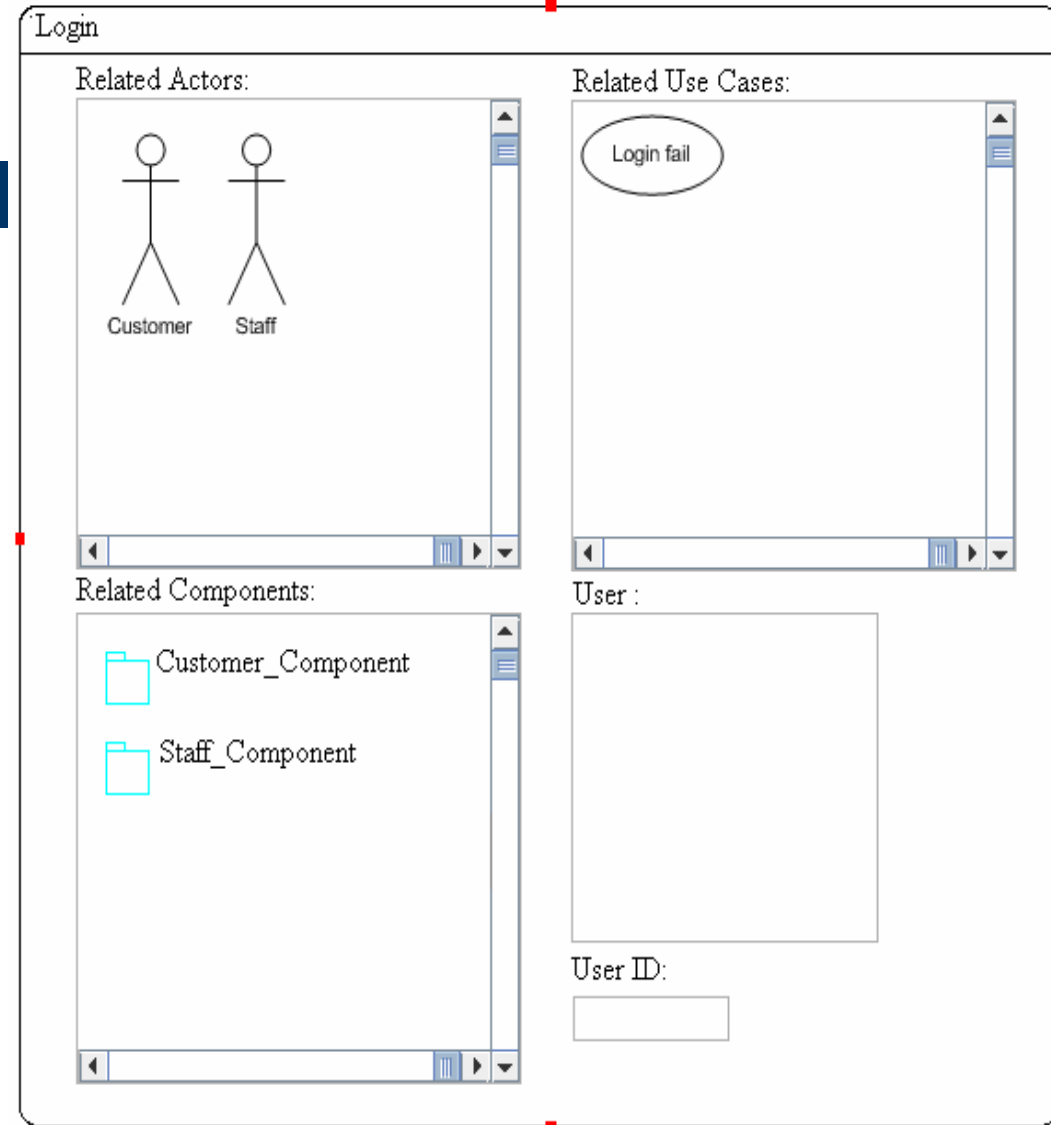
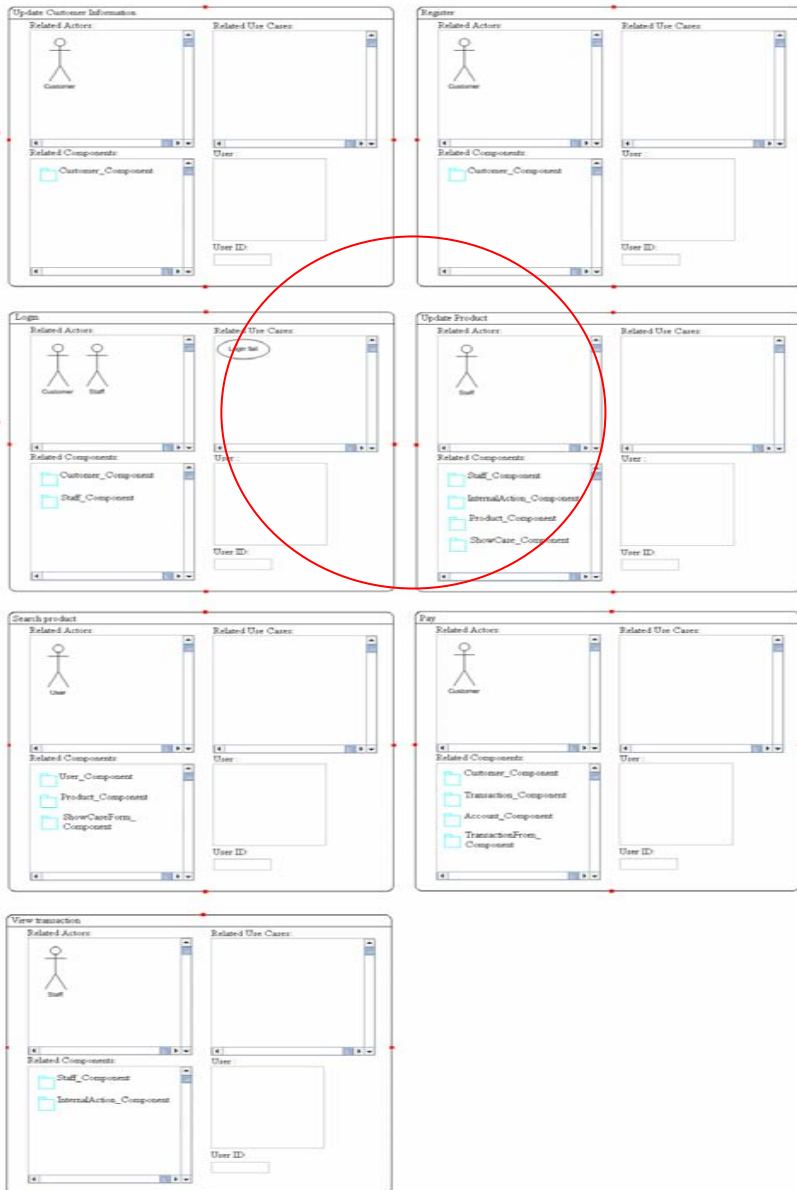


# eCASD Tool

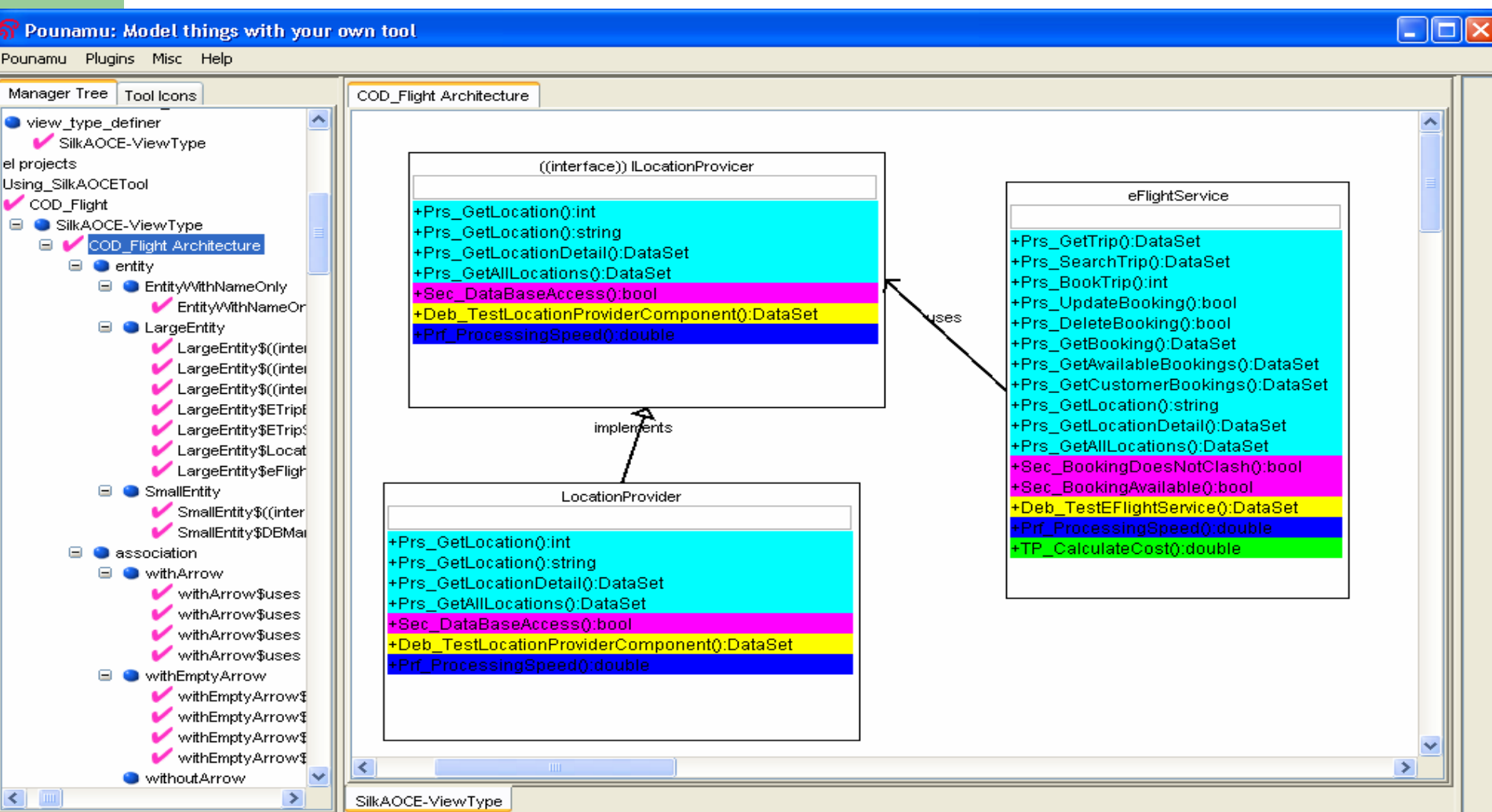
- Early Component and Software Development Tool (eCASD)
  - Allows for easier and more systematic development of component based software systems
  - More practical, accurate and efficient
    - New views
    - Novel diagrams
    - Code generation
    - Reverse engineering
    - Testing



# Early- component view using eCASD



# Depicting aspects visually in designs using Pounamu/Marama



general | undo list | redo list

22: withEmptyArrow\$implements has been saved to C:\Santokhs Copy of Pounamu\_Stuff\_13\_07\_2005\13\_07\_2005\pounamu\models\Using\_SilkAOCETool\COD\_Flight\association\_objects\withEmptyArrow

23: withEmptyArrow\$implements has been saved to C:\Santokhs Copy of Pounamu\_Stuff\_13\_07\_2005\13\_07\_2005\pounamu\models\Using\_SilkAOCETool\COD\_Flight\association\_objects\withEmptyArrow

24: COD\_Flight.xml has been saved to C:\Santokhs Copy of Pounamu\_Stuff\_13\_07\_2005\13\_07\_2005\pounamu\models\Using\_SilkAOCETool\COD\_Flight\COD\_Flight.xml

# Auto-Code Generation using Pounamu/Marama

The screenshot displays the Pounamu software interface. The title bar reads "Pounamu: Model things with your own tool". The menu bar includes "Pounamu", "Plugins", "Misc", and "Help".

The "Manager Tree" on the left shows a hierarchical structure of model components, including "handler\_definer", "model\_handler\_de", "model\_event", "model\_user", "visual\_handler\_de", "visual\_event", "visual\_user", "meta\_model\_definer", "SilkAOCEMeta\_M", "entity type", "association ty", "view\_type\_definer", and "SilkAOCE-ViewT". A context menu is open over the "SilkAOCE-ViewT" component, listing actions such as "rename this model", "view event triggering model handlers registered for this model", "save this model", "export this model", "close this model", "delete this model", "generate C# Code(s) for this model", and "generate AO-WSDL for this model".

The main workspace shows a UML class diagram titled "SilkAOCE-ViewType\_0". It features three classes:

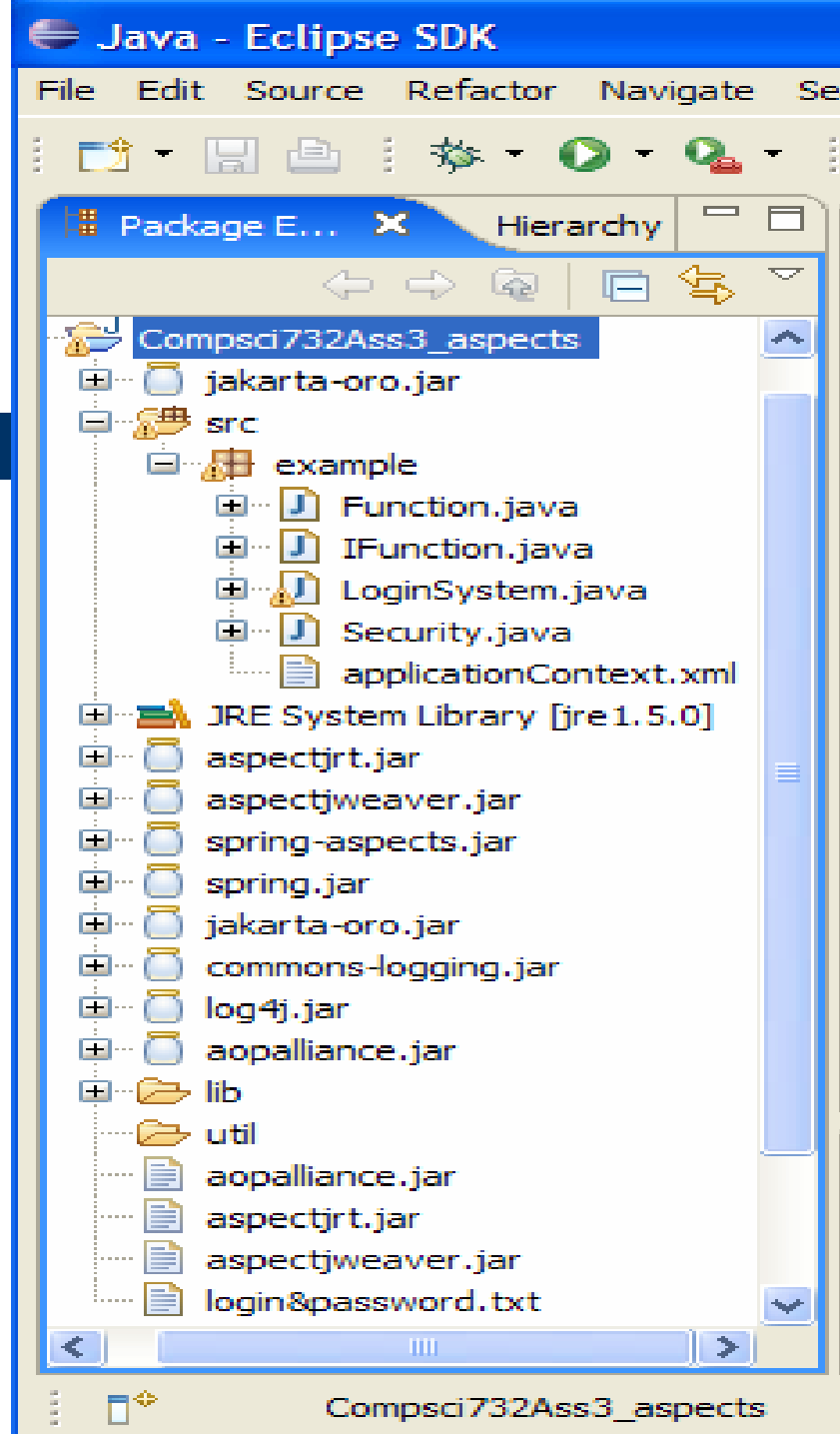
- HotelsDataManagement**: An interface class with three methods: `+Prs_HotelsDataSetfromCityCountry():DataSet`, `+Sec_AuthenticateUser():bool`, and `+Prf_InsertCustomerInfoIntoRoom():float`.
- HotelsDataManagementImpl**: A class that implements the `HotelsDataManagement` interface, containing the same three methods.
- HotelsDataBase**: A class that inherits from `HotelsDataManagementImpl`.

The diagram shows an "implements" relationship between `HotelsDataManagementImpl` and `HotelsDataManagement`, and an inheritance relationship between `HotelsDataBase` and `HotelsDataManagementImpl`.

The status bar at the bottom displays the text: "Welcome to use Pounamu".

# Assignment 3 Notes

- Eclipse IDE
- The directory structure of the software application of the 3<sup>rd</sup> assignment is shown on the right.



# 3<sup>rd</sup> Assignment Context file:

*src -> example.applicationContext.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans
```

```
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/aop
  http://www.springframework.org/schema/aop/spring-aop.xsd">
```

- <aop:config>

```
  <!-- to do.....-->
```

- </aop:config>

```
  <bean id="security" class="example.Security"/>
```

```
  <bean id="function" class="example....."/>
```

```
</beans>
```

# Something interesting about Innovations & AOP

- ***Online Article title: “The Most Important Software Innovations”***

<http://www.dwheeler.com/innovation/innovation.html>

To be a "software" innovation, it has to be a technological innovation that impacts how computers are programmed (e.g., an approach to programming or an innovative way to use a computer).

*If you look at the Appendix where they mention “**Software Innovations Being Considered**”, one of the innovation is our subject matter☺:  
**(Item no. 11) Aspect-oriented programming***

## Very Important Notice

---

- If you missed any class – please see me urgently or ask your friends who attended.
- Last class tomorrow – includes revision – please do attend.
- Please be clear, structured and systematic in answering your assignment.