

## **Software Architecting (2II10) 2005/2005**

### **Assignment**

The assignment of the course Software Architecture (2II10) is a case study, where you apply the techniques that you learned in the lectures. The assignment and its deliverables are described in this document.

### **Description**

Your team is chosen to create the architecture for the new Customer Relationship Management System (CRMS) of TOTAL COVER Insurance Company. (for a detailed description of the case refer to the file TC\_case.doc)

The desired functionality is described in the case study description. Additionally pay attention to extensibility, maintainability and scalability of your solution!

### **Deliverables**

#### **Requirements Document**

Due date: September 20<sup>th</sup>, 2005

- The description of the new CRMS for the TOTAL COVER Insurance Company states is not a requirements document.
- Based on this description, write a requirements document that meets the criteria discussed in the lectures (tractable, SMART,...)
- State any assumptions that you make!
- Limit to no more than 6 pages.

#### **UML Model**

Due date: October 18<sup>th</sup>, 2005

- Describe the architecture of the system in the UML model
- Use the Poseidon UML modeling tool, version 3.1. Gentleware provides a free 'Community Edition' version. Registration is required, but no fees must be paid.
  - <http://www.gentleware.com/index.php?id=ce>
- Make use of (at least)
  - Use case diagrams
  - Class diagrams
  - Sequence diagrams
- **Deliver a ZIP archive containing the 2 files:**
  - *ZUML of your model (Poseidon's native format)*
  - *XMI of your model*
  - The filenames must be your group number.
- *Hint (for the expected level of granularity): our 'reference model' has about 25 classes distributed over several class diagrams (this is sort of a medium value, models containing more or less classes are surely possible).*
  - *The UML model describes your solution in more detail than the ADD (Architecture Design Document) does.*

## Completed Questionnaire about UML modeling

Due date      October 18<sup>th</sup>, 2005

- A short questionnaire will be handed out to you. The purpose is to gain insight into the experiences that you got during architecting the UML model. Your answers to the questionnaire will not influence the grades not relevant for your grade (but is must be submitted).

## Logbook about UML modeling

Due date      October 18<sup>th</sup>, 2005

- Track the time you invested in creating the UML model.
- Report the time in the ExCel template.
- Submit one Logbook per team (containing the recorded time for all team members).
  - Hint: for your convenience, you can keep individual logbooks during the project and merge them before submitting.
- The recorded time is not relevant for your grade. But the completed logbook must be submitted.

## Architecture Design Document

Due date:      October 25<sup>th</sup>, 2005

- The ADD is NOT simply a collection of the UML diagrams in a word document.
- The ADD
  - lists the key design decisions and their motivations (rules, guidelines and principles proposed for the design of the system)
    - discuss the design decisions using the techniques from the lectures.
    - discuss in particular the non-functional properties (where appropriate, argue using calculations (that might be based on assumptions))
      - Maintainability
      - Extensibility
      - Scalability
  - is a draft of the architecture in UML using at least the following views:
    - structural view
    - dynamic view
- State any assumptions that you make!

### **Note:**

- *Work on the deliverables in parallel!*
- *Don't wait to begin with one deliverable until you have finished another deliverable that has an earlier due date. Otherwise you'll run out of time.*
- *Divide the work in your team!*

## Use Modeling Standards!

Modeling standards are conventions within an organization (your team) for building UML models. They are similar to coding standards in programming. Adherence to modeling standards by all team members leads to a homogeneous style of using the UML. As a

result, it is easier to read the model, especially for someone who is familiar with the modeling standards. Additionally modeling standards improve the quality of the UML model by helping you to avoiding mistakes in the model.

The modeling standards for your team are studyweb folder “Assignment”.

- ***When developing the UML model, you must adhere to the modeling standards.***
- Read the modeling standards carefully. If something is unclear to you, ask us.
- Review the model at several stages to ensure adherence to the modeling standards.

## HowTo's for using Poseidon UML

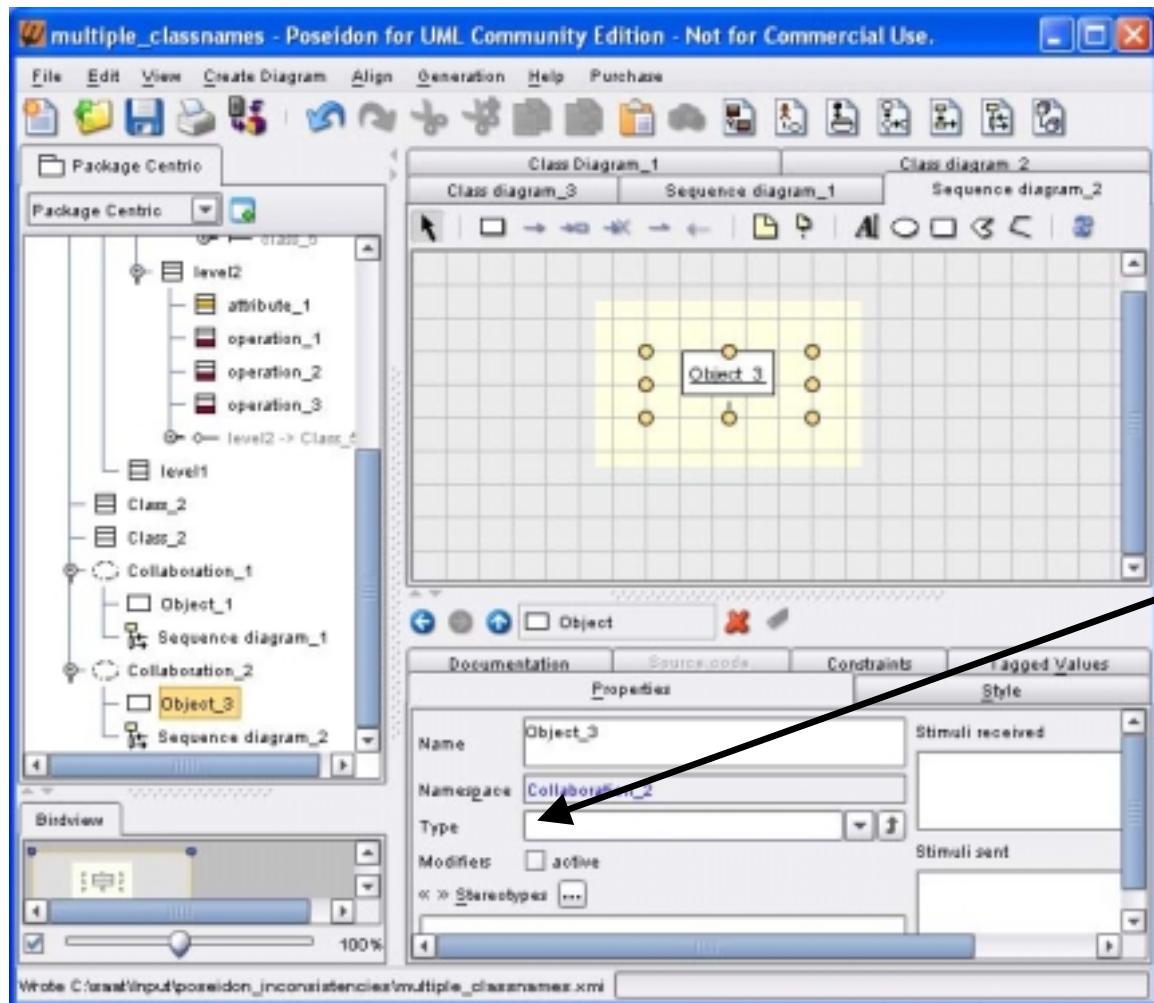
Poseidon is a CASE tool for developing UML models. It supports different diagram types (e.g. use case diagram, class diagram, sequence diagram...). The UI is very user friendly and most functionality is pretty much straightforward. You should consult its help function and manual. For serious problems, contact us.

In the following sections we briefly describe some functionality that you need and is not straightforward.

### ***Exporting the XMI file***

XMI is the XML based model interchange format for UML files. XMI enables several modeling tools and analysis tools to exchange models, which is a big benefit. Create the XMI file of your model in Poseidon via “File → Export Project to XMI...”.

### ***For Sequence Diagrams: Relating Classes and Objects***

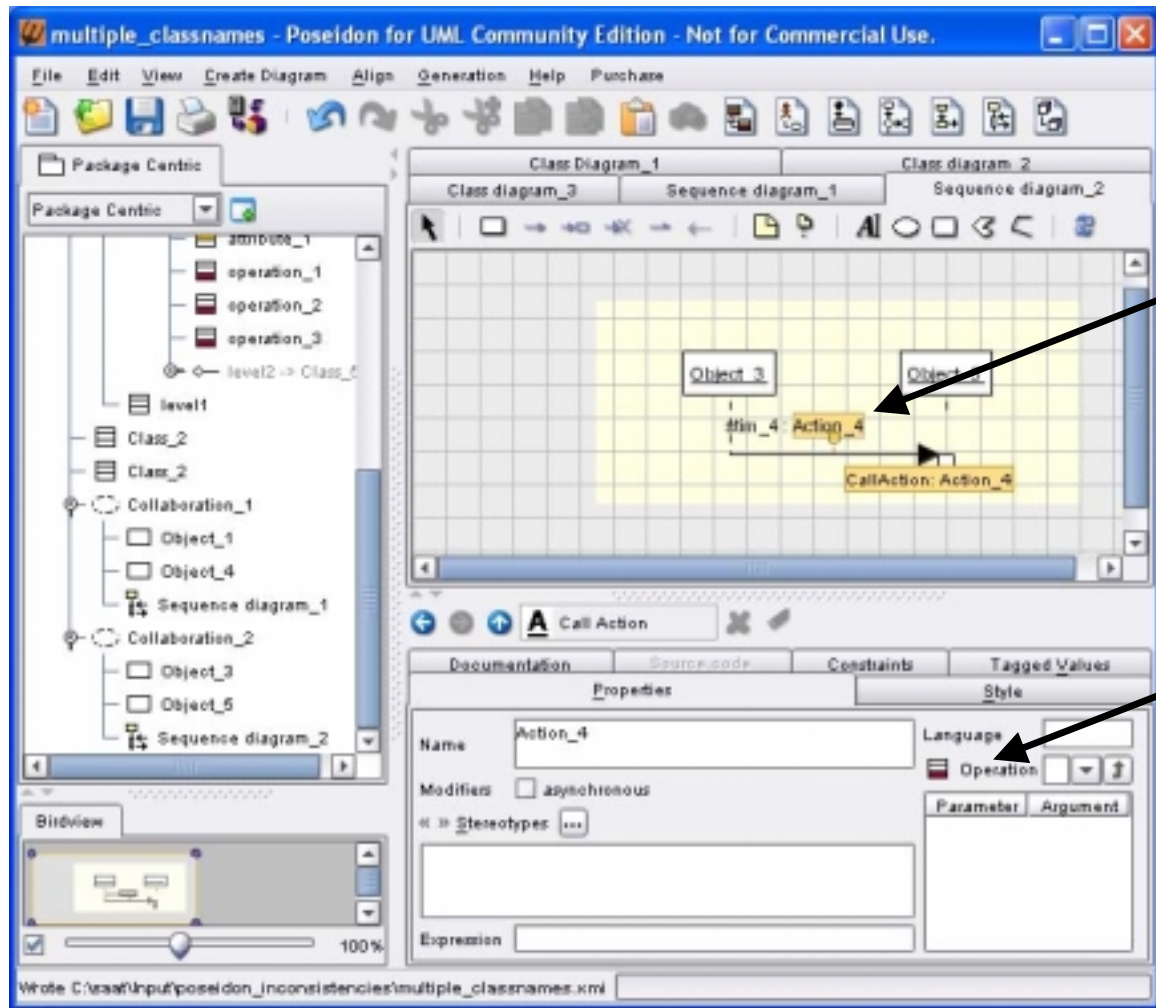


Objects represent instantiations of classes in sequence diagrams. Hence, an object (can) have a type.

When you created an object in the sequence diagram, you can specify the class (its type) as described below:

- Select the object
- In the object's *Properties* tab you can choose the *Type*.
- Choose a (existing) class.

### ***For Sequence Diagrams: Relating Operations and Messages***



Similar to the relation between classes and objects is the relation between messages and operations (i.e. the message's actions).

When you created a message in the sequence diagram, you can specify the operation (which is an operation of the called class that is represented by the called object) as described below:

- Select the messages *Action*
- In the action's *Properties* tab you can choose the *Operation*.
- Choose an (existing) operation.