Information on the lab

The deadline for the report is Monday 2006-09-18, 8.00

Important notes:

- You should solve the laboratory assignments in pairs.
- You have to perform the inspection individually, i.e. not in a group!
- You may use whatever tools you want to create documents and diagrams. For the diagrams we actually suggest pencil and paper.

Process:

1. Apply for a partner on the web. A link is provided on the course home page. You will receive an automatic email with contact information of your partner.
2. Produce the result report according to the guidelines later in this document. Make two copies of your solution.
3. Apply for an inspection pair on the web. A link is provided on the course home page. You will receive an automatic email with contact information of your inspection pair.
4. Meet with your inspection pair and give a copy of your result report to each member. Check quickly that the result report you are supposed to inspect is acceptable, i.e. does it contain all the required parts and the author's names, group and date? If the result report is not complete, ask the authors to update it and meet again.
5. Produce your inspection report, that is fill out the three checklists found later in this document (indivudually, not in a group).
6. Sign your for an inspection meeting on the list at the pinboard outside room 5B403.
7. Perform the inspection meeting with the teacher as moderator. The whole inspection group (i.e., two pairs) have to be present at the meeting. You have to bring you filled in checklists.

Unified Process Document Version Control System

Introduction

You recently graduated from university with a degree in computer engineering. Luckily, you got employed immediately by a company that develops tools to support software development processes. Recently, the development of a document version control system for the Unified Process was commenced. Key elements of the tool are:

- The artifacts (i.e., the documents) produced according to the Unified Process are stored in a central repository. The artifacts get version numbers.
- Users log in to check in (i.e., store) and/or check out (i.e., retrieve) artifacts from the central repository.
- The system respects the users' roles (e.g. the system analyst can check in use-case models whereas the component engineer cannot) and the current workflow (e.g. the use-case model cannot be checked in during design workflow).

Coincidently, your employer uses a customized instance of the Unified Process to develop the tool. The developed completed already two iterations.

Tasks

You are assigned to contribute to the extension of the tool in the forthcoming third iteration. You are supposed to specify, analyze, and design two use-cases, namely *Check In* and *End Activity*:

- The use-case *Check In* enables a worker to check in an artifact. You should carefully consider when and who may check in which artifacts.
- The use-case *End Activity* enables a worker to end a workflwo (i.e., an activity) and to start the next subsequent workflow (according to the UP, e.g. analysis followed by design). You should carefully consider when and who may end which workflow.

In more detail your tasks are to:

Prepare
Make yourself familiar with the use-case, analysis, and design models for the Unified Process Document Version Control System.

Extend the use-case model
Create a use-case model for the two above mentioned use-cases. In particular, create the following documents and diagrams:

- A use-case diagram containing the existing and new actors and use-cases.

- Two use-case descriptions, each consisting of a brief description, an activity diagram describing the flow of events, a precondition, a postcondition, and special requirements.

Remember that you should explicitly state who (which worker role) and when (in which workflow) which workflows may be ended and which artifacts may be checked in.

Extend the analysis model

Create an analysis model for the two above mentioned use-cases. In particular, create the following documents and diagrams:

- An analysis class diagram containing the existing and new classes.
- Two analysis use-case realizations (one for each of the above use-cases). Each use-case realization is supposed to consist of a collaboration diagram, a description of the event flow, and special requirements.

Extend the design model

Create a design model for the two above mentioned use-cases. In particular, create the following documents and diagrams:

- A design class diagram for the server package containing the existing and new classes (omit existing attributes and operations).
- Two design use-case realizations (one for each of the above use-cases). Each use-case realization is supposed to consist of a sequence diagram.
- A document briefly describing each new operation.

Checklist for a Use-Case Model Extension

Item	**Pass** **Fail** **Partial**	**Comment**
Do the symbols used in the use-case diagram conform to the UML standard?		
Are the brief use-case descriptions complete, clear, and concise?		
Are the use-case pre- and postcondition complete and unambiguous?		
Are the use-case descriptions consistent with the use-case pre- and postcondition?		
Do the symbols used in activity diagrams conform to the UML standard?		
Are the activity descriptions complete, clear, and concise?		
Is the use-case model extension free from unnecessary detail with respect to the way the system is going to be implemented?		

Checklist for a Analysis Model Extension

Item	Pass Fail Partial	Comment
Do the symbols used in the class diagram conform to the UML standard and UP?		
Do the symbols used in collaboration diagrams conform to the UML standard?		
Are the classes in the class diagram consistent with the use-case realizations (or are there too few or too many)?		
Are the associations in the class diagram consistent with the use-case realizations (or are there too few or too many)?		
Are the event flow descriptions complete, clear, and concise?		
Are the event flow descriptions consistent with the collaboration diagrams?		
Are the use-case realizations fulfilling the requirements expressed by the corresponding use-case?		

Checklist for a Design Model Extension

Item	**Pass** **Fail** **Partial**	**Comment**
Do the symbols used in the class diagram conform to the UML standard?		
Do the symbols used in sequence diagrams conform to the UML standard?		
Are the classes in the class diagram consistent with the use-case realizations (or are there too few or too many)?		
Are the associations in the class diagram consistent with the use-case realizations (or are there too few or too many)?		
Are the operation descriptions complete, clear, and concise?		
Are the operation descriptions consistent with the sequence diagrams?		
Are the use-case realizations fulfilling the requirements expressed by the corresponding use-case?		