Asg.2 continues from Asg.1 and applies to the same problem domain – the development of a Time Logging software tool. The “reference” tool is Time Logger from Responsive Software. The website (http://www.responsivesoftware.com/) contains a detailed description of the Time Logger’s functions.

Special requirements

Start the modeling tasks by accessing the Asg1 solution, i.e. either the solution developed by yourself or the solution provided by your instructor. This document, when applicable, refers to the solution available to the instructors in the Instructors Area on the RASD 2/e website.

Note that as Asg.1 contained five tasks (but the solution to Task 2 was included in the assignment paper), the tasks of Asg.2 start with Task 6.

As you know by now, in Rose you can create multiple class diagrams for the same model (project) but the diagrams are integrated and based on the same repository. This means that if you have a class, say Employee, then if you modify it in one diagram, the changes will automatically show up in other diagrams that contain the same class. Like in Asg.1, please name classes (and, if necessary, other modeling entities) with numbered suffixes. For example, to have various versions of class Employee, name them Employee6 in Task 6, Employee8 in Task 8, etc. If you need multiple versions in the same task, then you can use names such as Employee6a, ...

Task 6  (? marks)

Refer to the following two windows of Responsive Software’s Time Logger. In the solution to Asg.1, a class called PStopwatch modeled the window on the left. The data content of this window (i.e. the data in the window’s fields) was placed in PStopwatchBean class. Similarly, a class representing the window on the right can be named PTimeRecEdit and a class for its data content can be named PTimeRecEditBean.
6a Construct a class model for `PStopwatchBean` and `PTimeRecEditBean`. Use generalization (implementation inheritance). Show visibility of attributes. Explain the model. Can you think of another solution with generalization? Explain pros and cons of various solutions.

6b Construct an alternative class model for `PStopwatchBean` and `PTimeRecEditBean`. Use aggregation instead of generalization. Show visibility of attributes. Explain the model. Can you think of another solution with aggregation? Explain pros and cons of various solutions.

Task 7  (? marks)

Refer to Task 6. Assume that there is a need for classes `PStopwatchButton` and `PTimeRecEditButton` to represent the action buttons on the two windows (customarily, classes are given singular names, although in this case a plural name, such as `PStopwatchButton`, may seem more appropriate). Consider only push buttons (such as Panel or OK) but do not consider “edit” buttons indicated by the plus, pencil and minus icons. Consider also combo boxes (i.e. single down arrows, such as next to Client field).

7a Construct a class model for `PStopwatchButton` and `PTimeRecEditButton`. Use abstract class(es) and implementation inheritance. Show visibility of methods. Explain your model.

7b Construct an alternative class model for `PStopwatchButton` and `PTimeRecEditButton`. Use interface inheritance. If you find it gainful, you can also use abstract class(es) and implementation inheritance. Show visibility of methods. Explain your model.

Task 8  (? marks)

Refer to the solution to Task 3 in Asg.1. Consider the modified excerpt from the Task 3 class model in which the association is replaced by the composition relationship, as in the model below. Consider the overall functionality of Time Logger, as delineated (for example) by Task 1 in Asg.1.

Construct a statechart diagram for class `TimeLogRecord`. Explain your model.
Task 9  (? marks)

Consider a UX scenario when double-clicking on a timelog record in the primary window of Time Logger (as below) brings up a Time Log Editor window as shown in Task 6. Suppose that the user wants to modify the End Time to a later time and that the system rejects such modification if it would overlap with the Start Time of the next timelog record. Consequently, the user presses Cancel in order to return to the primary window.

9a Identify the UX elements in the above scenario and design a behavioral UX collaboration for the scenario. Explain your model.

9b Construct a structural UX collaboration model corresponding to the behavioral UX collaboration in Task 9a.

Note that Rational Rose does not provide explicit support for the UX modeling. Please use UML notes to represent stereotypes and tags that you cannot properly show on the diagrams.

Task 10  (3 marks)

Using Rational Rose, refer to the Help chapter entitled: “Rose Data Modeler”. Refer to the entity class diagram in the answer to Task 3 (Asg.1)

10a Refer to the Help section: “Transforming an Object Model to a Data Model – DM” and transform the entity class diagram into a relational database schema. Use the schema to create a data model diagram. Has Rational Rose done a good job? Discuss the obtained diagram.

10b Ignore the schema and the model obtained in Task 10a. Create a brand new schema and a data model diagram corresponding to the entity class diagram. Determine primary keys of tables, specify column properties, establish relationships between tables, and determine declarative referential integrity constraints. Explain the model.