

# Part 3 of Assignment Advanced Topics in Computer Science (2018/2019), version from 19-5-19

---

## Deadlines

The deadlines are firm and there are two possible deadlines:

- **June, 14<sup>th</sup> at 23:00.** In this case, students are called to discuss the different parts of the assignment on June, 19<sup>th</sup> at 2pm in room 1BC45. (Primo Appello)
- **July, 1<sup>st</sup> at 23:00.** In this case, students are called to discuss the different parts of the assignment on July, 5<sup>th</sup> at 10am in room 2BC30. (Secondo Appello)

If students want to submit to any later exam attempt (Appello), they need to contact the teacher by email to arrange it.

Note that the grade of this assignment part will only be communicated during the discussion sessions (e.g. on June, 19<sup>th</sup>).

## Evaluation

Models are evaluated based on meeting the technical requirements (i.e. does the model correctly implement the specification?) while not being overly complex.

This assignment is to be solved **in a group composed by two students**. While it is encouraged to discuss with students of other groups, the solutions have to be **original**. Solutions of the two or more groups that are almost identical will be considered as plagiarism (“copiato”), all those groups will be just given 0 points, with no possibility to repeat the assignment.

Part I and Part III will be each given a grade between 0 and 40; Part II will be given a grade between 0 and 20. The maximum sum will be 100, which will be later multiplied by 0.3 to scale between 0 and 30 and rounded to the closest integer. If the sum is greater than or equal to 99 (i.e. the average is greater than or equal to 29.7), this will lead to “cum laude”.

# Goal of the Assignment and Description of the Provided Part of the Model

Part 3 requires to:

- build a YAWL model that describes the process of managing road-traffic tickets and adheres to the requirements below;
- deploy it in YAWL and define a sufficient number of users;
- provide a report that discusses the solution proposed.

## Requirements

The management of a ticket starts when a violation is committed by an offender.

The following list describes the main requirements of the road-traffic ticket management in YAWL:

1. When a violation is committed, the *Insert Ticket* task is performed. The execution of this task consists of providing the following information: <sup>1</sup>
  - the number of the article that is violated;
  - the name and the address of the offender;
  - the amount (as a double) that corresponds to the violation.
2. After inserting the ticket into the system, the ticket is prepared and sent by post to the offender. This task requires all the information provided via *Insert Ticket* as input.
3. After preparing and sending the ticket, the management office is awaiting for receiving payments. When a payment is received, a task "*Process Payment*" is executed. This task reads the name and address of the offender and produces the amount paid as output, which needs to be added to the amount that has already been paid.<sup>2</sup>
4. When the ticket is fully paid, the ticket is automatically closed and archived.
5. After 180 days, if the offender has not paid the amount due for their ticket in full, the ticket is prepared for credit collection. This corresponds to executing an automatic task *Add Ticket for Credit Collection*, which concludes the management of the ticket.
6. If the ticket is not sent within 10 days after being inserted, it expires, meaning that the offender does not have to pay the amount related to the violation. This corresponds to executing an automatic task *Archive the Ticket as Expired*, which concludes the management of the ticket.

Define a different role for each of the manual tasks, and assign two different human resources to each role.

Note that these requirements are simplified, if compared with those of Parts 1 and 2 of the assignment.

---

<sup>1</sup> Compared with the requirements of Task 1 and Task 2, you do not need to provide the ticket number (to keep the assignment part reasonably simple).

<sup>2</sup> Differently from Task 1 and Task 2, you can keep the payments in a single double instead of keeping a list.

**Important note:** The above process description is not intended to be a step-by-step description of the process, but rather some abstract requirements. If you feel that anything is underspecified, **you must resolve this underspecification** and come up with a concrete model satisfying all requirements. **The purpose of the walk-in sessions is that you ask about your ideas and assumptions as you come up with a solution.**

### The structure of the report for Part 3

In a report briefly describe the model and your design choices. Include a diagram of the corresponding YAWL net that is as readable as possible. The purpose of the report is to relate your technical solutions in the YAWL net to the informal requirements. For this, your report should discuss every assumption that you have made which is not explicitly stated in the text. Also you should explain the task's decompositions and net-to-task-variable mapping that are not trivial (i.e. any mapping that is not just a simple one-to-one mapping). Explain how you implement the timers. Ensure that the report lists all variables that you introduced, with their type, and includes the purpose of each of them. Ideally, the report is between 3 and 5 pages, including textual description and model.

### How to submit the assignment for Part 3

- Provide the YAWL model for Part 3 as a file named "**ID[studentID]-part3.yawl**" where studentID is the student identifier of one of the two members of the group (i.e., your file should be called "ID0463033-part3.yawl", if the student id of one of two member is "0463033").
- Provide the zip file of the apache-tomcat installation of your YAWL (i.e. create a zip called "**ID0463033-part3.zip**" that contains the directory ...\\YAWL-4.2\\engine\\apache-tomcat-7.0.65 or similar)
- Provide a report, explaining the model of Part 1, as a pdf file named "**ID[studentID]-part3.pdf**" (i.e., your file should be called "ID0463033-part1.pdf", if the student id of one of two member is "0463033"). Ensure that the report clear states the name and student id of the group members.

