Rigorous Software Engineering 2019
Part 2: Alloy Modeling Task

Published: March 18, 2019
Firm deadline for submission of `Alloy` model: April 05, 2019 (23:59 CET)

Your task is to design a POS system for Roberto's Pizza Place based on the description provided in the earlier UML modeling task. The emphasis of your task is on designing the data model, formalizing it in `Alloy`, and checking whether certain properties hold for your `Alloy` model.

You should use the UML model from Fig. 1 to guide the design of the `Alloy` model (so **do not use the UML model you submitted**).

We intentionally made parts of the system description (provided as part of the UML modeling task) ambiguous. It is your responsibility to clarify any ambiguities with the assistants (your "customers"). If you have questions for the customer, send an e-mail to rse-tas@lists.inf.ethz.ch.

You should base your `Alloy` model on the (incomplete) `Alloy` file that we provide. In particular, implement all the provided functions. *Do not change the names of the signatures and do not change the signatures of the functions.*

Important: Use version 4.2 (platform independent) of Alloy. See http://alloytools.org/download.html.

1 Alloy Model

Task A. (25 points) Create a static `Alloy` model of all the data structures representing entities of the restaurant POS system. Include all the relevant details, relations and facts. In this diagram, we included two additional relation: *nextOrder* and *previousOrder*; and one additional class: *Time*. Ask your TA in order to clarify any ambiguities. In addition, document any detail that cannot be encoded in the `Alloy` model.

Make sure to implement all of the provided functions correctly and do not change their names or signatures. These functions will be used for some automated tests and will significantly affect the number of points obtained for this task.

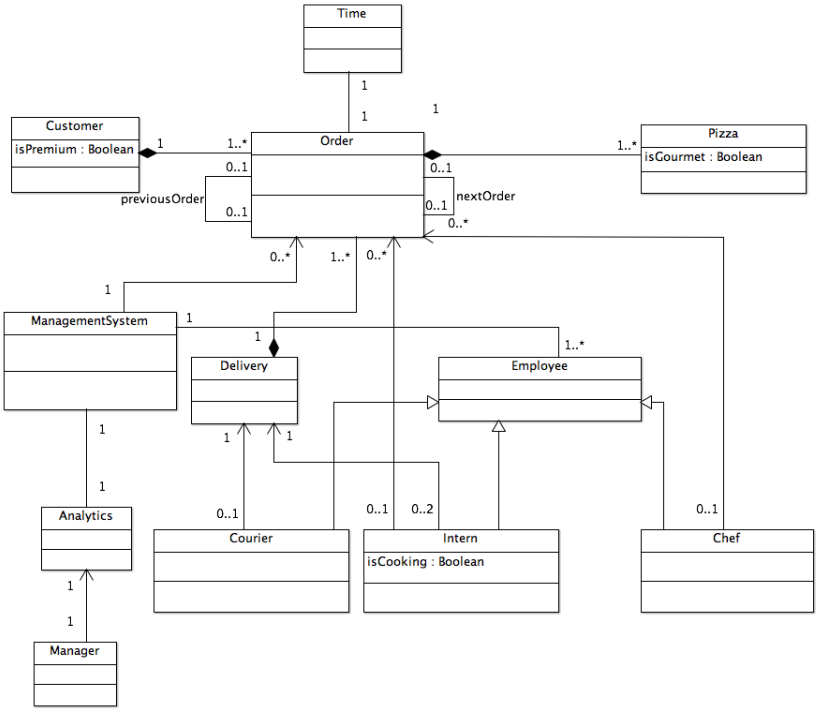

Figure 1: UML model

Task B. (15 points) Generate the following instances using `Alloy`, showing that your model is not overly restricted, or explain why they are infeasible if you cannot generate them:

1. There are four chefs, three couriers, one manager, two interns, and one restaurant.
2. An employee is either a chef, courier, or an intern.
3. Some customers are premium customers. Premium customers can up to two undelivered orders at a time while ordinary customers can only have one undelivered order at a time.
4. Some pizzas are gourmet pizzas. Only premium customers can order gourmet pizzas.

5. An order contains between one and three normal pizzas or one premium pizza. Chefs can only handle three orders at the same time. An intern can handle a maximum of two orders at the same time.

6. A delivery can contain up to three orders. Orders are delivered either by

   - one courier,
   - two interns, or
   - one intern and one courier.

7. An order can only be delivered if all the ordered pizzas have been cooked.

8. An order can be handled if all the preceding orders have been handled and there are available chef(s) or intern(s).

2 Deliverables

Submit your solution by email to rse-tas@lists.inf.ethz.ch. Each team should submit a single .zip file named **[team number]**_alloy.zip that includes:

1. An `Alloy` file with all the required code. The file must include short comments that explain your formalization.

2. A PDF file that includes:

   - Image exports of all the generated instances from Task B, with short descriptions.
   - A list of instances from Task B that you were not able to generate, together with a short explanation why the instance is not feasible.
   - A list of requirements from the project description that cannot be expressed in the `Alloy` model.

3 Resources

- `Alloy`: http://alloy.mit.edu/alloy/
- PDF creation: You can use Microsoft Word or LaTeX to create your document and **export it to PDF**.