

# Rigorous Software Engineering 2019

## Project 1 (Part 2: Alloy Modeling Task)

### Group 10:

Alexander Wälchli (16-915-233), Julian Büchel (16-943-938),  
Julian Roth (16-946-063), Flavio Schneider (17-923-327)

April 05, 2019

### Deliverables:

### Assumptions:

- We changed the Intern/Delivery and Courier/Delivery relation to be 0-1. This is so we can have couriers and interns without deliveries. Otherwise the number of employees would be bounded by the number of deliveries from below which doesn't make sense in our opinion.
- We added *isCooked* to Pizzas, that says if that Pizza has been cooked.
- Added *isCompleted* to Order if the order is completed and can be delivered. If the order is completed, then all pizzas in the order have been cooked. The opposite direction is also true.
- Added *canBeDelivered* to Deliveries, that says if Delivery can start to be delivered. If the delivery can be delivered, then all orders in the delivery are completed. The other direction is not necessarily true.
- Added *isDelivered* to Deliveries, that says that the delivery has been completed. If the delivery has been delivered, then *canBeDelivered* is true. The other direction is not necessarily true.
- An Order is only being cooked if an Intern or a Chef has been assigned to it. If an order has no Chef or Intern, then it is not being cooked. After an order has been completed, then it will keep its intern/ chef (we can do this because the relation between chef/intern and order is 0..\*). That makes sense because we might want to save the employee that cooked an order for complaints/ legal reasons.
- A Delivery is only being delivered if it is assigned to a Courier/ Intern. This is only possible if the delivery can be delivered. If the delivery has no intern/ courier, then it is not being delivered.

- An order can only be handled by either an intern or a chef, not by both at the same time.
- Multiple orders can share the same Payment. This is also possible across multiple Customers. E.g. Order 0 by Customer 0 and Order 1 by Customer 1 use the same payment. We wanted to ask about this during Office Hours on Friday, April 5th, but no TAs were in the room at 2pm. Therefore we couldn't clarify if that assumption is reasonable.
- Multiple orders can share the same Address. This is also possible across multiple Customers. E.g. Order 0 by Customer 0 and Order 1 by Customer 1 use the same address. We wanted to ask about this during Office Hours on Friday, April 5th, but no TAs were in the room at 2pm. Therefore we couldn't clarify if that assumption is reasonable.
- All times are positive and unique.  $\Rightarrow$  There is no *startTime* or *endTime* that is equal to another *startTime* or *endTime*.
- The field *isCooking* of Intern says that the intern acts as a chef. If it is true, then that intern has 0 deliveries. If it is false, the intern has 0 orders.
- Only one restaurant can be modeled at the same time  $\Rightarrow$  Only a single Manager exists.
- We have also decided that the restrictions 2. - 8. from Task B make a lot of sense and therefore implemented them as facts. Therefore any instance that we create in Task B is also a instance for 2. - 8.

#### **Naming convention:**

A relation between *source* and *target* is named *sourceTarget*.

#### **Requirements from the project description that cannot be expressed in the Alloy model:**

All the requirements from the project description can be generated in the Alloy model

#### **Generated instances:**

Our model can generate instances that enforce all of the restrictions from Task B simultaneously. We give two images: One that generates an instance of the specified size from Task B.1 and one instance of larger size. We have also included both PNGs in the zip file for easier viewing and higher resolution.

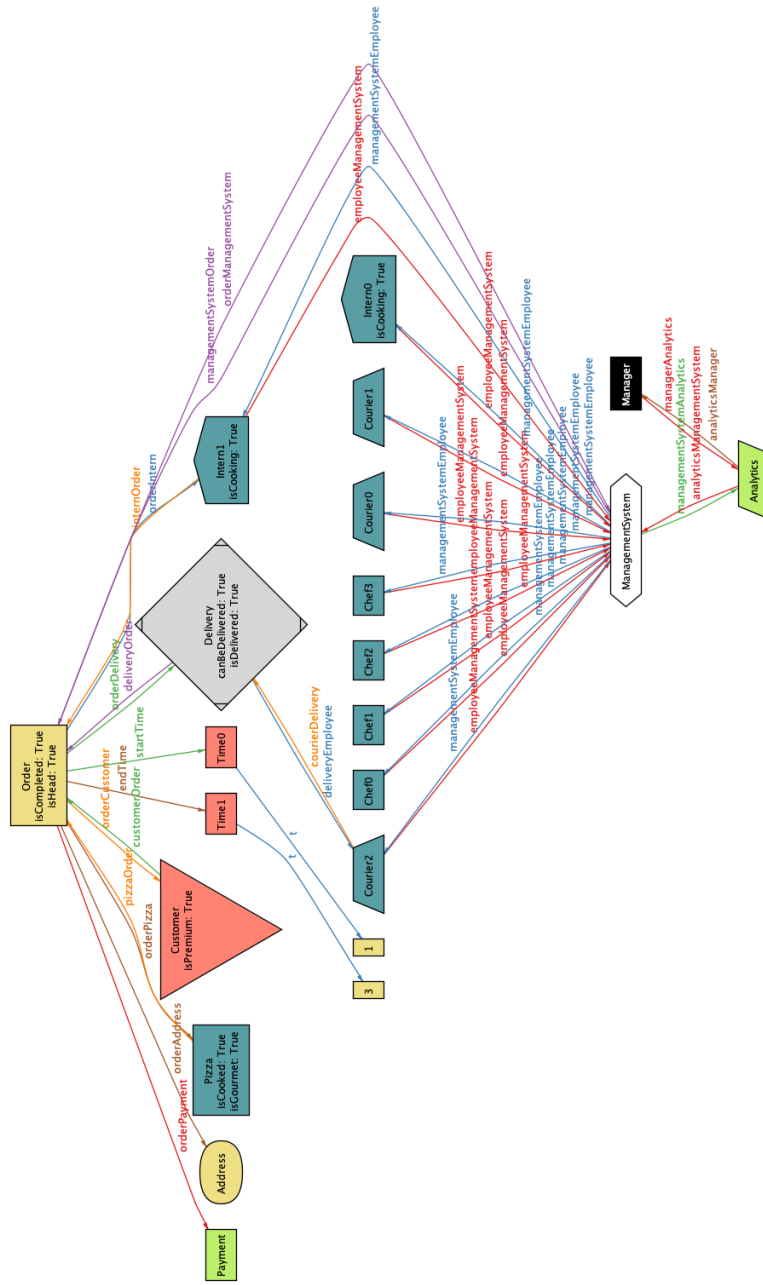


Figure 1: Alloy instance for Task B.1

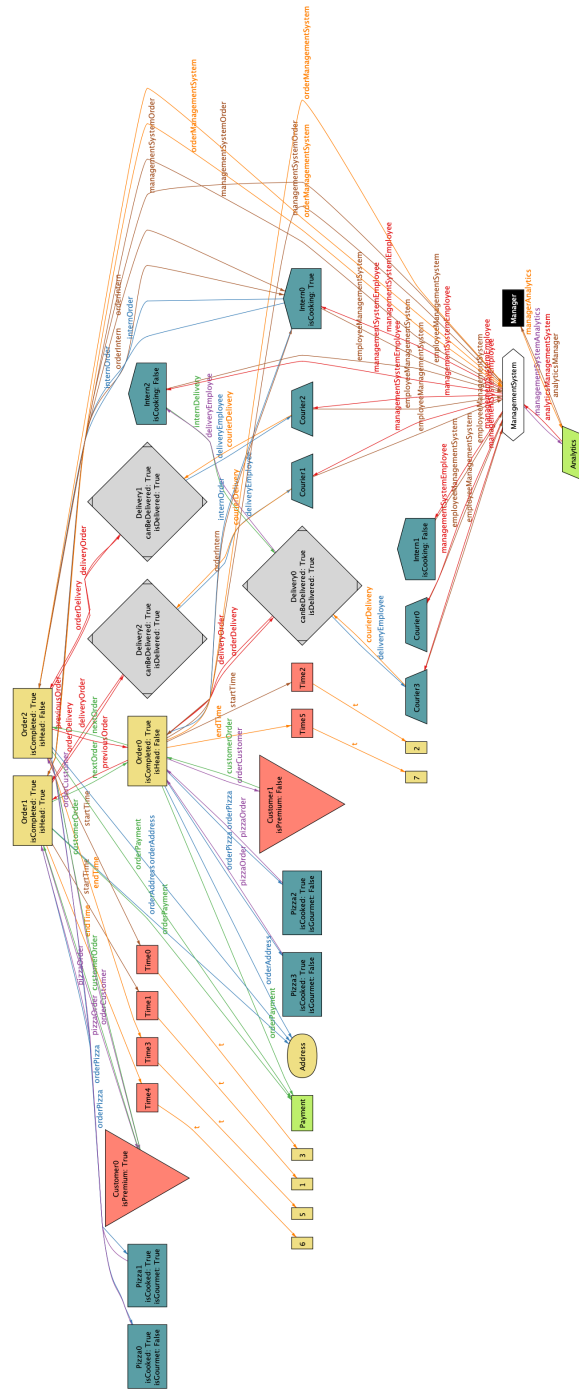


Figure 2: Alloy instance of greater size