Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Semesterproject

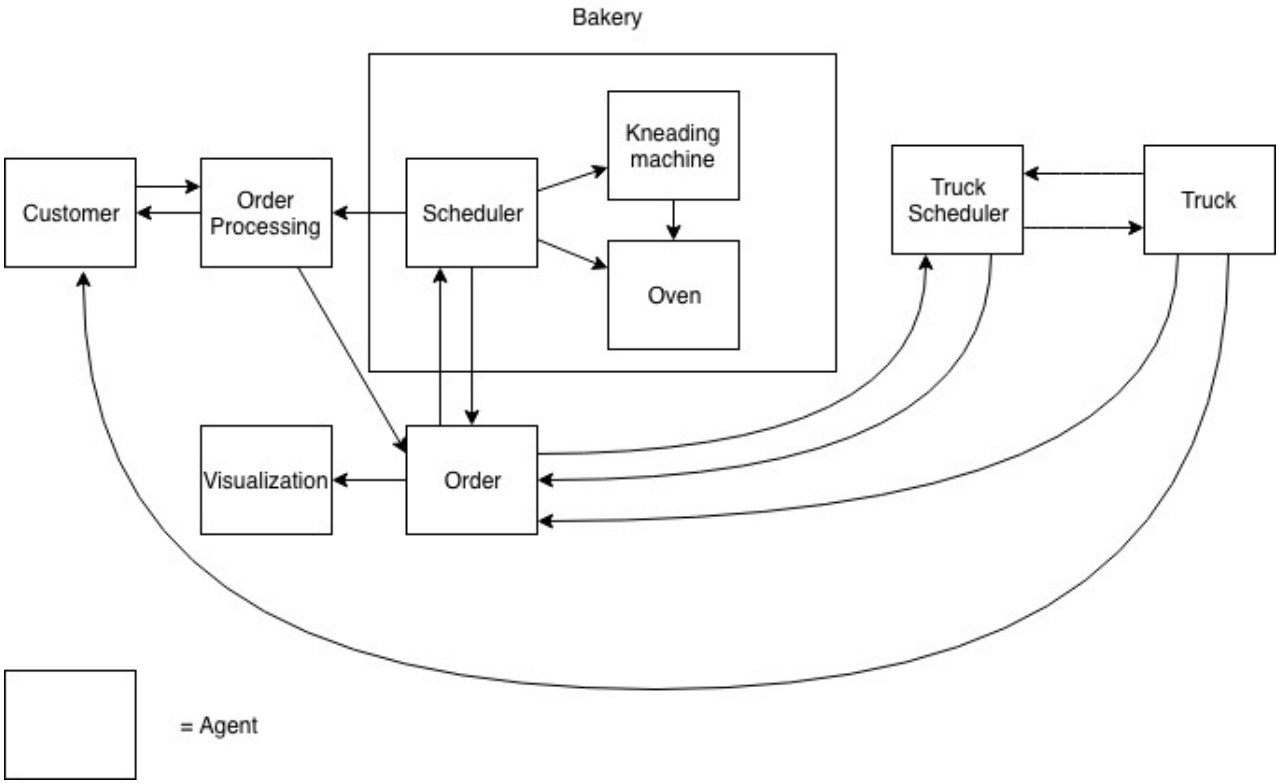
Multi Agent and Agent Systems

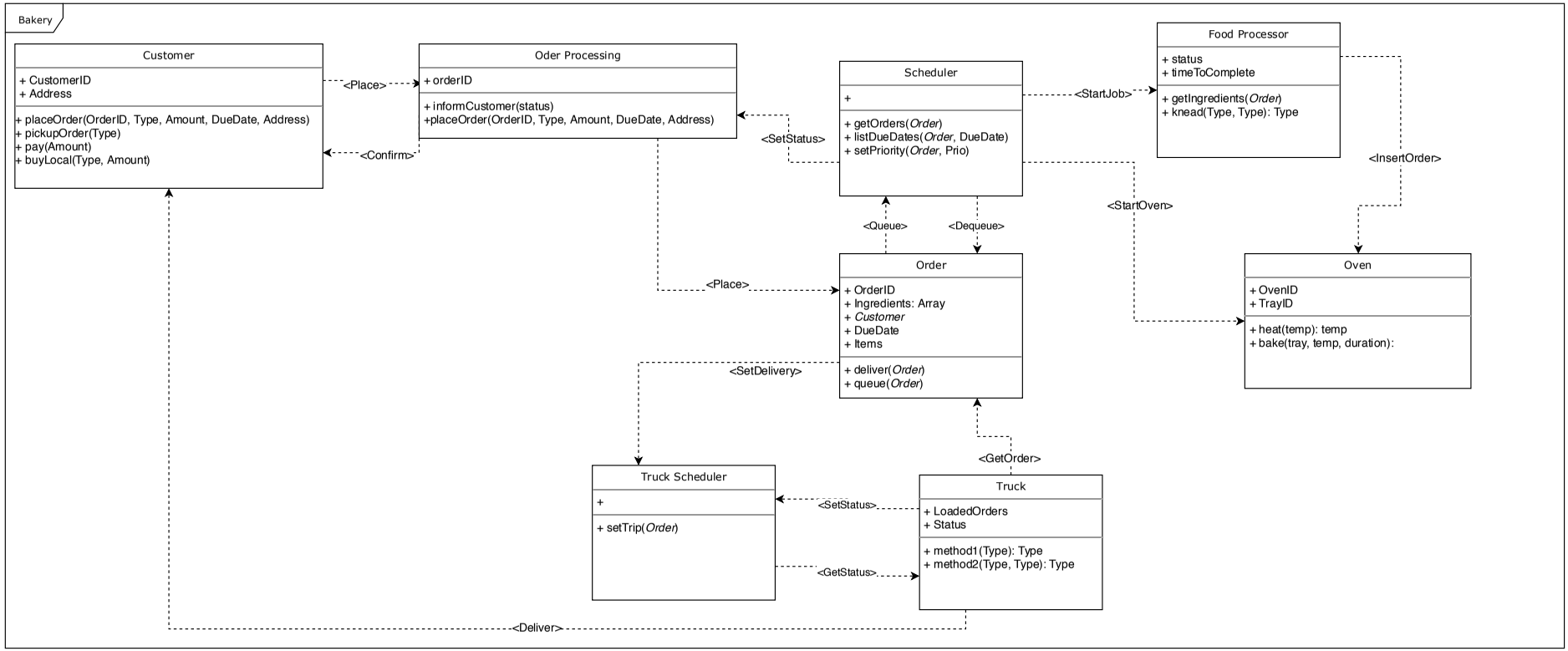
Department of Autonomous Systems
Referent: Prof. Dr. Gerhard Kraetzschmar

Submitted by: Team PJT

Sankt Augustin 4. November 2018

Architecture




Bakery
Customer
+ CustomerID
+ Address
+ placeOrder(OrderID, Type, Amount, DueDate, Address)
+ pickupOrder(Type)
+ pay(Amount)
+ buyLocal(Type, Amount)
<Place>
<Confirm>
Oder Processing
+ orderID
+ informCustomer(status)
+placeOrder(OrderID, Type, Amount, DueDate, Address)
<SetStatus>
Scheduler
+
+ getOrders(Order)
+ listDueDates(Order, DueDate)
+ setPriority(Order, Prio)
<StartJob>
Food Processor
+ status
+ timeToComplete
+ getIngredients(Order)
+ knead(Type, Type): Type
<InsertOrder>
<StartOven>
<Queue>
<Dequeue>
<Place>
Order
+ OrderID
+ Ingredients: Array
+ Customer
+ DueDate
+ Items
+ deliver(Order)
+ queue(Order)
Oven
+ OvenID
+ TrayID
+ heat(temp): temp
+ bake(tray, temp, duration):
<SetDelivery>
<GetOrder>
Truck Scheduler
+
+ setTrip(Order)
<SetStatus>
<GetStatus>
Truck
+ LoadedOrders
+ Status
+ method1(Type): Type
+ method2(Type, Type): Type
<Deliver>

Aggregation of order data

Aggregation of order data can be done in the following manner:

- **An aggregation of a customer's orders for each day or each date <ddd.hh>**
  → Use of a Hashmap. Key is date value is order. The advantage is that a hashmap has got an index. That means that worst case runtime for searching for an order within hashmap is $O(n) = 1$

```
Hashmap<Date, Order> hmMapDaily = new Hashmap<Date, Order>();
hmMapDaily.put(new Date(), new Order());
Order co = hmMapDaily.get(date);
```

- **An aggregation of all orders for a particular product for each day or each date**
  → Hashmap of Hashmaps. One entry within Hashmap represents one product. Key is product value is a hashmap. One Hashmap within Hashmap has as key a date, as value an array of orders.

```
Hashmap<ProductId, Hashmap<Date, Orders[]>> hMapProduct;
hMapProduct.put(new ProductId(), Hashmap<Date, Orders[]>);
Hashmap<Date, Orders[]> hmDate = hMapProduct.get(ProductId);
```

  So hMapProduct would look the following way:

$$hMapProduct = \begin{pmatrix} \{ProductId, Hashmap < Date, Orders[] >\} \\ . \\ . \\ . \\ \{ProductId, Hashmap < Date, Orders[] >\} \end{pmatrix}$$