Gerhard Kraetzschmar
Argentina Ortega
Alexander Moriarty

MAAS Projects

Flying Saucers Bakery

- We model a typical German bakery.
- The bakery offers a number of different products.
- The bakery has numerous customers.
- The bakery delivers products via a number of trucks.
- We look at order processing, the production process, and product delivery

Flying Saucers Bakery: Order Processing

- Some customers (mainly sales shops, supermarkets) order between 50% and 100% of their demand at end of the day before delivery (next day orders). The remaining demand is ordered on the same day as delivery.
- Some customers (elderly homes, hospitals, canteens) order all their demand once a day for next day delivery.
- Some customers (catering services, clubs, associations, etc. organising special events) order their demand several days ahead of delivery.
- To guarantee freshness, products can only be produced on the day of delivery.

Flying Saucers Bakery: Formalizing Order Processing

- We work on a common system clock <ddd.hh> consisting of a day number and an hour. E.g. delivery at <12.04> means that an order must be delivered on day 12 no later than the end of hour 04.
- An order consists of the following information:
  - A customer id
  - An order date (day and time at which the order is released)
  - A delivery date (day and time at which the order must latest be delivered)
  - A vector over the range of products with the # of items ordered
- Example: [“WhizKidSchool”, <11.16>, <12.09>, [100, 120, 0, 0, …]]

Flying Saucers Bakery: Production Process

- The production of a product of the bakery follows this process:
  - First, the dough must be prepared in a kneading machine. This takes a certain dough preparation time, which is independent of the number of product items.
  - Next, the dough must rest. This takes a certain resting time.
  - Next, the dough must be prepared for baking. This takes a certain item preparation time for EACH item.
  - Next, the product must be baked. This takes a certain baking time.
  - After baking, the product must cool before it can be boxed and delivered. The minimal cooling time is proportional to (baking temperature minus 40° Celsius).

Flying Saucers Bakery: Production Process

- Further details and constraints of the production process:
  - The bakery has a number of ovens, each of which has 4 equally-sized baking slots with individual controllable baking temperature.
  - Changing the baking temperature requires time for adjustment, proportional to the temperature difference.
  - For each product, a number of items simultaneously fit into an oven slot.
  - Slots are always filled only with items of a single product. In/out same time.
  - There is a limited number of dough preparation machines and item preparation work places.
  - For now, we assume no constraints on storage areas for
    - resting dough and
    - items cooling after baking.
  - A production shift lasts from midnight to lunch <ddd.00> – <ddd.12>

Flying Saucers Bakery: Formalizing Product Information

- Product Information:
  - product id
  - dough prep time
  - resting period
  - item prep time
  - # of items fitting in oven slot
  - baking time
  - baking temperature
  - boxing temperature
  - cooling time factor (in sec/°C)
  - # of items fitting into a box
  - production cost
  - sales price

Flying Saucers Bakery: Formalizing the Production Process

- Factory information:
  - number of dough kneading machines
  - number of ovens
  - oven heating time factor (sec/°C)
  - oven cooling time factor (sec/°C)

Flying Saucers Bakery: Product Delivery

- The bakery delivers the products on or before the due date to its customers, using a number of trucks.
- For transport and delivery, products are boxed.
  For each product, a certain number of items fit into a box.
  Only items of same kind in a box.
- Trucks can transport a maximum number of boxes per tour.
- Trucks, customers, and the bakery have a location <x,y>.
- A street network is defined by a set of labeled direct connections between locations of customers, and the bakery. The labels correspond to distance.

Flying Saucers Bakery: Formalizing Product Delivery

- Packaging information: associating the order vector [100, 120, 0, 0, …] with a box vector [2, 3, 0, 0, …]
- Information on trucks:
  - truck ID
  - load capacity (maximum number of boxes)
  - load: a list of modified orders (item vector replaced by box vector)
  - location
- Street network: as a labeled graph of vertices V and edges E where
  - the set of vertices V contains all customers and the bakery
  - the set of edges E is a subset of V x V x $\mathbb{R}^+$
  - Note: for vertices a, b, two edges (a, b, x) and (b, a, y) with x =/= y may be defined

Project Work: First Steps

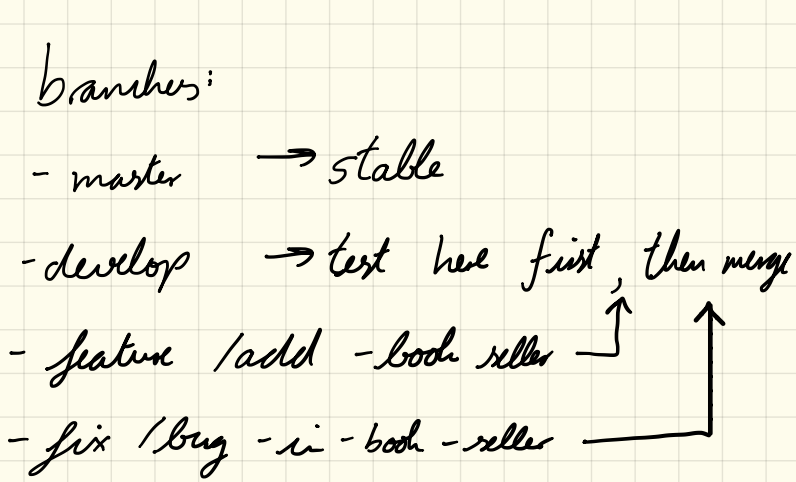
- Get acquainted with the implementation platform:
  - Download and install software
  - Check correct installation
  - Consult documentation
  - Work through tutorial
- Create some toy example:
  - Create three to four agents
  - Let them exchange (send and respond to) some messages

Project Work: Second Steps

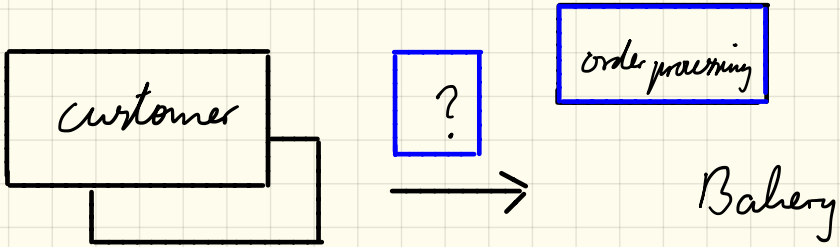
- In our scenario, focus on the order processing first:
  - You want to create agents representing customers:
    - These agents should be able to read a schedule of orders from a file.
    - They need to send these orders to the bakery agent at the right time.
  - You probably also want to have an agent representing the bakery:
    - This agent should take the orders for customers and store them in some appropriate way.
    - Agents to be implemented later (for production and delivery) may want to know:
      - An aggregation of a customer's orders for each day or each date <ddd.hh>
      - An aggregation of all orders for a particular product for each day or each date
      - Think of other kinds of aggregations and how to represent them.
  - Discuss whether you need additional agents. If yes, what for.
- Implement and test your multiagent system.

branches:

- master → stable
- develop → test here first, then merge
- feature/add-book-seller
- fix/bug-in-book-seller



MAAS PROJECTS:



- order is object for storing info
- customer
- order processing
- delivery
- system clock
- scheduler
- Visualization agent

// How to process order?

// Who is an agent, what is an agent?

// orders processing interface

// An object isn't able to do smth. by itself

// orders have a lifecycle

// common

Stages:

- order processing
- Bakery
  - dough prep
  - baking and cooling
  - packaging and loading
- delivery

Order as agent (heating etc. - Auftrag als Agent modellieren - wann kommt wie an wann bei wem fertig

Scheduler als zentrale Instanz

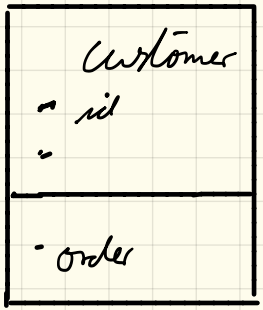
Offer als Agent nehmen - 1 Customer Agent

Ein globaler Agent für System Clock

Kundenmaschine als Agent

Lieferwagen als Agent

7.13 10.13 2Mi für Fertigung


Customer
- id
-
- order