**Concepts and Models of Knowledge Engineering, 2016W**

**Group 12 – DoBot Burger**

Berghofer Maximilian Paul, a1147722

David Götzinger, a1148353

Christoph Kern, a1148588

# Use Case Description

The goal of this project is to create a method which makes it possible to control a DoBot located in the OMILab with a modelling method created with ADOxx 1.5. The use case is to model a receipt for a burger in ADOxx and after the receipt is finished the DoBot should be controlled to stack the burger in the right order.

# Problem Statement

There are three parts that need to be addressed to finish the project. First a REST client has to be implemented which controls the DoBot. The second part is to create an ontology that has the coordinates for the different ingredients which can be used to create the burger. Since there was no existing ontology that contained the coordinates for the burger ingredients we determined our own. The last part is to create a modelling method in ADOxx which was extended with scripts.

# Tile Summary

The project aims to control a DoBot to build a burger. This is done by using ADOxx to create a recipe which includes certain constraints in the form of ADO scripts. A Java REST client then reads position data from an ontology and controls the DoBot.

# Experiment Description

Like already mentioned the project consists of an ontology, a Java REST client and a modelling method created in ADOxx 1.5.

# KE Concept Description

*Ontology:* We used an OWL ontology created in Protégé. The ontology saves the coordinates of the different ingredients. There are two kinds of ingredients, the main ingredients which are used for the burger and side dishes.

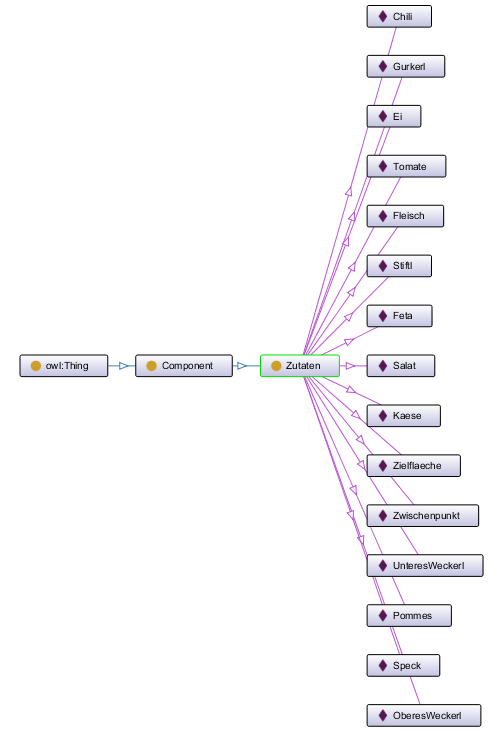


Figure 1: Picture of used Ontology

*The modelling method:* The modelling method is created with ADOxx 1.5. It consists of two model types. A model type for creating the receipt, which has all the ingredients and two kinds of relations (one for the side dish and one for the main ingredients). The second model only has the ingredients and is for saving quantity of the different ingredients (which is saved in an attribute).

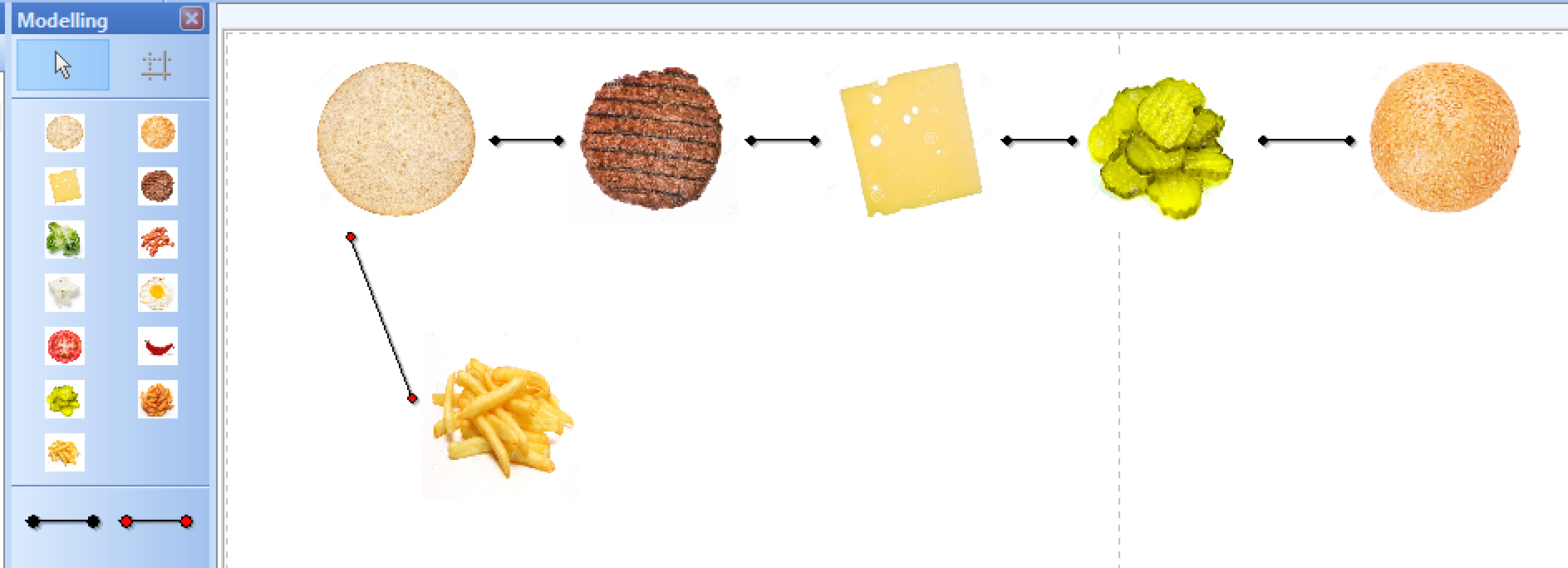


Figure 2: Model type for receipt



Figure 3: Model type for ingredients

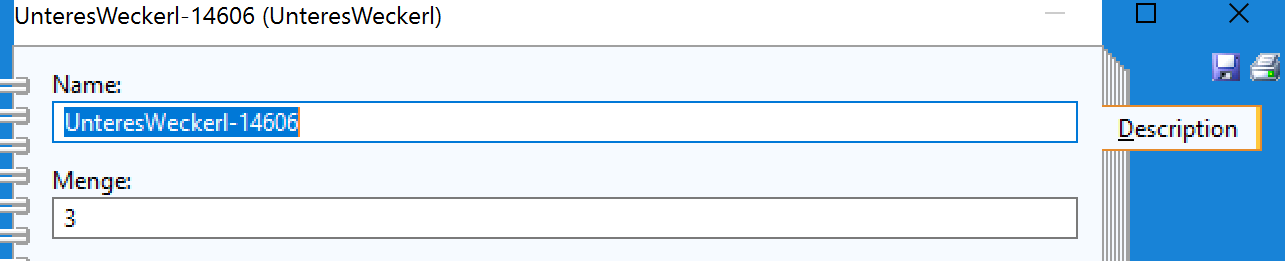


Figure 4: Quantity of ingredients

# Validation Environment Description

*Scripts:* The modelling method is extended with two scripts. The first script is to check if the quantity of the ingredients is enough to make a burger. The second script is to trigger the Java REST client and needs to be located in C:\.

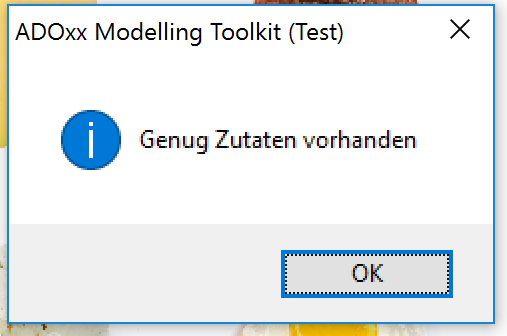


Figure 5: Successful quantity check

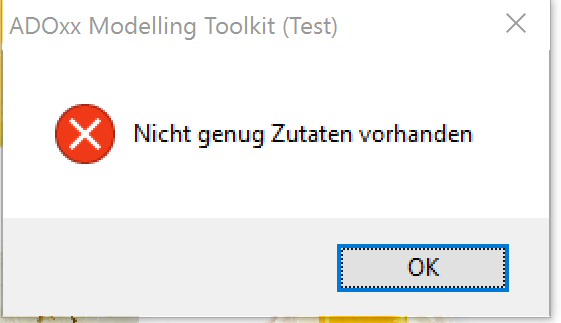


Figure 6: Unsuccessful quantity check

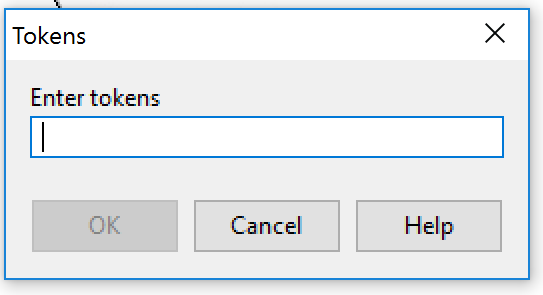


Figure 7: Enter token for controlling DoBot

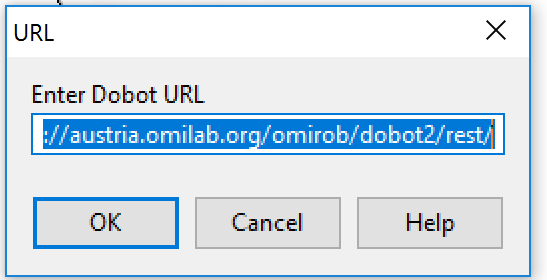


Figure 8: Enter URL for DoBot

*REST Client:* The client is implemented in Java and is used to control the DoBot. It parses the built ontology to read ingredient position data and controls the DoBot by calling the REST API.