1. Modelling robotics

Module specification	Explanation		
Teacher Name			
Training Topic	Robotics application in Virtual Laboratory		
Training Code	UNIOULU_01		
Module Name	Modelling robotics		
Module duration	100 minutes		
Module objective	• Install modelling tools • Understand basics of modelling • Model of simple real-life case		
Mode of provision	Classroom		
Laboratory structure	Time (min)	Objective	Performed by?
	10	Introduction	Teacher
	15	Mobelling languages implemented in beeup	Teacher
	10	Installing beeup	Students
	20	Using beeup	Teacher and Students
	15	Simple models	Students
	25	Warehouse example	Students
	5	Finishing	Teacher

1 Introduction

Objectives

- Understanding what modelling is
- Install Bee-Up modelling tool
- Basic understanding of modelling languages included in Bee-Up
- Create some simple models
- Warehouse example

Models are abstraction of object being modelled. Aim of modelling is to simplify problem at hand and conceptualize system. Models can be used as tools for planning. In this context models are diagrams based on modelling language. These models can be broadly divided to static models describing structures and relationships and dynamic models describing actions and states.

Creating a model helps in designing complex systems. Models can start with high level of abstraction and have hierarchical structure each layer defining smaller parts until model is specific enough for implementation. For example one action can be getting item from warehouse, that is divided to moving to and from the item and picking item up, which are further divided to route planning and obstacle avoidance and moving arm and grabbing item.

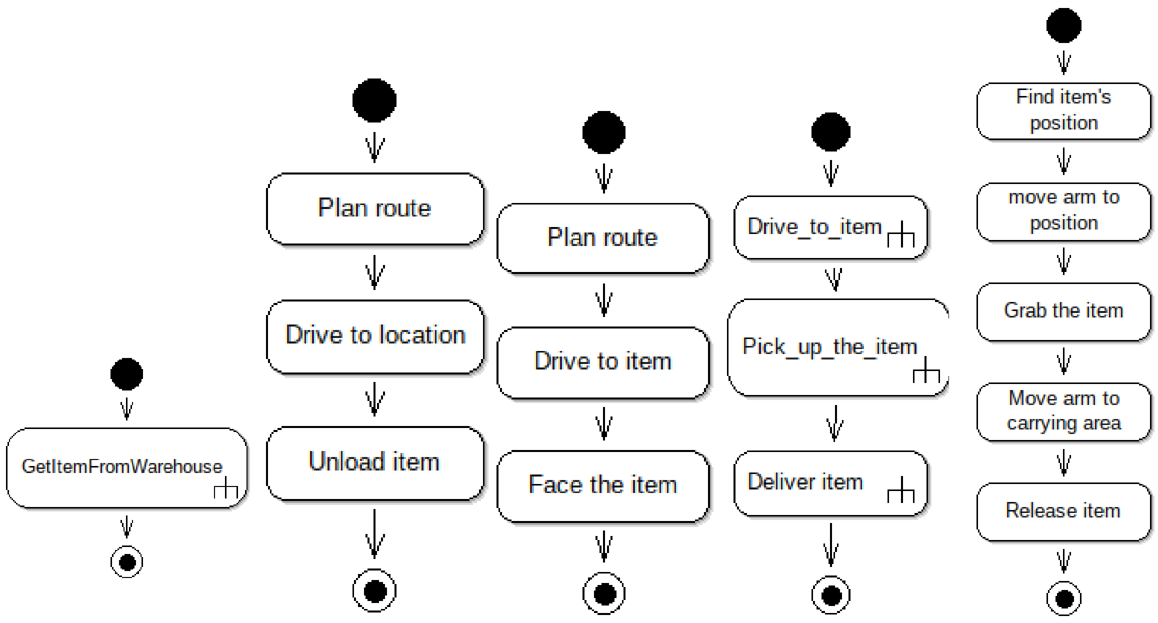


Illustration 1: Example of task refiment

Why modelling is important

Model value

Abstraction of system reduces complexity of presentation. Visual cues provide information in more intuitive way than plain text.

Formalized semantics allow models to be machine analyzed with simulations and validation. Formalized semantics also allow code generation from models nad automatic transformations between models.

Models can be used for documentation or design

Bit about modelling tools

Drawing models can be done with simple tools like ms-paint or gimp but they don’t have any support for modelling and editing models is difficult. Some drawing tools like drawio are designed for creating diagrams, but they don’t enforce modelling language syntax so it is earier to make mistakes in model. Drawing tools don’t also include tools for running simulations on models or analysing them. Modelling tool should have support for modelling language and enforce ih no models created adhere to standards

2 Beeup

Bee-Up is implemented using Adoxx metamodelling platform. It provides support for different modelling languages and some ADOxx platforms processing functionality. Bee-Up also extends the processing capabilities with additional mechanism and algorithms like:

Uniform simulation of process models

Analysis of Petri Nets through manual or automatic execution.

Generation of SQL-create statements from ER-models

Adoxx query language

Exporting models in different formats (XML, RDF, ADL) and generating graphics of model(JPG PNG)

Support for AdoScript (https://www.adoxx.org/AdoScriptDoc/index.html). AdoScript can be used to add external commands to model for quick prototyping

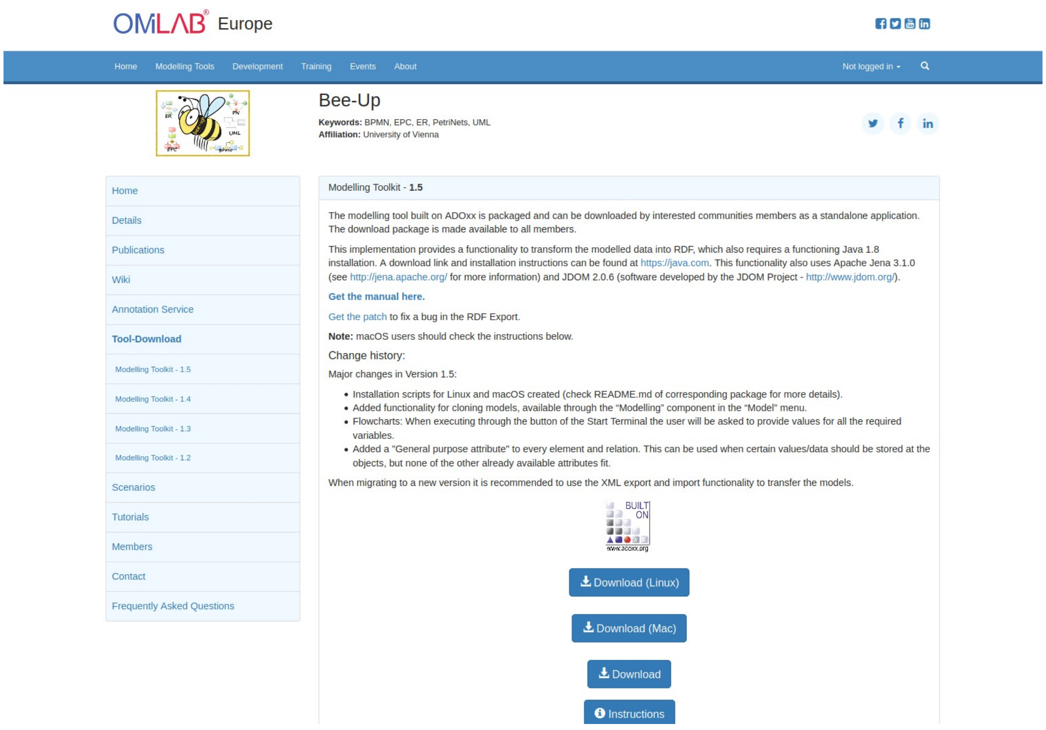
Demo video of omirob http://vienna.omilab.org/repo/files/bee-up/Media/BeeUp_omiarm.mp4 z

Beeup supports several different modelling languages that can together be used to represent complex systems

2.1 Installation

Beeup is developed for windows but can be run on linux or mac using wine and docker. Installation script works on Ubuntu and Fedora, but can be modified to work on other distributions.

Download BeeUp from https://austria.omilab.org/psm/content/bee-up/download?view=download



Bee-Up

Keywords: BPMN, EPC, ER, PetriNets, UML
Affiliation: University of Vienna

Modelling Toolkit - **1.5**

The modelling tool built on ADOxx is packaged and can be downloaded by interested communities members as a standalone application. The download package is made available to all members.

This implementation provides a functionality to transform the modelled data into RDF, which also requires a functioning Java 1.8 installation. A download link and installation instructions can be found at https://java.com. This functionality also uses Apache Jena 3.1.0 (see http://jena.apache.org/ for more information) and JDOM 2.0.6 (software developed by the JDOM Project - http://www.jdom.org/).

Get the manual here.

Get the patch to fix a bug in the RDF Export.

Note: macOS users should check the instructions below.

Change history:

Major changes in Version 1.5:

- Installation scripts for Linux and macOS created (check README.md of corresponding package for more details).
- Added functionality for cloning models, available through the "Modelling" component in the "Model" menu.
- Flowcharts: When executing through the button of the Start Terminal the user will be asked to provide values for all the required variables.
- Added a "General purpose attribute" to every element and relation. This can be used when certain values/data should be stored at the objects, but none of the other already available attributes fit.

When migrating to a new version it is recommended to use the XML export and import functionality to transfer the models.

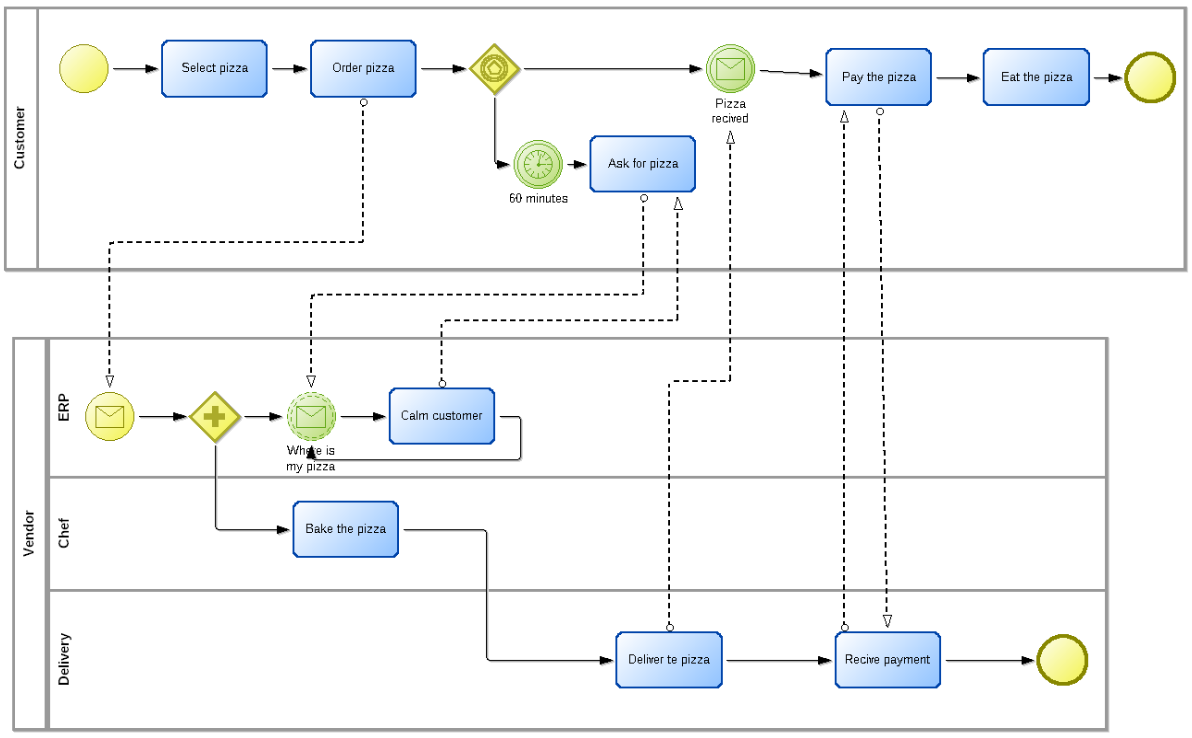
Extract the package
Package holds all the necessary files for installing beeup as well as manual.
Run setup.exe or install_linux.sh or install_mac.sh
On Mac and Linux it is necessary to add execution privileges to the installation script.
Installation checks prerequisites and informs what needs to be installed. Installer then tries to solve the dependencies and creates SQL Server instance if necessary. By Default Bee-Up creates and uses the database with name 'beeup15' and user Admin with password: Password
If Installation was successful there should be shortcut on desktop for starting Bee-up.

3 BPMN Business process model and notation.

BPMN is used to visualize business process in a flow-chart format. Describes business process at human level rather than the software engine level.

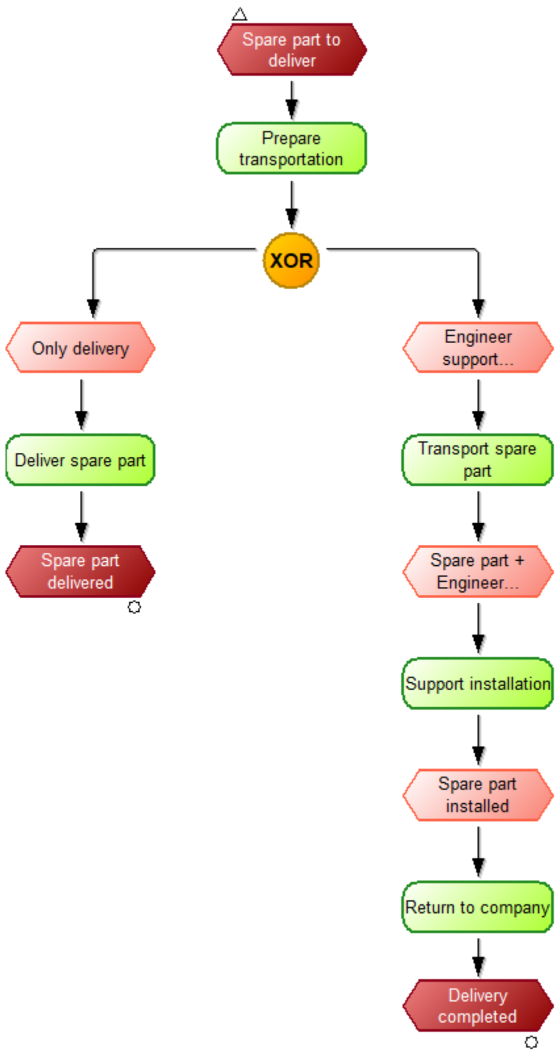
1 OMG (object management group) standard designed to support the business process management paradigm.
2 Extensive range of diagrammatic possibilities

BPMN diagrams consist from four basic element categories: Flow objects, Connecting objects, Swim lanes and artefacts. Flow objects can be events, activities or gateways that determine forking and merging of paths. Connections represent sequence and message flow or association. Swim lanes are used to separate different organisations and to categorise activities according to function or role



4 EPC and eEPC: Event driven process chain

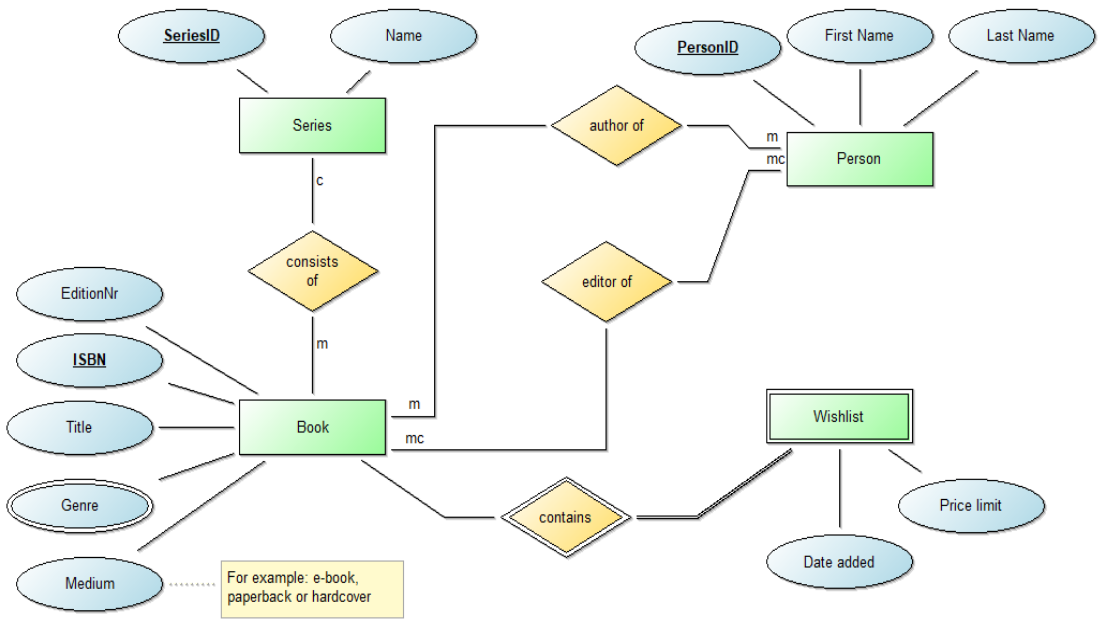
Main elements in event driven process chains are functions and event that alternate, each event is followed by a function and vice versa. Functions can have other object connected to them giving more information about process. Control flow can also have logical connectors controlling flow. There are three logical relationships in EPC: XOR(branch/merge) AND(fork/join) and OR(OR)



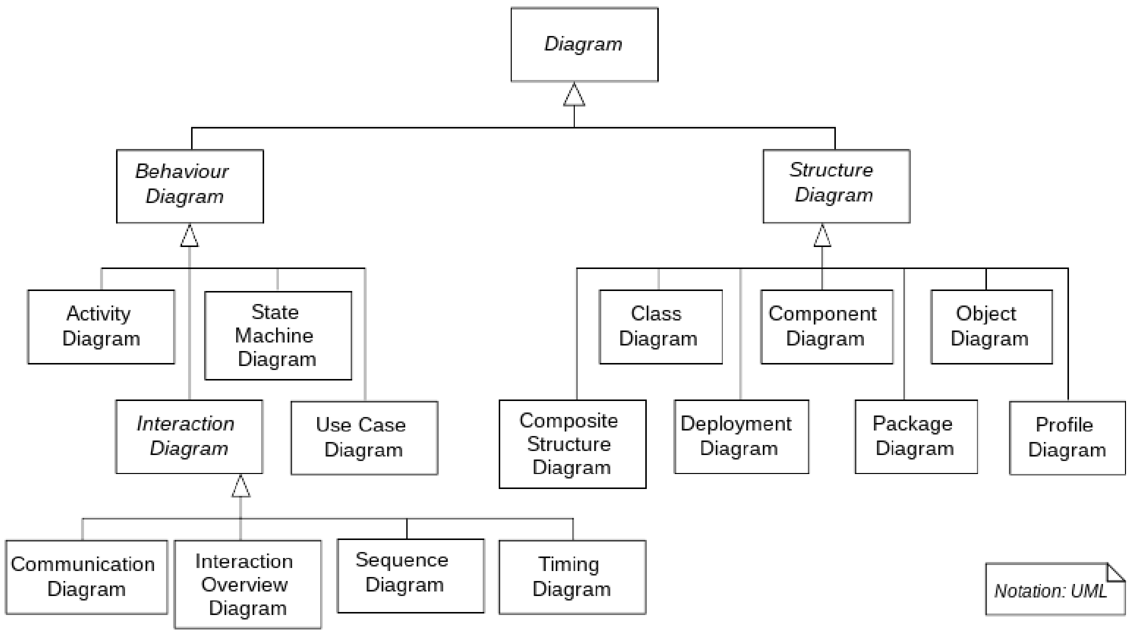
5 ER: Entity relationship

For data modelling.
Entities, attributes and relationships

Describes data schema in graphical form. Consists of entities connected by relationships. Entities can have properties which include identifiers called primary keys. Entity relationship models are traditionally build with two or three levels of abstraction highest of which is conceptual followed by logical and physical. Conceptual model establishes the overall scope of what is to be included within the model set. Logical ER model can be build on conceptual model, but it is not necessary if the scope of the model is small. Physical data model contains enough detail to produce database.



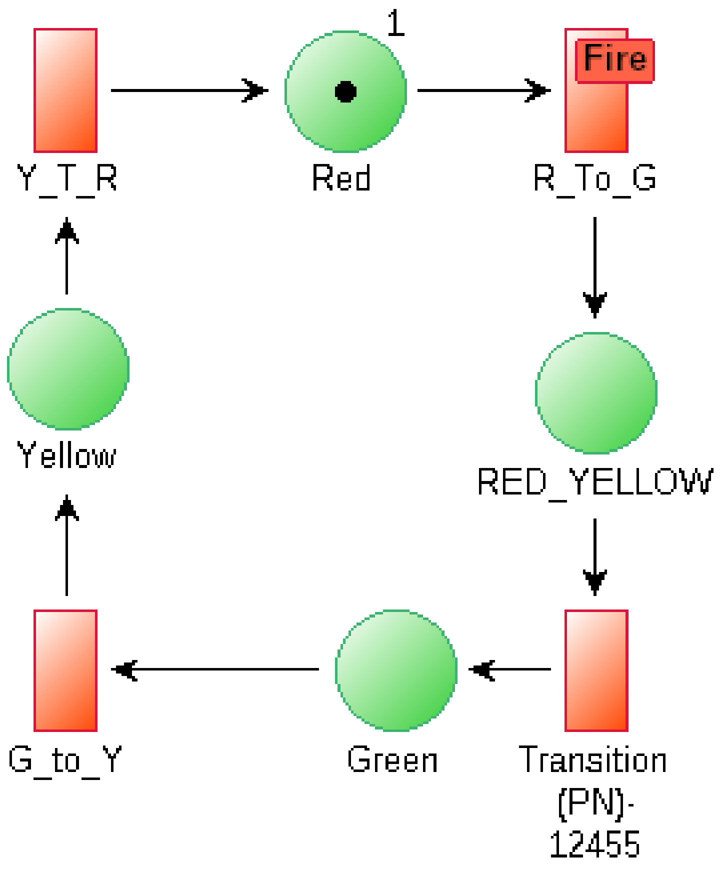
1. **Behavior Diagrams** include the Use Case Diagram (used by some methodologies during requirements gathering); Activity Diagram, and State Machine Diagram.
2. **Interaction Diagrams,** all derived from the more general Behaviour Diagram, include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram.



7 PN Petri Nets

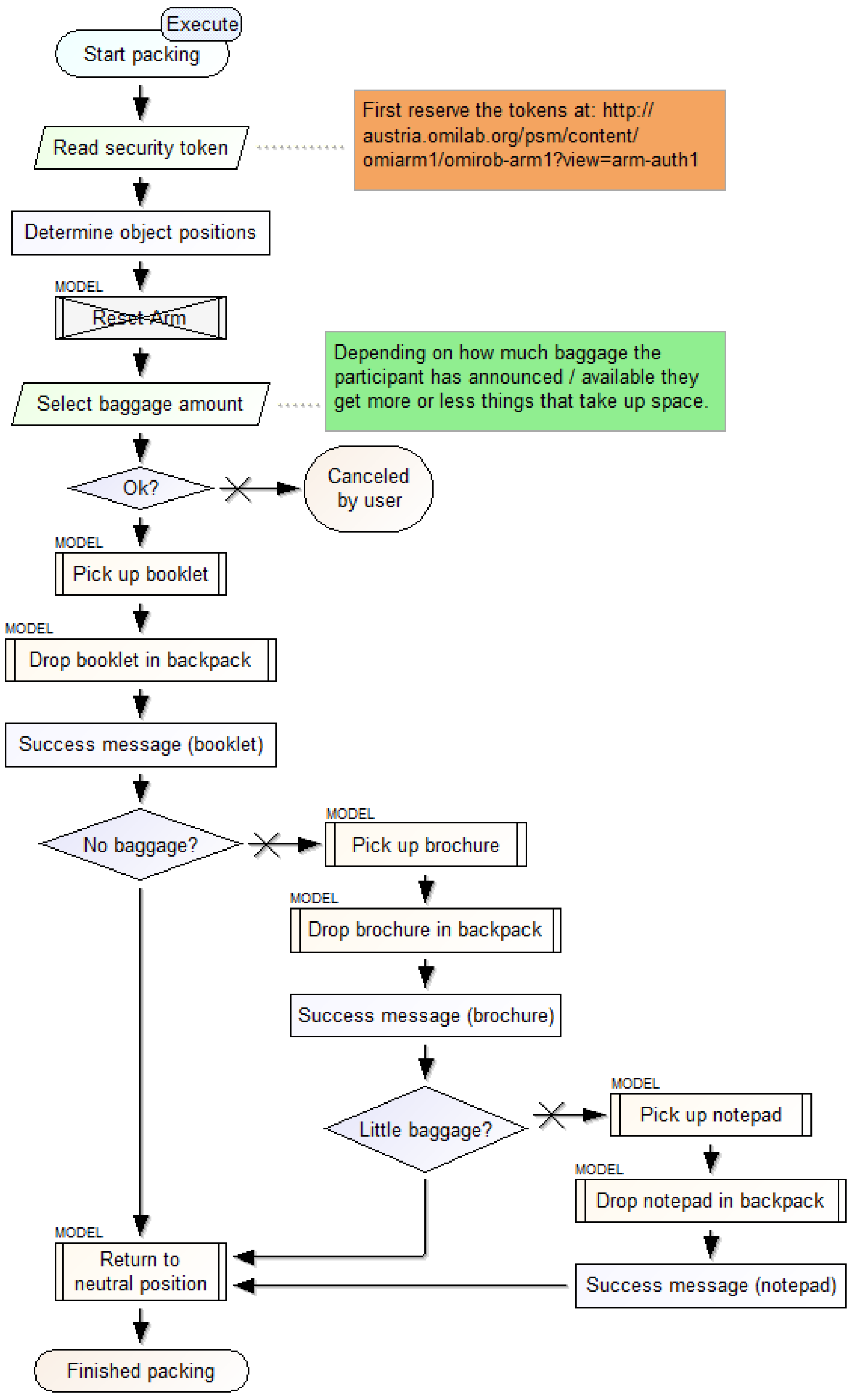
Long standing diagrammatic modelling method.
Minimal but powerful schematic.
Strong mathematical foundations
Places, transitions, arch, tokens.

Petri Nets are composed of places and transitions. Places can contain arbitrary amount of tokens that represent resources. Transitions connect two or more places via arch. Arch are directional. Transitions can be fired consuming tokens from places that haze connections to it and producing new tokens to places that connect from it. Petri nets have rigid mathematical foundations. Petri nets can be analysed mathematically without need for simulation.



8 Flowcharts

Simple flowchart without extra features. Five kinds of nodes start, operation, external operation, decicion, and end. Flowchart made with BeeUp can call adoxxscript scripting language and with adoxx script it is possible to call external programs and http-apis. This allows interaction with web-services real-world objects for testing.


Execute
Start packing
Read security token
First reserve the tokens at: http://austria.omilab.org/psm/content/omiarm1/omirob-arm1?view=arm-auth1
Determine object positions
MODEL
~~Reset Arm~~
Select baggage amount
Depending on how much baggage the participant has announced / available they get more or less things that take up space.
Ok?
Canceled by user
MODEL
Pick up booklet
MODEL
Drop booklet in backpack
Success message (booklet)
No baggage?
MODEL
Pick up brochure
MODEL
Drop brochure in backpack
Success message (brochure)
Little baggage?
MODEL
Pick up notepad
MODEL
Drop notepad in backpack
Success message (notepad)
MODEL
Return to neutral position
Finished packing

9 Using BeeUp

Create new model by choosing model → new

Choose model type from dialog in the centre (BPMN)

Give model a name and optionally version number. Choose model group to witch model should belong to. If necessary create new model group. Note that model groups can be nested.

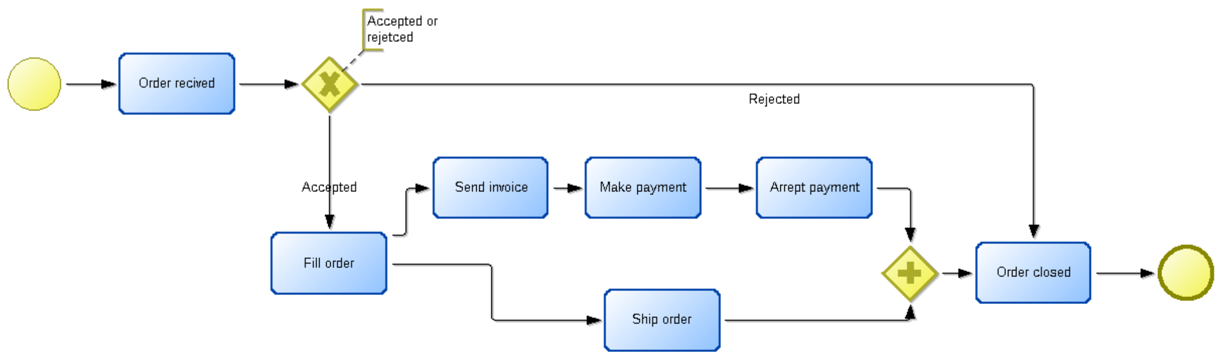
On left side of the screen there are model explorer to switching between models and modelling window showing available modelling objects.

To add objects choose it from modelling window and click on the modelling area.

To add relation choose relation from modelling window and click and drag it between objects.

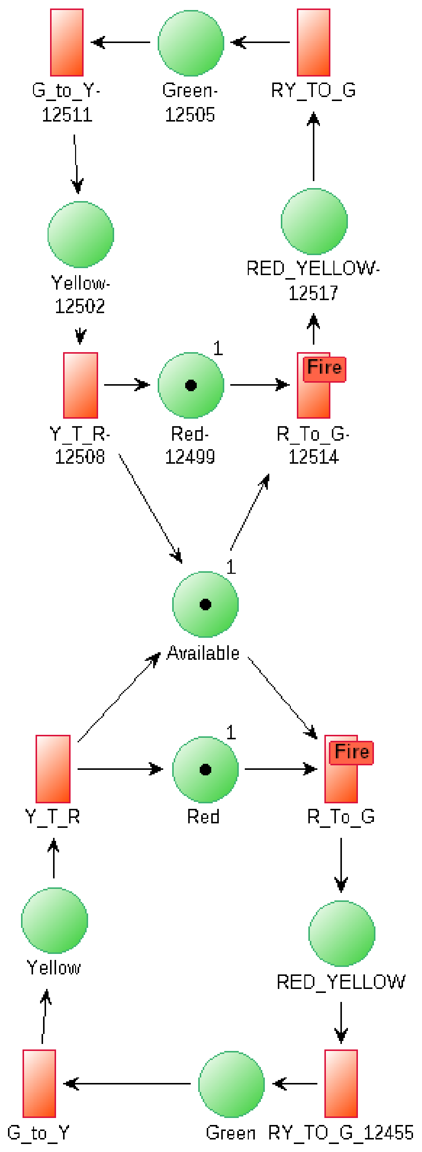
Double click on objects on modelling area to open notebook containing information for the object. Clicking on object name when object is chosen lets you to modify the name.

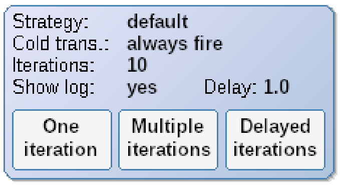
9.1 Guided example:



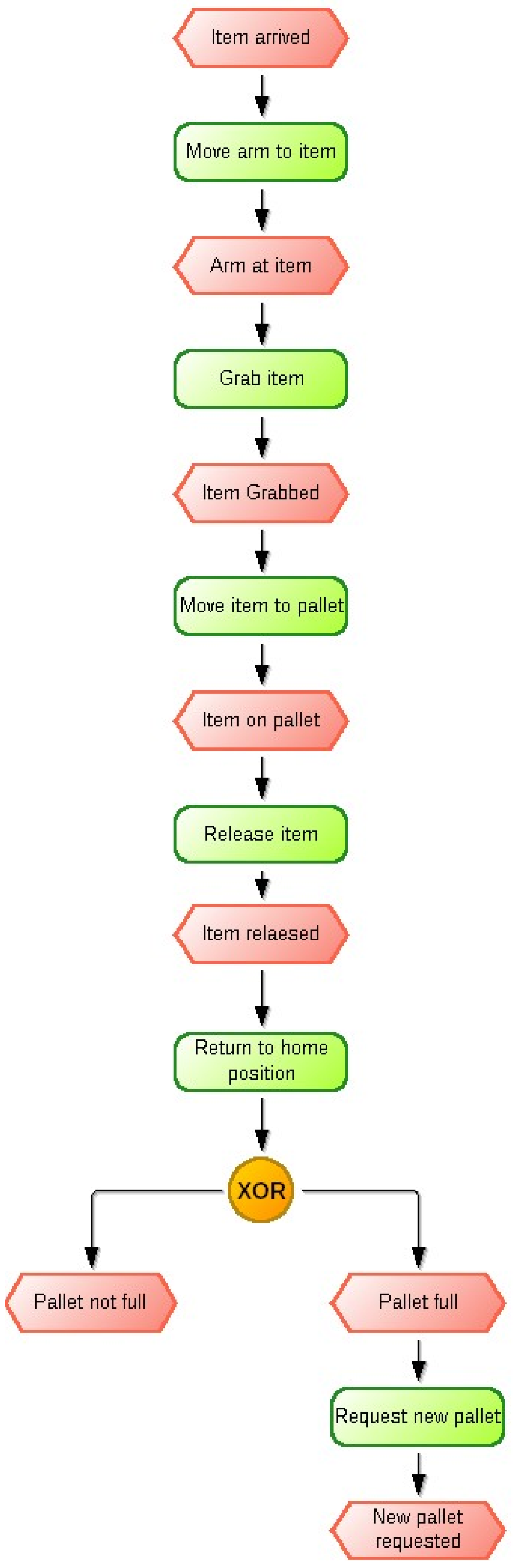
Exercise two traffic lights with Petri Net

Possible solution:




```
Strategy:     default
Cold trans.:  always fire
Iterations:   10
Show log:     yes        Delay: 1.0
```

Exercise pick and place robot ECP
Conveyor belt bring new items and they are moved to pallet with robotic arm., when pallet is full new one is requested.


Item arrived
Move arm to item
Arm at item
Grab item
Item Grabbed
Move item to pallet
Item on pallet
Release item
Item relaesed
Return to home position
XOR
Pallet not full
Pallet full
Request new pallet
New pallet requested

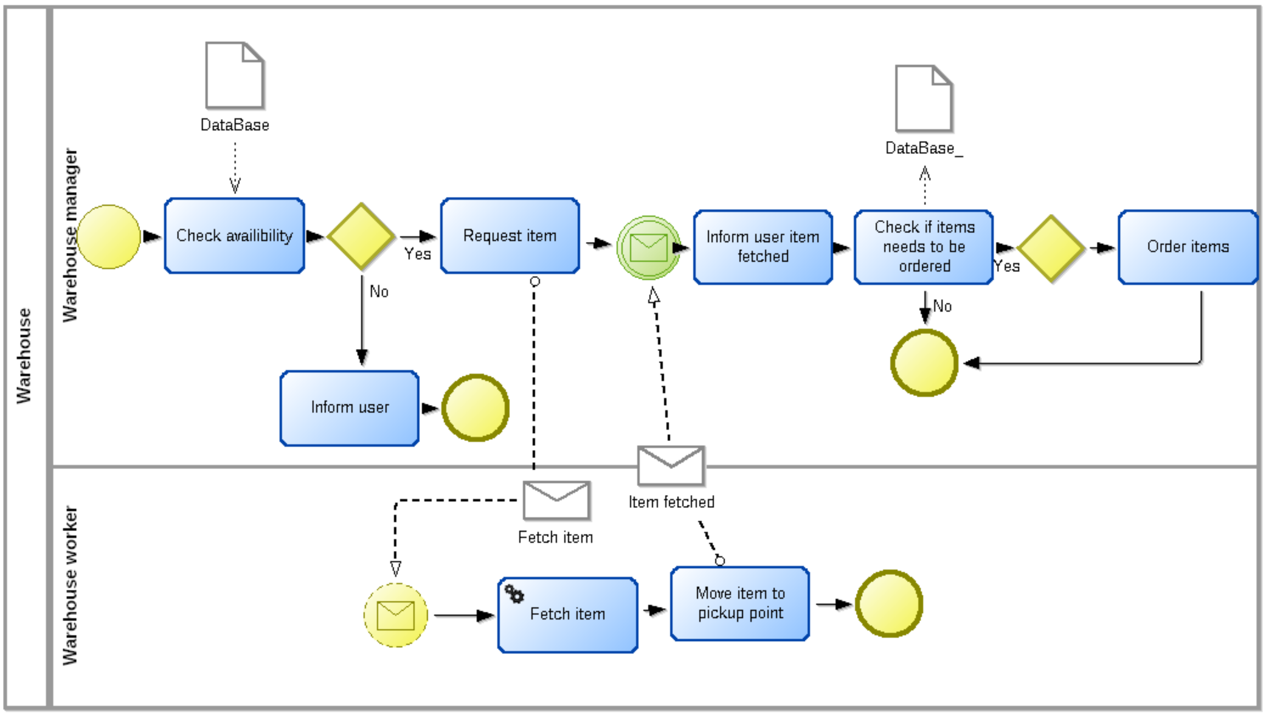
9.2 *Warehouse example:*

Exercise warehouse BPMN:

User request item to be delivered system checks item availability and handles inventory management. Warehouse system fetches item and moves it to pickup point.

Two systems one for managing inventory and one for getting items. What or who actually perform these tasks doesn’t matter.

Human readable level.



Flowchart of item fetcher

Mobile robot with arm. Different levels of details.

- Locate item In warehouse
- Autonomous driving to item
- Finding and grabbing item
- Loading item to robot
- Autonomous driving to pickup point
- Unloading item
- Informing uesr

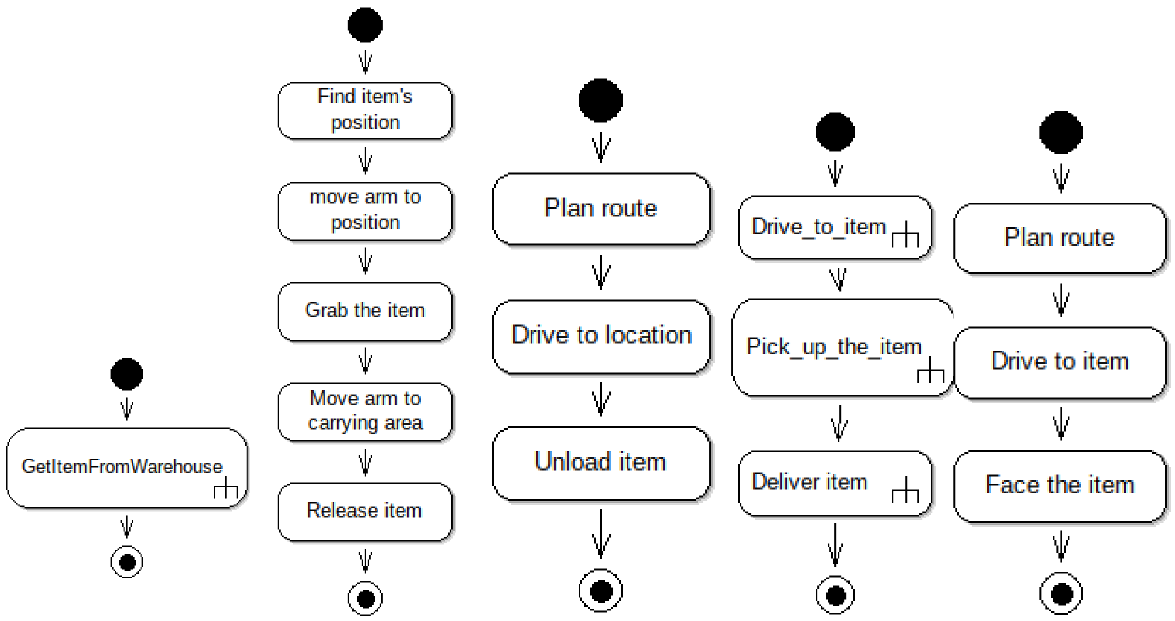


Illustration 2: Example of task refiment

10 Conclusions

You should have some idea of modelling and how to create models. Familiarity with basic usage of BeeUp.

BeeUp tool can be used to create models in different modelling languages. Conceptual models allow simple presentation of complex systems. Models can be used for designing systems