

AUTOMATION OF ASSEMBLY LINES ASSISTED BY A ROBOTIC ARM AND A MOBILE ROBOT

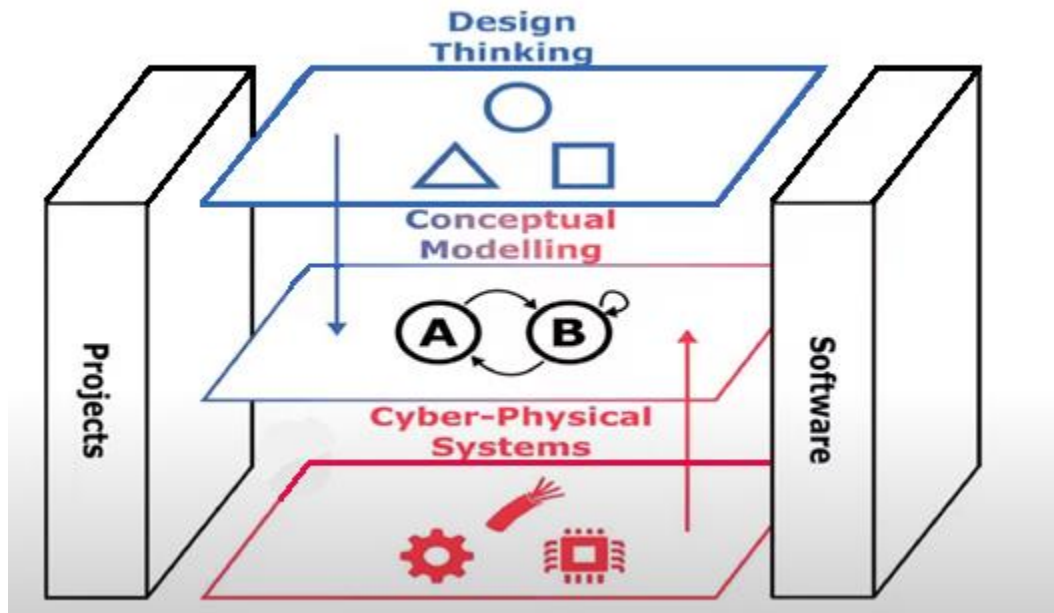
Baltes Octavian-Isaia
Lucian Blaga University of Sibiu
octavian.baltes@gmail.com

Objectives

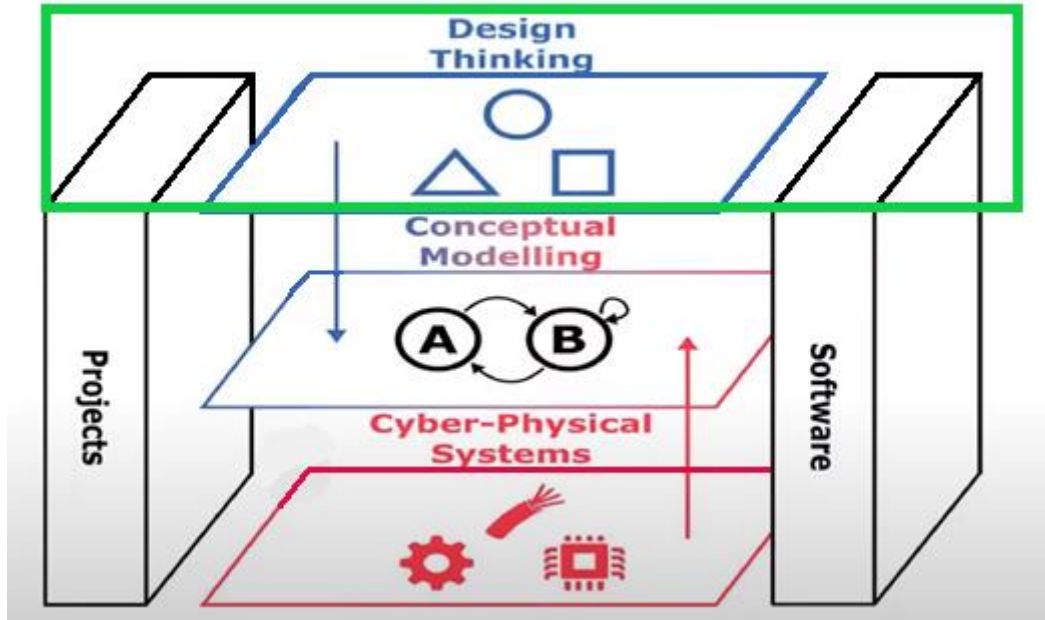
- Simulation of assembly line automation using modeling languages.
- Handling a robotic arm and a mBot.
- Creating the model to solve the problem.
- Writing the code to solve the problem.
- Prototype to simulate the AGVs in logistic processes.

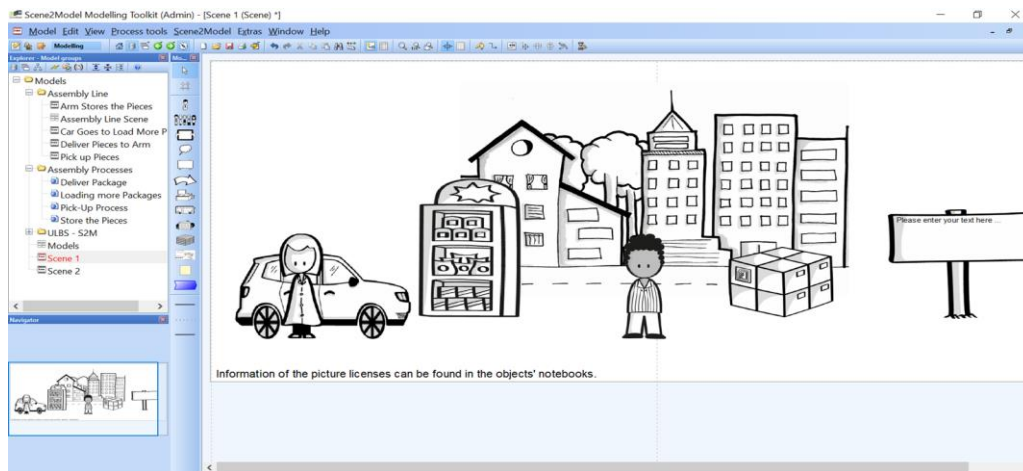
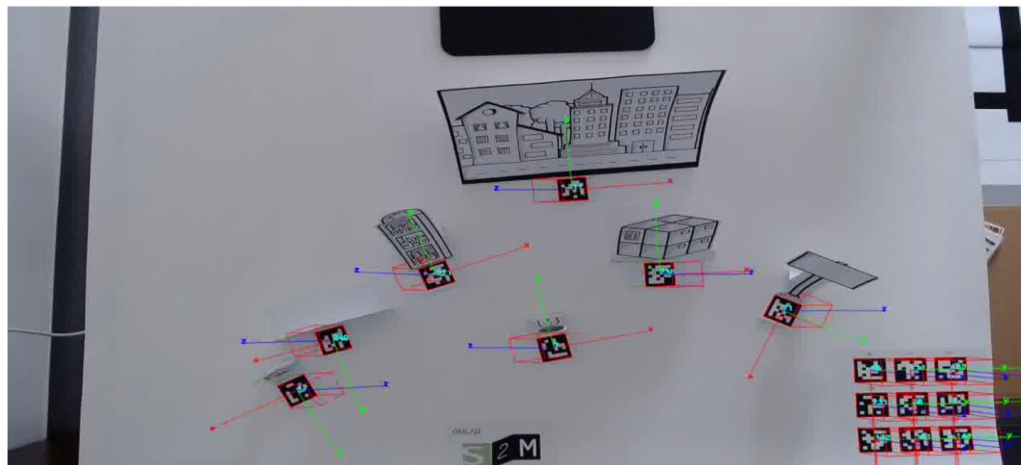


The 3 main approaches suggested by OMiLAB



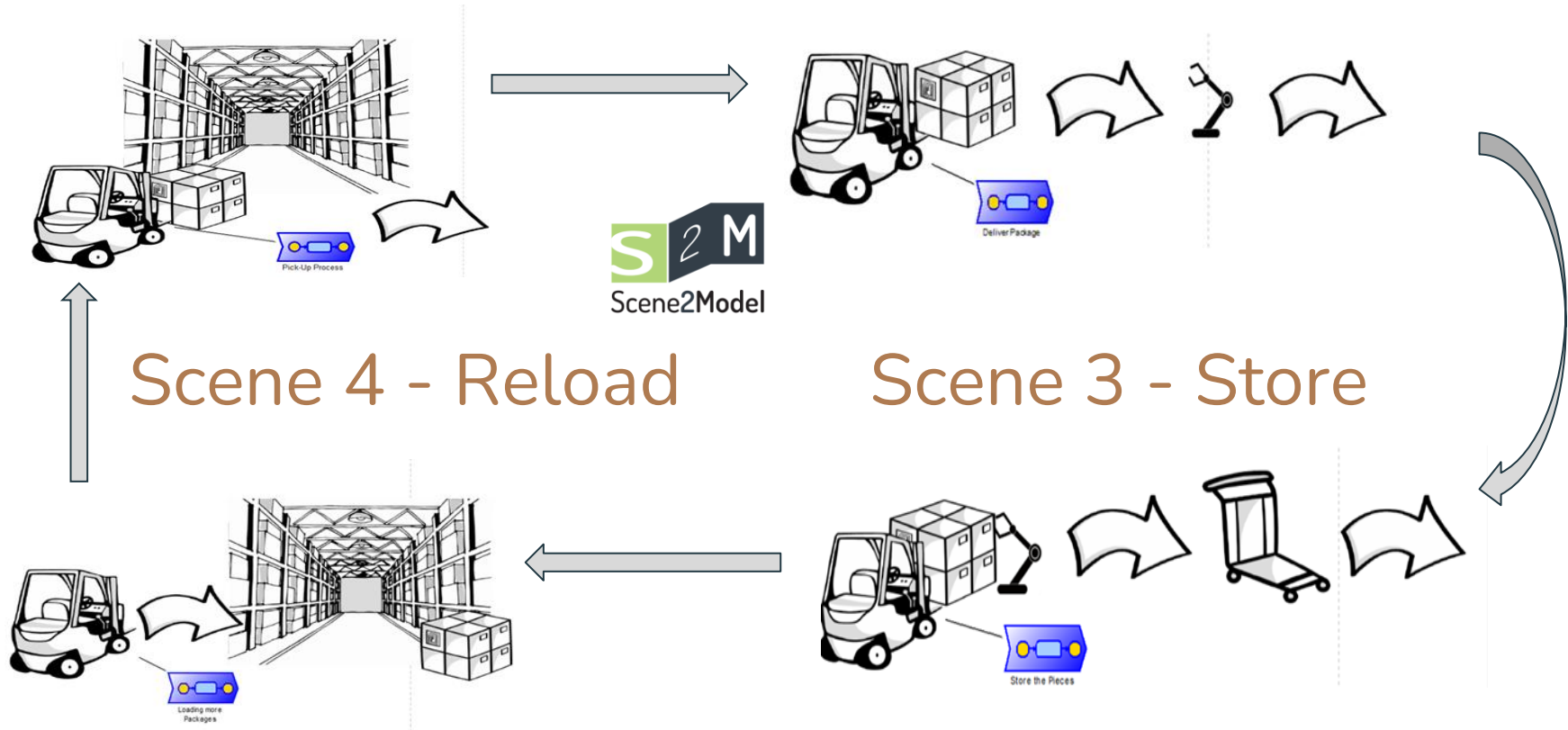
1. Design Thinking



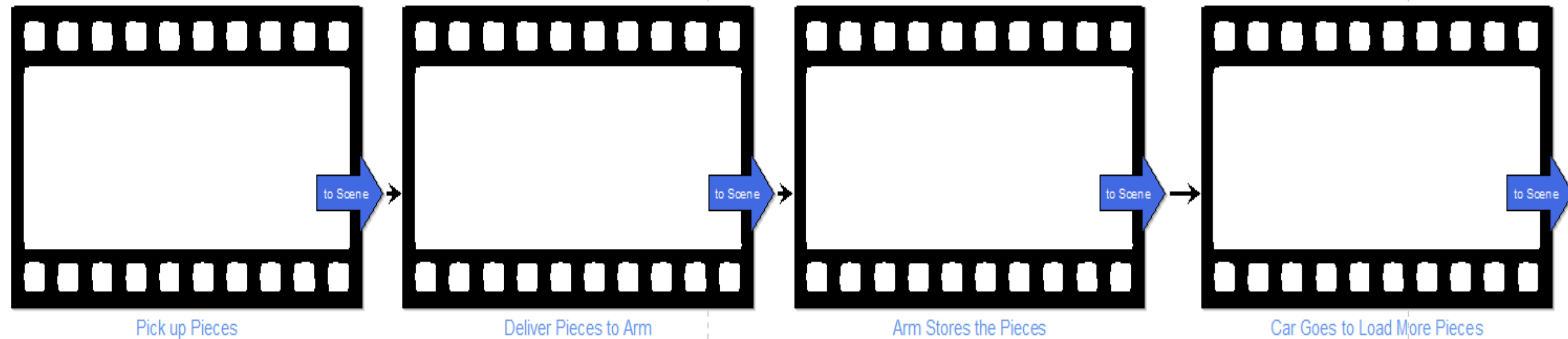


Scene 1 - Pick-up

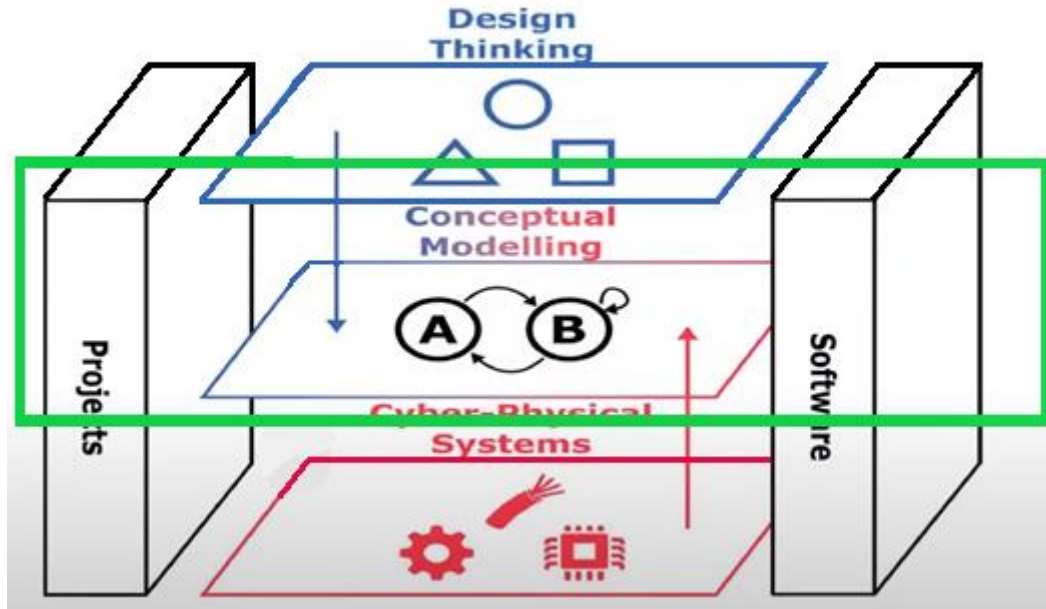
Scene 2 - Deliver

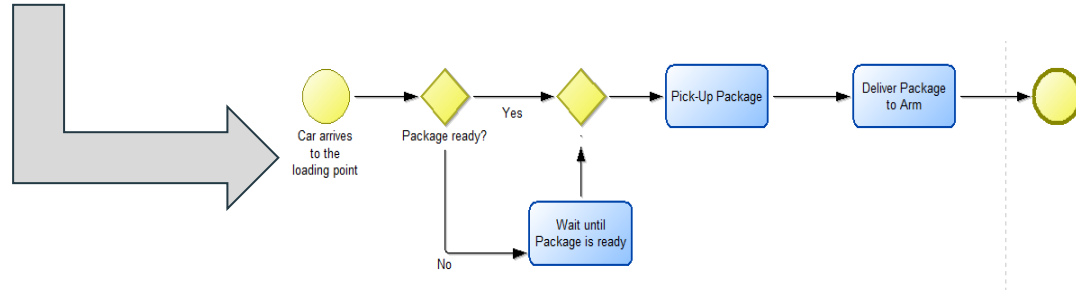
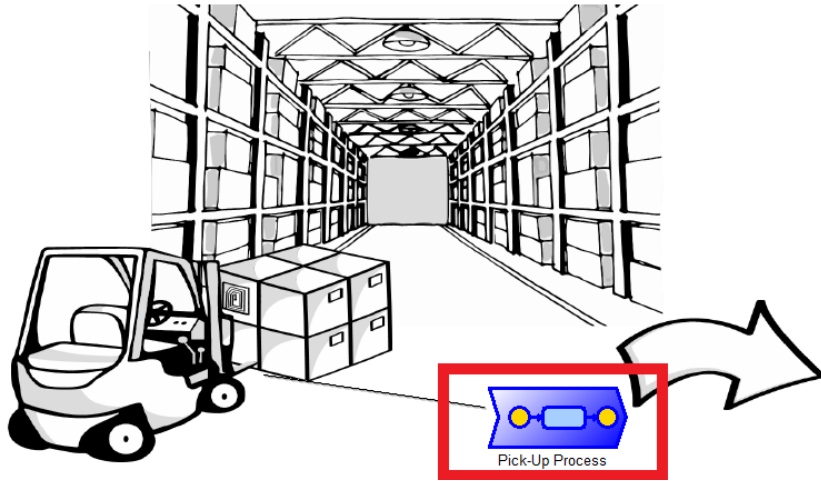


Storyboard

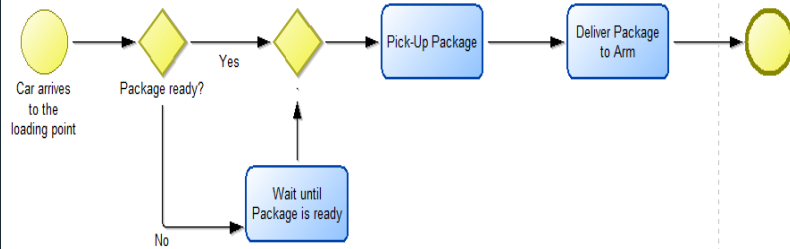


2. Conceptual Modelling

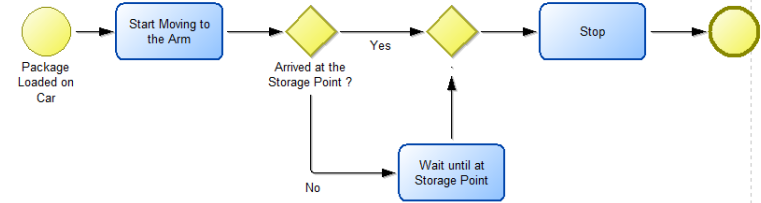




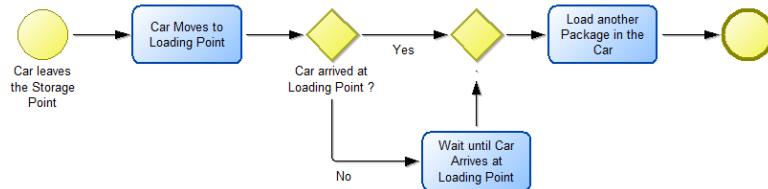
Scene 1 - Pick-up



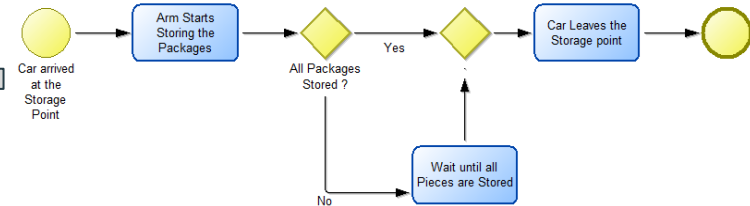
Scene 2 - Deliver



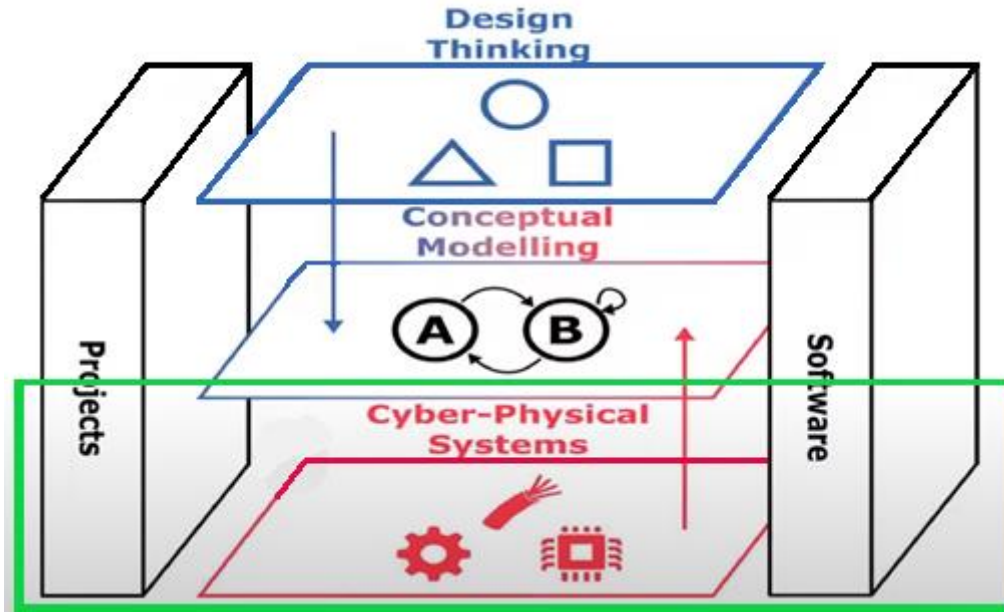
Scene 4 - Reload

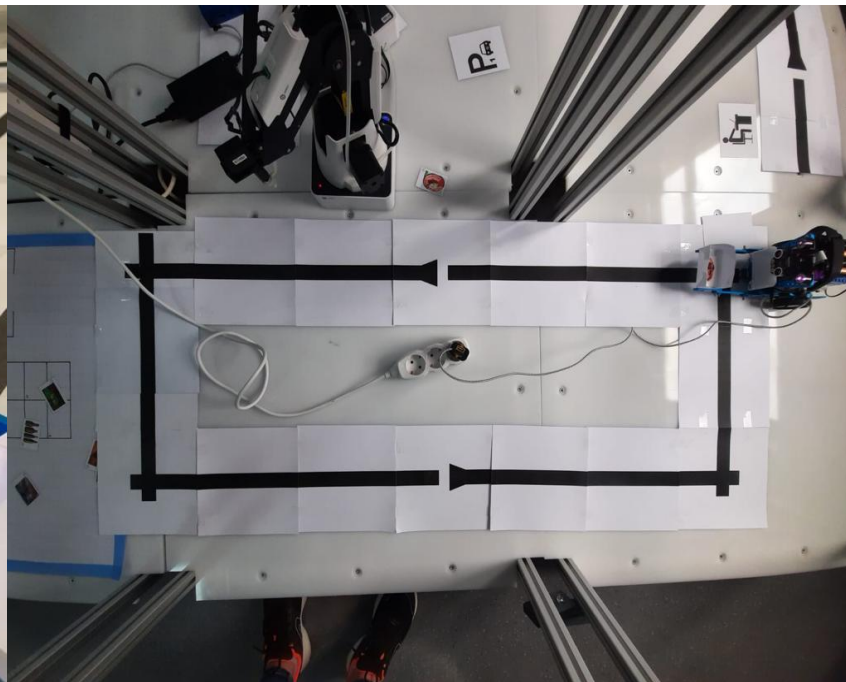
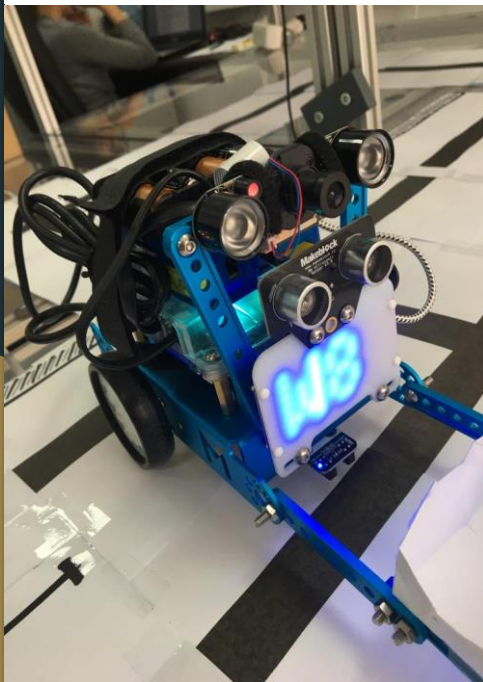


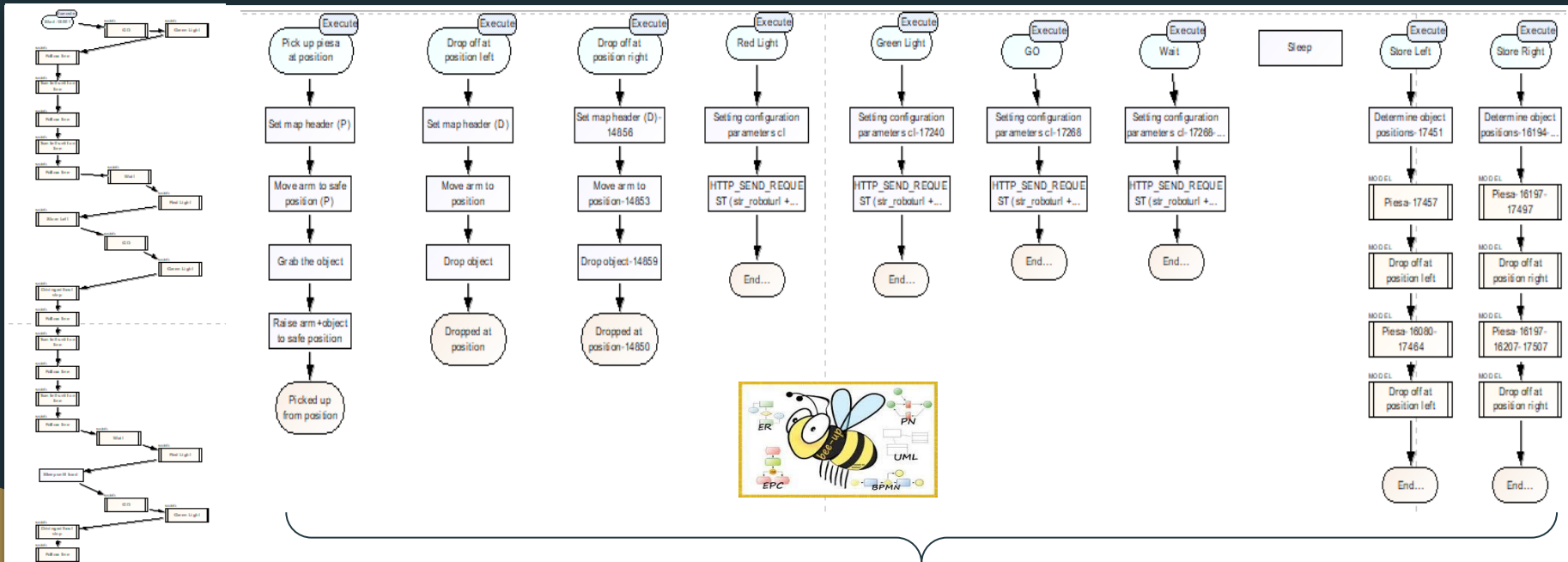
Scene 3 - Store



3. Cyber-Physical System







Sub-models used to create the main model.

← The complete model of the application.

MODEL

Drop off at
position left



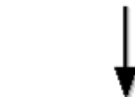
Execute

Drop off at
position left



Required Variables:

	Variable name	Variable type	Default value
1	str_roboturl	string	"http://10.14.10.253:8080/dobot/api/operation/"
2	xPos	any	10
3	yPos	any	211
4	zPos	any	30



Set map header (D)



Operation code:

```
map_headers := {"Content-Type": "application/json"}
```



Move arm to
position



Operation code:

```
HTTP_SEND_REQUEST (str_roboturl + "moveToPosition?x=" + STR xPos +
"&y=" + STR yPos + "&z=" + STR zPos) str_method:("POST") map_reqheaders:
(map_headers) str_reqbody:("") val_respcode:val_httpcode
map_respcode:map_respcode str_respcode:str_respcode
```



Drop object



Operation code:

```
HTTP_SEND_REQUEST (str_roboturl + "turnOffSuctionCup")
str_method:("POST") map_reqheaders:(map_headers)
str_reqbody:("") val_respcode:val_httpcode
map_respcode:map_respcode str_respcode:str_respcode
```

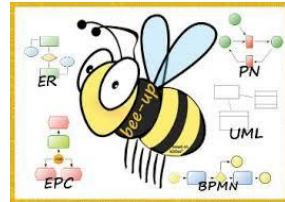


Dropped at
position



ULBS

Universitatea "Lucian Blaga" din Sibiu



Conclusions and further developments

I believe that this type of project has wide applicability and can be successful in industrial companies.

From a software point of view, I used Bee-Up as a tool for the CPS part and the Flowchart as a working method, and for the Design Thinking and Conceptual Modeling parts I used Scene2Model.

In the future, I want to make the robotic arm smarter, attaching a camera so that the parts are separated based on QR codes. Apart from the automation problem, I want to make a project in which the arm is used as a 3D printer.

As for the mBot, I want to attach at least 2 proximity sensors so that it can move alone on an obstacle course.

References

- Dimitris Karagiannis, Heinrich C. Mayr, John Mylopoulos – “Domain-Specific Conceptual Modeling”
- Bee-Up Tool : <https://austria.omilab.org/psm/content/bee-up/info>
- Scene2Model Tool : <https://austria.omilab.org/psm/content/scene2model/info>
- ADOxx : <https://www.adoxx.org/live/home>
- OMiLAB Approach : https://www.omilab.org/assets/docs/OMiLAB%20Laboratory%20Layout_DRAFT.pdf



Thank you for your attention !

