Centralized Goal Reasoning for Logistics Robots

Lab Course Winter Term 2021/2022

Till Hofmann, Tarik Viehmann

October 13, 2021

Content

Goals

For You

- Get familiar with robot software development
- Learn about systems, tools, robot control
- In particular: in-depth contact with reasoning system

For us

- Sparring partner for our robot agent
- Collect ideas for future improvements of our system
- Data for comparison of systems

Have fun in the process!

Smart Factory in the Context of Industry 4.0

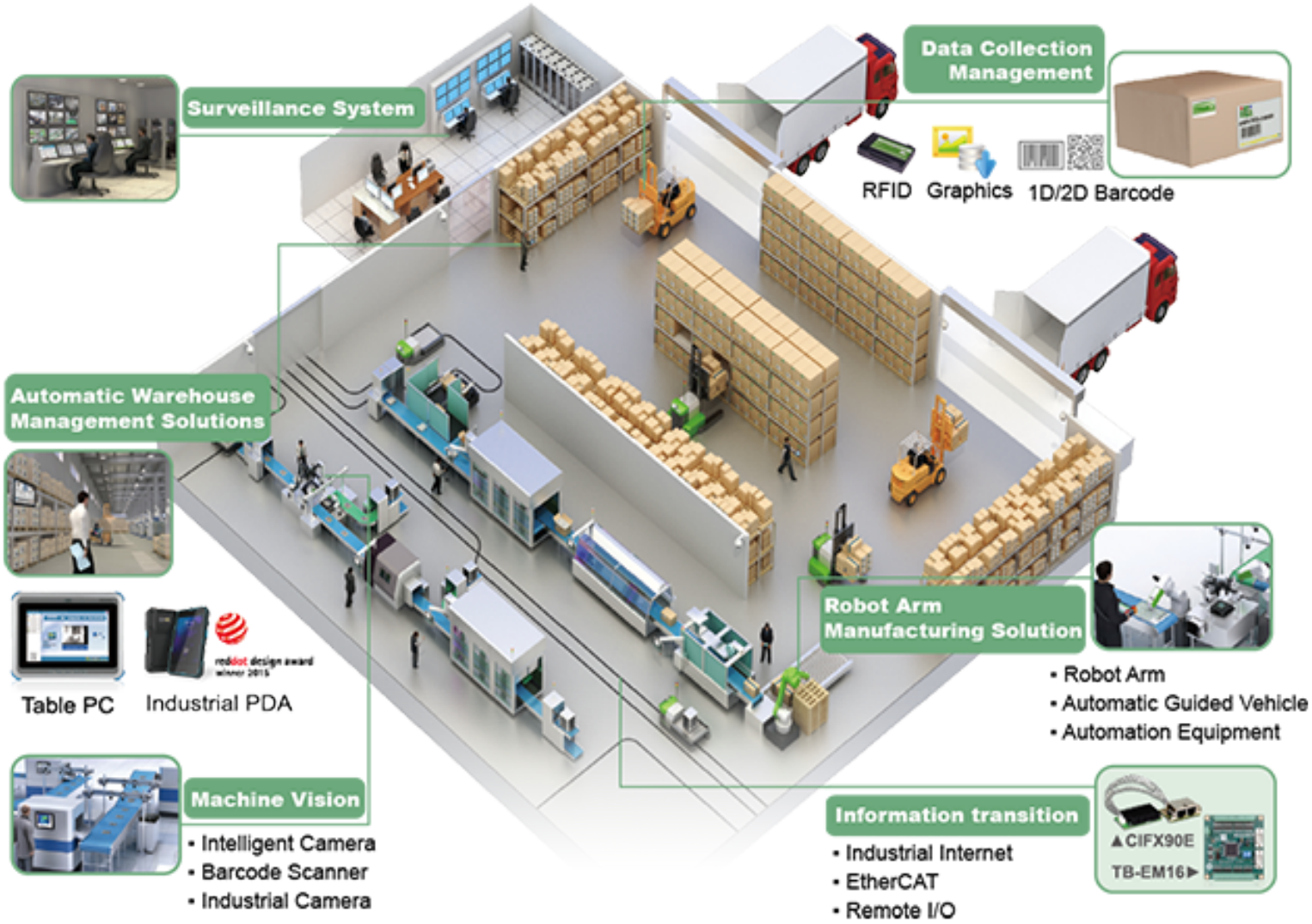


Image credit: *IEI Corp.*

What is Industry 4.0?

- *Old:* Linear production, only a few product variants (*Ford Model T*)
- *New:* Individual production, lot size 1

$\Rightarrow$ *Smart Factories*

- *Old:* Manual labor, “dumb” machines (e.g., exactly one motion)
- *New:* Physical and software components are intertwined

$\Rightarrow$ *Cyber-Physical Systems*

- *Old:* Almost no production data, local instruments on the machine
- *New:* Machines produce a continuous data flow

$\Rightarrow$ *Internet of Production*

RoboCup Logistics League

- Replication of a smart factory
- Part of the RoboCup Robotics competition
- Robots have to operate machines to manufacture products
- Machines are static and can do one step of a multi-step production
- Teams have to fulfill dynamic orders

RoboCup Logistics League

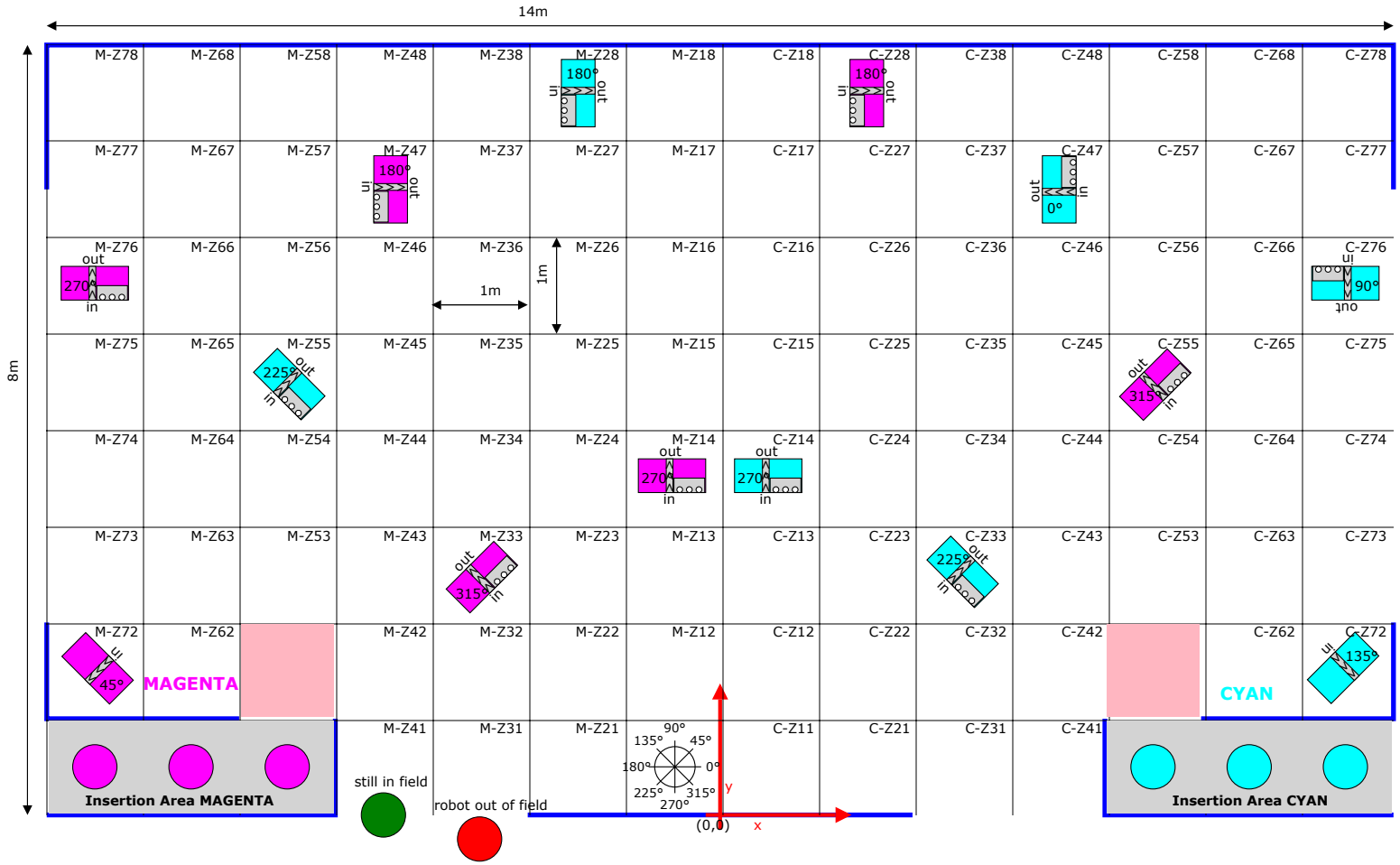
RoboCup Logistics League

Game Basics

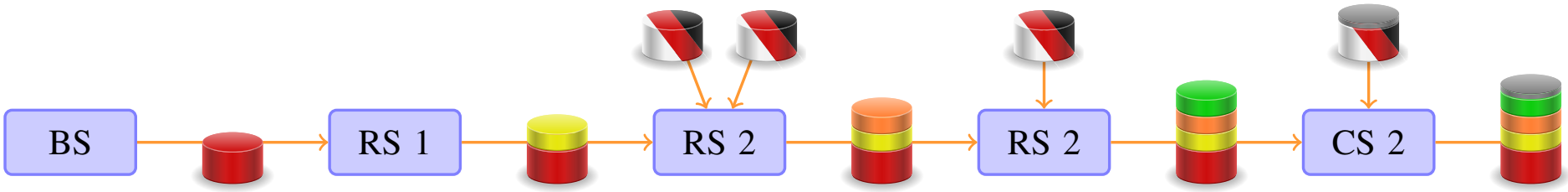
- Task: In-factory production logistics
- Goal: variant production
- Product orders are placed dynamically
- Two teams playing on common field
- Each team has 3 robots
- Multi-robot coordination task

RoboCup Logistics League

Playing Field



RoboCup Logistics League – Production



Product Composition

- Products of four complexities (number of rings)
- Base (3 colors) + 0–3 rings (4 colors) + cap (2 colors)
- Order of ring colors is important
- Some ring colors require additional material
- Actual product variants randomized by referee box
- Orders have lead time of a few minutes

Order Elements (posted dynamically by refbox)

- Product to deliver (and number thereof)
- Time window in which to deliver

RoboCup Logistics League – Machines

Common

- Based on Festo MPS
- Marker to identify machine
- Signal light to indicate state
- Each team has exclusive set
- Similar handling for all types

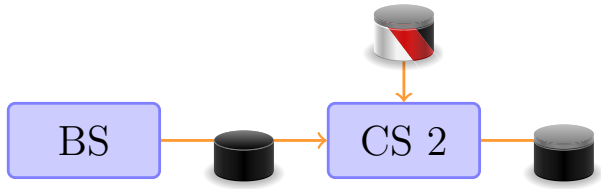
Machine Types (per team)

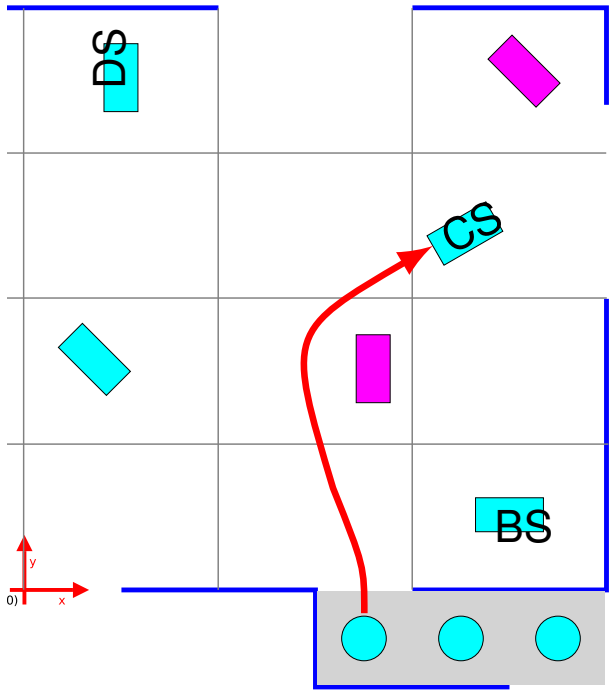
1× Base Station (BS): retrieve bases
2× Ring Station (RS): mount colored rings
2× Cap Station (CS): buffer/mount caps
1× Storage Station (SS): buy products
1× Delivery Station (DS): final delivery

RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS

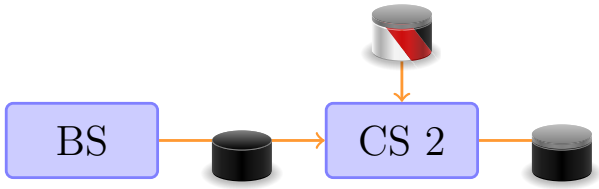


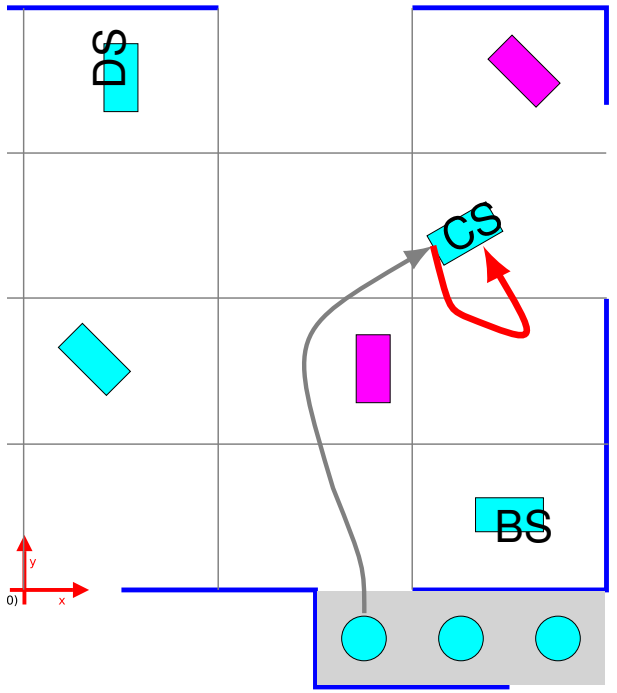


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS

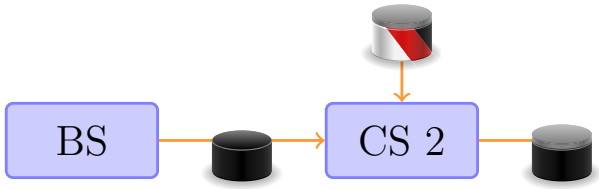


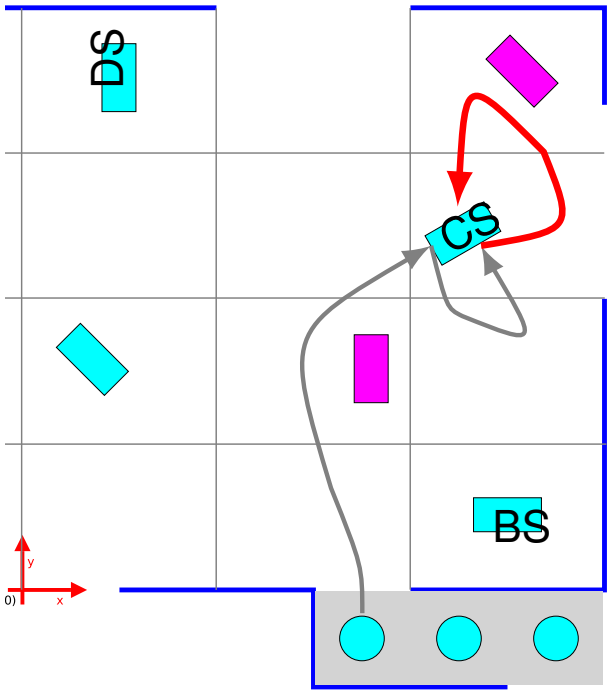


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base

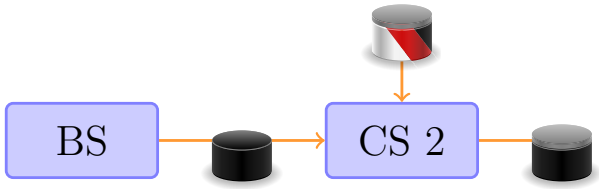


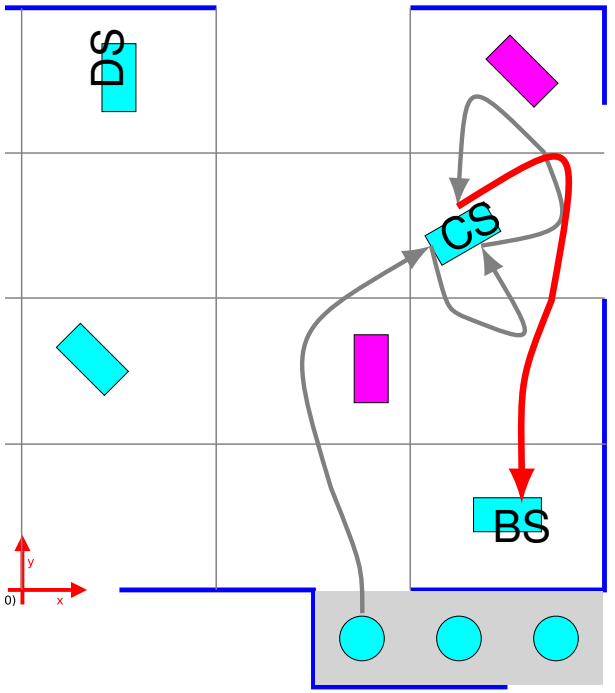


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS

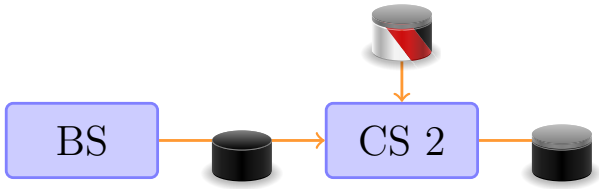


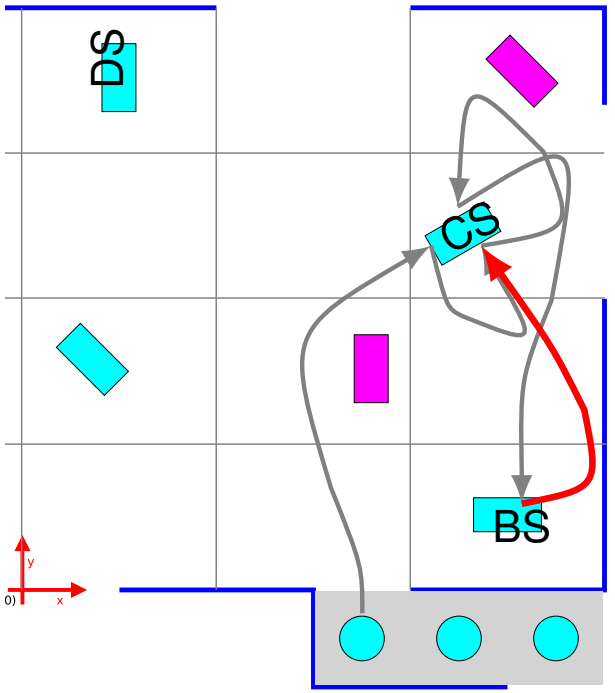


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS

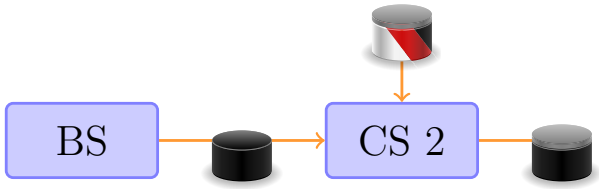


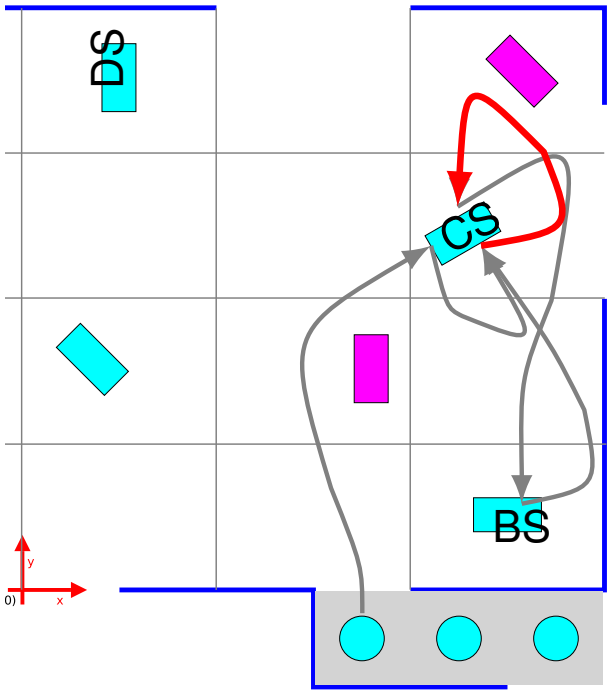


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS

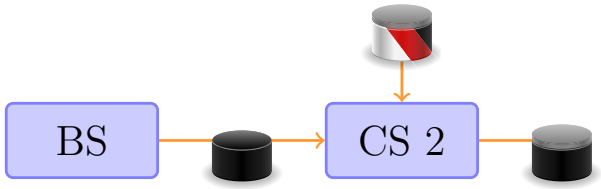


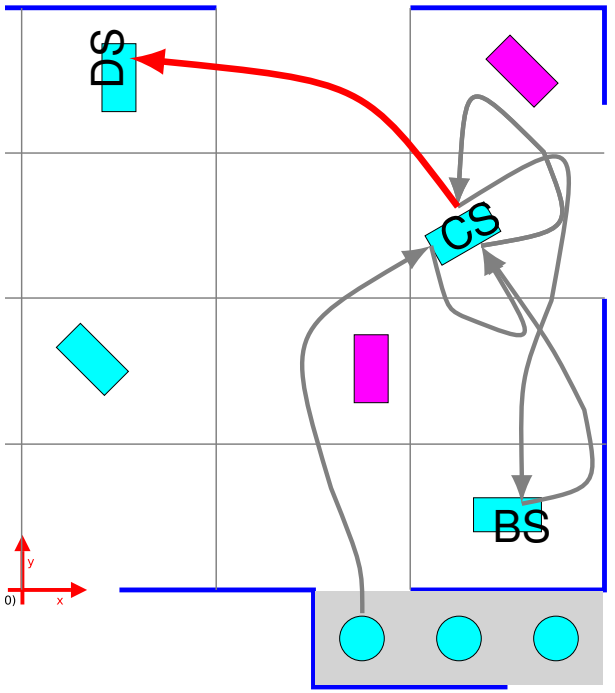


RoboCup Logistics League – Production Example

C_0 Production

- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS

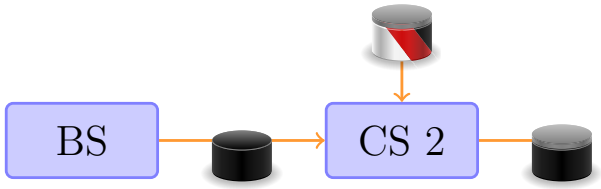


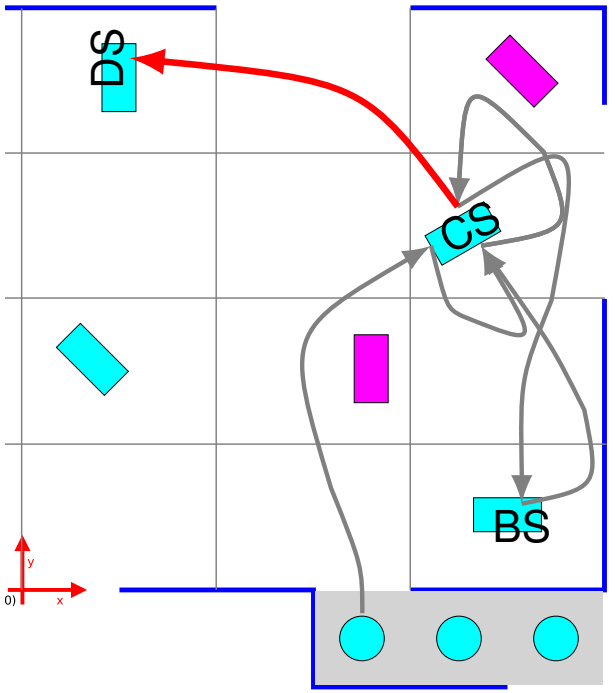


RoboCup Logistics League – Production Example

C_0 Production

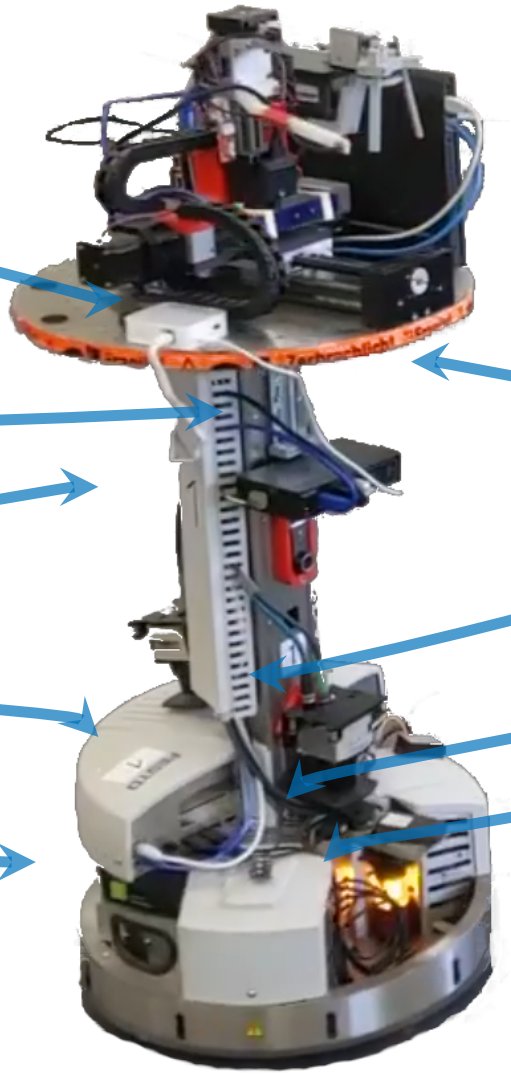
- Retrieve base with cap from shelf at CS
- Prepare CS to retrieve cap
- Feed base into CS
- Discard cap-less base
- Prepare BS to provide black base
- Retrieve base from BS
- Prepare CS to mount cap
- Feed black base to CS
- Retrieve black base with cap from CS
- Prepare DS for slide specified in order
- Deliver to DS





Already simple product has several fragile points and cooperation potential.

RCLL Robot Platform (Team Carologistics)



Game Phases

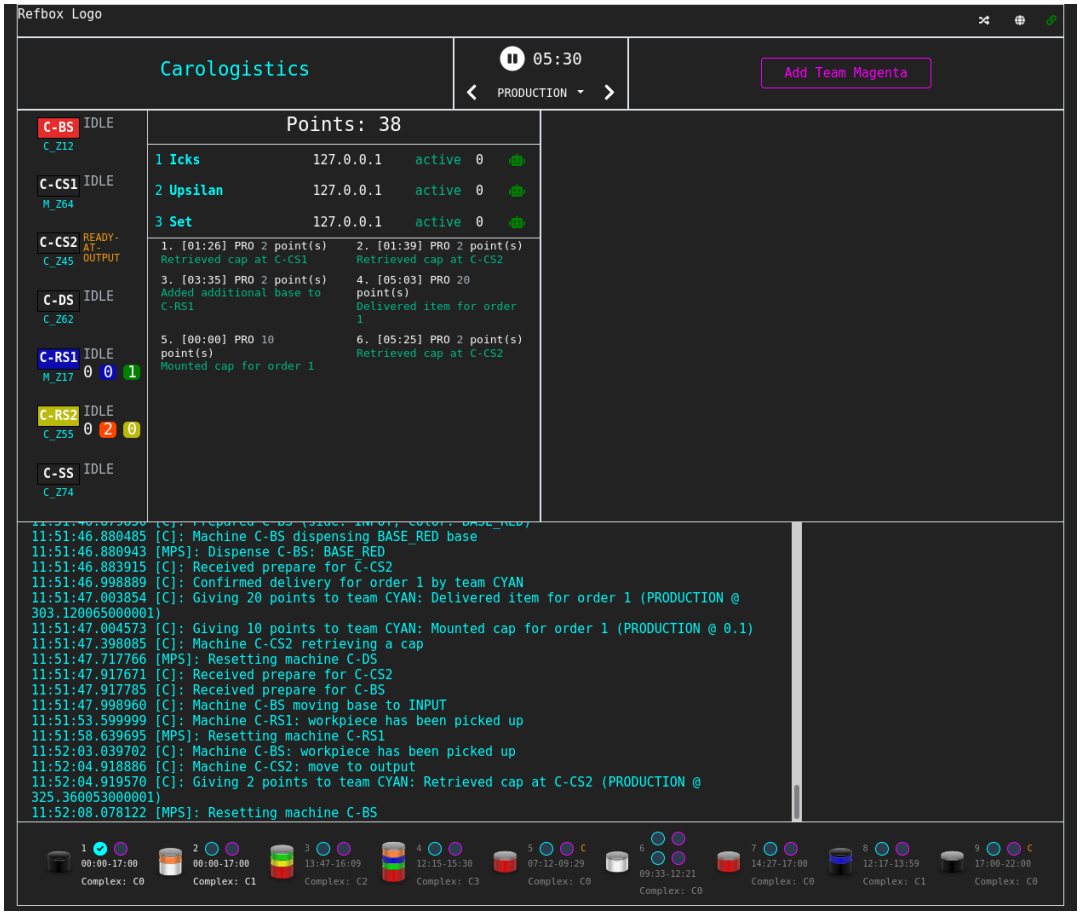
Exploration (3 min)

- Machines are placed randomly on the field (with constraints)
- Robot must recognize and announce machine position and type

Production (17 min)

- Orders are posted dynamically, e.g.
  “Deliver 1 product with red base, yellow and green ring, gray cap in time window [123, 206] to gate 3”
- Robots must complete production chain leading to products
- Coordination is required for effective resource usage
- Machines may go out-of-order

Semi-autonomous Referee Box



Tasks

- Determines randomized orders and machine failures
- Posts orders dynamically
- Scoring and evaluation
- Instructs MPS stations
- Provides controls to the user

Planning and Benchmarking

- Accountable environment agency
- Same controller in simulation
- Records extensive data
- Limited uncertainty

$\Rightarrow$ **Repeatable benchmarks**

Videos

- RoboCup 2017 finals
- Production Challenge C3
- RoboCup 2019

Carologistics RoboCup Team

- World champion 2014, 2015, 2016, 2017, 2019, 2021
- Fully integrated system released as Open Source Software

https://www.carologistics.org

Software Stack Overview

Fawkes

- Robot Software Framework and middleware
- Provides building blocks and connects software components

Behavior System

- Behavior Engine: reactive execution middle layer
- CLIPS Executive: goal reasoning system written in CLIPS

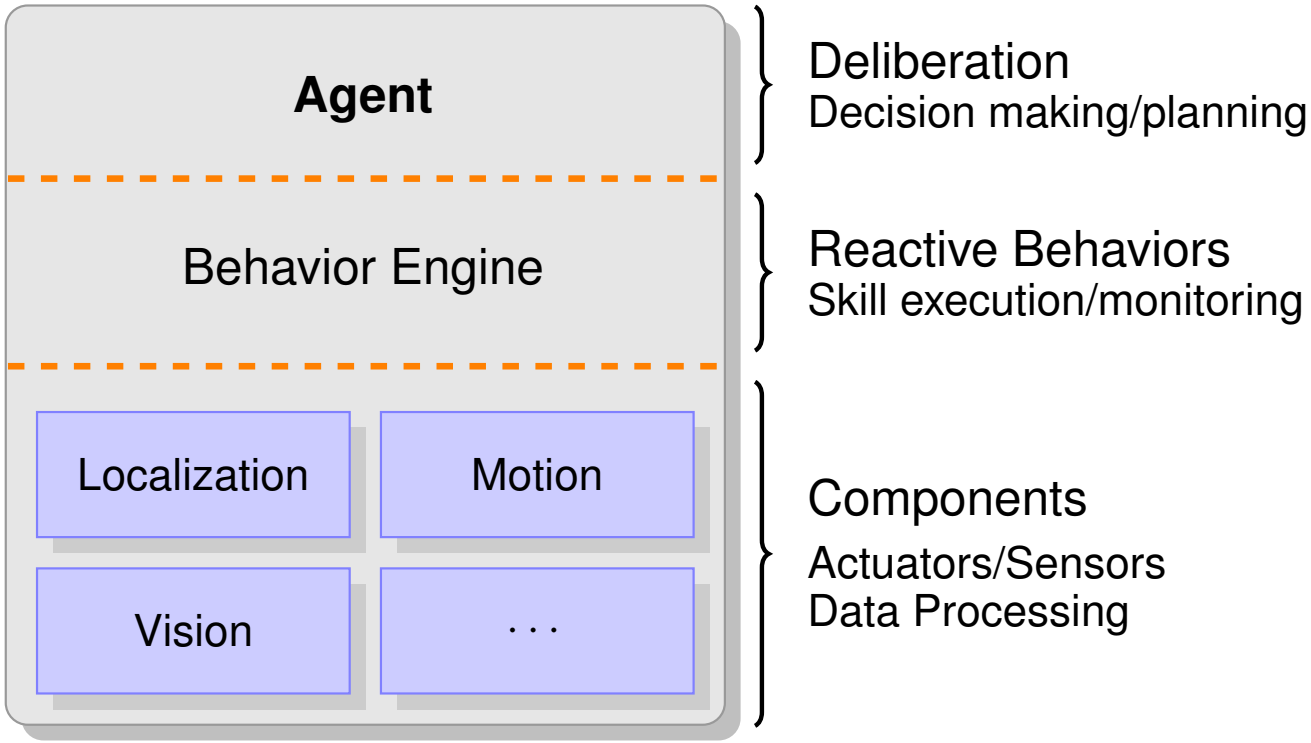
Fawkes

- Robot Software Framework providing basic building blocks
- *Component-based* architecture (plugins)
- Hybrid *BlackBoard/messaging* data exchange
- Multi-threaded and distributable
- Aspect-oriented framework feature access
- Structured and synchronized main loop

`https://www.fawkesrobotics.org`

[SIMPAR 2010]

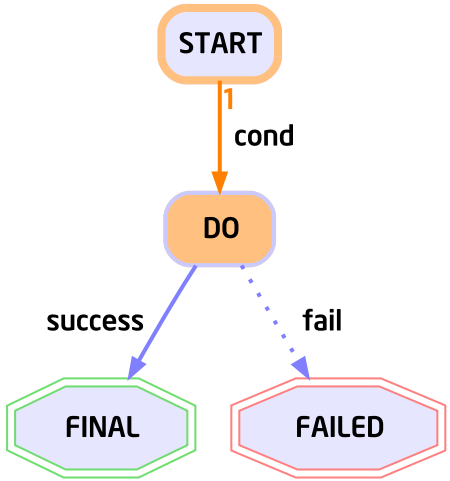
Behavioral Architecture



[AAAI Spring Symp 2013]

Lua-based Behavior Engine

- Basic actions for reasoning layer
- Emphasize description over programming
- Allow programming where necessary
- Modeled using Hybrid State Machines
- Abstract low-level system
- Implemented for Fawkes and ROS
- Written in the Lua scripting language



Variable table

x	5.2
y	4.3
error	
. . .	

fawkesrobotics.org/p/behavior-engine

[RoboCup 2009]

Goal Reasoning with the CLIPS Executive

- Typically: reason about actions, but goal is fixed
- *Goal reasoning*:
  - Explicitly represent goals
  - Continually reason about goals
  - Dynamically adjust and prioritize goals
  - Model flow along *goal life cycle*
- Reason about *what* to accomplish, only then *how* to accomplish it

[Aha, 2018, Roberts et al., 2014]

Goal Life Cycle

Components:

Goal Reasoner

Formulates and expands goals

Goal Expander

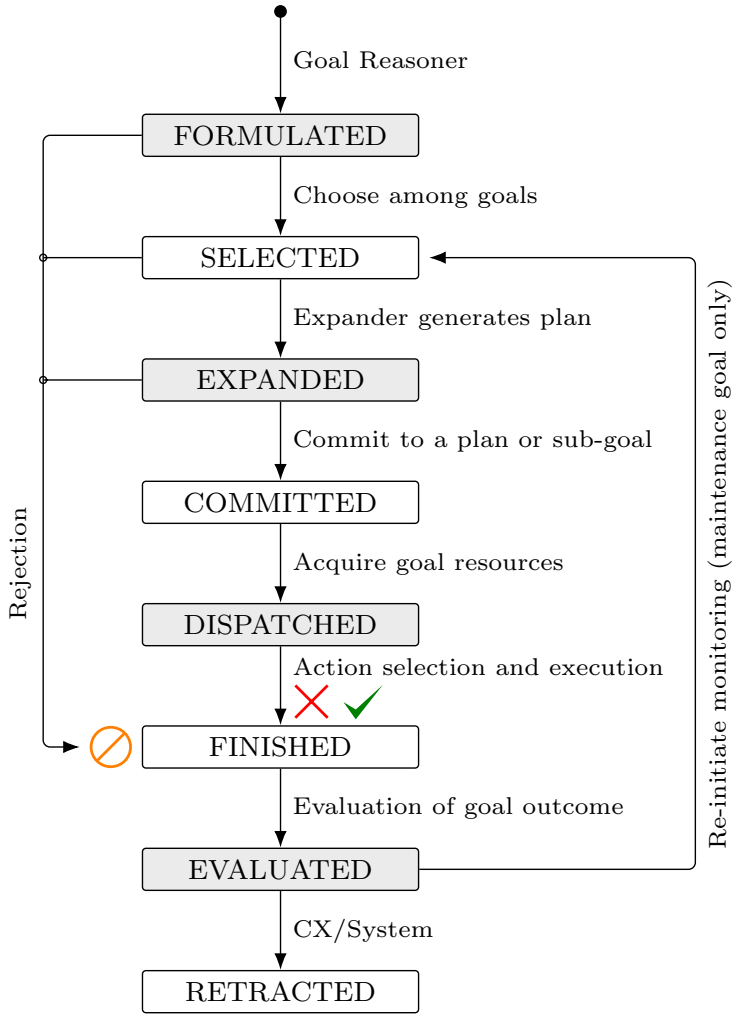
Expands a goal into a plan,
e.g., with PDDL planning

Action Executor

Executes a single action on the robot,
e.g., with the behavior engine

Monitor

Monitors the execution of a plan,
adapts plan dynamically,
e.g., by retrying actions



[Roberts et al., 2014, ICAPS 2019, RoboCup 2019]

Project Outline

Goal

Implement a centralized goal reasoning agent for the RoboCup Logistics League in the CLIPS Executive.

Steps

1. Get familiar with Fawkes, the RCLL, and the CLIPS Executive
2. Accomplish simple tasks with a single robot
3. Familiarize yourself with the existing distributed goal reasoning
4. Develop a centralized agent

Project Outline

Goal

Implement a centralized goal reasoning agent for the RoboCup Logistics League in the CLIPS Executive.

Milestones

November	Get acquainted to the overall system, call skills
December	Simple drive-around tasks
January	Production phase
February	Refine agent strategy
March	Conclusion and Tournament

Perspective

labcegor 2020

- So far: decentral, incremental agent
- Let students figure out how to write a central agent
- Very good performance, high peaks, less robust
- Perfect simulation environment

labcegor 2021

- Now we started building a central agent as well
- Still development, not capable of playing full RCLL game yet
- Execution Monitoring of key interest
- Here: No need to start from scratch, feel free to look into our code
- Focus on wide set of benchmarks, including error simulations etc

Administrivia

Rescission Policy

Up to three weeks **from now on** you are allowed to recede from the seminar without any consequences.
A later rescission will be graded as a failed attempt!

Accounts

- Register on GitHub and send me your handle (after this talk)
- Join our Slack
- Split into groups of 2 or 3 students
- Group accounts for lab machines

Working Environment

- Lab is open for usage (2G)
- Remote desktop to lab machines
- Ask/discuss on Slack whenever you want
- One lab day per week for general discussion

Conclusion and Questions

Implement a centralized goal reasoning agent for the RoboCup Logistics League in the CLIPS Executive.

- Logistics League Simulation based on Gazebo
- Centralized goal reasoning with the CLIPS Executive
- Document thoroughly and precisely
- Beat the world champion, at least sometimes
- **Have fun and join efforts**

It's your turn

1. Please introduce yourself:
   - Course of study
   - Background and previous experience
     (e.g., lectures, programmming languages)
   - Robotics background, if any?
2. Questions?
3. Find groups
4. Determine lab day

Next time:

- In-depth introduction to goal reasoning with the CLIPS Executive
- Remote access to lab machines
- Setting up your working environment

References I

Tim Niemueller, Alexander Ferrein, Daniel Beck and Gerhard Lakemeyer.

Design Principles of the Component-Based Robot Software Framework Fawkes.

Proceedings of the Second International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2010), 2010.

Tim Niemueller, Alexander Ferrein and Gerhard Lakemeyer.

A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao.

RoboCup 2009: Robot Soccer World Cup XIII, 2009.

Tim Niemueller, Alexander Ferrein and Gerhard Lakemeyer.

Incremental Task Level Reasoning in a Competitive Factory Automation Scenario.

AAAI Spring Symposium: Designing Intelligent Robots, 2013.

David Aha.

Goal Reasoning: Foundations, Emerging Applications, and Prospects

AI Magazine, 39/2, 2018.

References II

Mark Roberts, Swaroop Vattam, Ronald Alford, Bryan Auslander, Justin Karneeb, Matthew Molineaux, Tom Apker, Mark Wilson, James McMahon, and David Aha.

Iterative Goal Refinement for Robotics

WS on Planning and Robotics (PlanRob) at International Conference on Automated Planning and Scheduling (ICAPS), 2014.

Tim Niemueller, Till Hofmann and Gerhard Lakemeyer.

Goal Reasoning in the CLIPS Executive for Integrated Planning and Execution.

Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS)

Till Hofmann, Nicolas Limpert, Victor Mataré, Alexander Ferrein, Gerhard Lakemeyer.

Winning the RoboCup Logistics League with Fast Navigation, Precise Manipulation, and Robust Goal Reasoning.

RoboCup Symposium – Champion Teams Track