The Carologistics RoboCup Logistics Team 2025

Daniel Swoboda[2], Matteo Tschesche[3], Tarik Viehmann[1], Alexander Ferrein[3], and Gerhard Lakemeyer[1]

[1] Knowledge-Based Systems Group, RWTH Aachen University, Germany
[2] Chair of Machine Learning and Reasoning, RWTH Aachen University, Germany
[3] MASCOR Institute, FH Aachen University of Applied Sciences, Germany

Abstract. The Carologistics team participates in the RoboCup Logistics League for the thirteenth year. The RCLL requires precise vision, manipulation and path planning, as well as complex high-level decision making and multi-robot coordination. We provide an overview of our approach with focus on navigation, perception, and high-level reasoning.

The team members in 2025 are Saurabh Borse, Himanshu Grover, Henning Hedemann, Samridhi Kalra, Zhen Yan Khaw, Sam Köhler, Ulzhalgas Rakhman, Daniel Swoboda, Matteo Tschesche, Tarik Viehmann, Tim Wendt, and Jacob Weyer.

This paper is based on 2024's team description [22].

1 Introduction

The Carologistics RoboCup Team is a cooperation of the Knowledge-Based Systems Group (RWTH Aachen University), the Chair of Machine Learning and Reasoning (RWTH Aachen University), and the MASCOR Institute (FH Aachen University of Applied Sciences). The team was initiated in 2012. Doctoral, master, and bachelor students of all partners participate in the project and bring in their specific strengths tackling the various aspects of the RoboCup Logistics League (RCLL): designing hardware modifications, developing functional software components, system integration, and high-level reasoning and control of a set of cooperating mobile robots.

Our team has participated in RoboCup 2012–2024 and the RoboCup German Open (GO) 2013–2025. We were able to win the GO 2014-2019 and 2025 as well as the RoboCup 2014-2017, 2019, and 2021-2022 [16,17,15,4,3,24,26], demonstrating flexible and efficient task coordination, robust manipulation, reliable collision avoidance and self-localization. In the 2025 season, the main objective is to continue our multi-year project of modernizing our oftware stack by migranting the final components from fawkes to ROS 2. We are also in the process of a major revision of our gripping system to improve speed and reliability. We also hope that this new gripper system will be able to handle more complex manipulation scenarios than the current one, further improving the efficiency of our robots.

In the following, we provide an overview of our system, starting with our robot platform in Section 2. In Section 3, we continue by describing our approach to path planning, before we explain our approach to perception and in particular conveyor belt detection in Section 4. In Section 5, we summarize our current approach to high-level decision making and give an outlook on our upcoming changes, before we conclude in Section 6.

1.1 The RoboCup Logistics League

The RoboCup Logistics League (RCLL) [9] is a RoboCup [6] competition with a focus on smart factories and production logistics. In the RCLL, a team of mobile robots has to fulfill dynamically generated orders by assembling workpieces. To assemble such products, the robots operate and transport workpieces between static production machines. The major challenges of the RCLL include typical robotics tasks such as localization, navigation, perception, and manipulation, with a particular focus on reasoning tasks such as planning, plan execution, and execution monitoring.

The game is controlled by a semi-automatic referee box (*refbox*) [18].

2 The Carologistics Platform

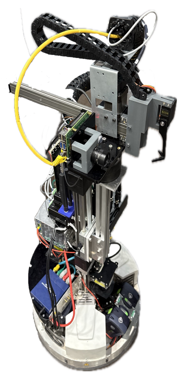

Fig. 1: The Carologistics Robotino

The standard robot platform of this league is the Robotino by Festo Didactic [5]. The Robotino is developed for research and education and features omni-directional locomotion, a gyroscope and webcam, infrared distance sensors, and bumpers. The teams may equip the robot with additional sensors and computation devices as well as a gripper device for product handling. The Carologistics Robotino is shown in Figure 1.

Sensors We use one forward-facing and one tilted, backward-facing SICK TiM571 laser scanner for collision avoidance and self-localization. Using a second laser scanner in the back allows us to fully utilize the omni-directional locomotion of the Robotino. In addition to the laser scanners, we use a webcam for detecting the MPS identification tags, and a Raspberry Pi Global Shutter Cam camera for conveyor belt as well as workpiece detection.

2.1 Gripper System

Our new gripper system consists of two linear axes, one rotational axes, and a one-fingered gripper, as shown in Figure 1. The three axes are driven by stepper motors, which allows movements with sub-millimeter accuracy. The axes are controlled by an Arduino, which in turn receives commands from the Robotino main computer. The linear axis used for forward movement was replaced with a much faster axis changing from thread to a rack axis.

The gripper grasps from above, using one movable, bracket like finger to push the workpiece from the conveyor belt towards the other static gripper, securing it with the force of a servo motor. This more simple design was chosen to our previous self-securing design, as it is less likely to get stuck on parts of the MPS or the conveyor belt. It's also more reliable as the force of the servo is directly applied to the bracket instead of the old design, which used a belt to transmit the force.

2.2 Architecture and Middleware

The software system of the Carologistics robots combines two different middlewares, Fawkes [10] and ROS [19] (specifically, ROS 2 jazzy). This allows us to use both, legacy software components from fawkes, as well as newly developed and standard tools from the ROS2 ecosystem. The overall system, however, is still integrated using Fawkes. However, many of the components, especially related to navigation, are now communicating directly in ROS, without any interaction with fawkes.

The overall software structur for fawkes-sided components is inspired by the three-layer architecture paradigm [2], as shown in Figure 2. It consists of a deliberative layer for high-level reasoning, a reactive execution layer for breaking down high-level commands and monitoring their execution, and a feedback control layer for hardware access and functional components. The communication between single components – implemented as *plugins* – is realized by a hybrid blackboard and messaging approach [10].

In previous years most of the implementation was done within fawkes (including data processing, hardware drivers, low-level routines and high-level reasoning) , with only the navigation being handled by ROS tools.

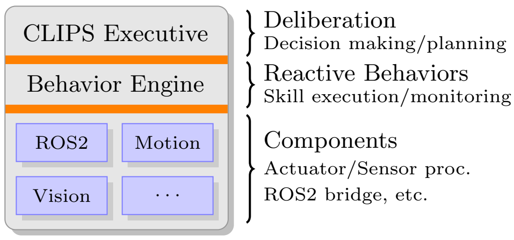

Fig. 2: Behavior Layer Separation [14]

However, over the past years, the team has shifted more and more components to ROS2, as it is largely accepted as the de-facto standard for robotics software. We already attempted to switch to ROS2 in the previous year, however, we were not successful in achieving a full migration. This year we are continuing our efforts to switch to ROS2, with the goal of having all components running in ROS2.

Currently, we are implementing our first high-level agent for ROS2 [27], machine detection via lasers, and a new behaviour engine. Our behavior engine will switch from fawkes hybrid state machines to the ROS2-based hybrid state machines using FlexBE [28,21]. The ROS2 CLIPS Executive has made significant progress and is now able to interact with other ROS2 components in a reliable way. Furthermore, it was extended signficantly with support for temporal planning and a new goal model [8]. We will use these developments to build a new high-level agent for ROS2.

3 Navigation

Our current approach to navigation is focused around the usage of the navigation2 stack of ROS [7]. Towards that, we built a simulation in Webots that should help us advance our current navigation solution by allowing for easier testing.

We decided to work towards a multi-agent path finding (MAPF) solution back in 2022 [25], but could not fully commit to it as of yet. However, by the end of 2023 we had a first working prototype and we are still advancing towards that direction. This year we will do trials regarding the stability of our network setup that we improved last year through the usage of an Ubiquiti LR access point and dedicated PCI wifi cards on the robots. A master thesis was done in 2024 which pointed to some issues for multi-agent navigation which is the synchronization of transforms between the agents, the resulting path following, and inclusion of new goals while a MAPF system is already executing. We worked on the path following and solved the inclusion by setting previous paths as given without reconsidering them. We handle the transform synchronization by using Zenoh for DDS in ROS 2. Therefore, we will test our current MAPF solution extensively.

4 Perception

Every production step in the RCLL comes down to a pick and place task on or from a narrow conveyor belt that is only a few millimeters wider than the workpiece itself. Producing a medium-high complexity product can already involve 18 pick or place operations. Since a single manipulation error is likely to result in total loss of the product, reliability (and therefore precision) is of paramount importance. The 2022 season saw the successful integration of a RGB-based neural region of interest (ROI) detector based on the YOLO [20] framework. The ROIs are used to calculate relative positions in a closed-loop position-based visual servoing (PBVS) task [23].

Integrating this new approach led to a significant increase in reliability, speed, and accuracy over the old approach used in prior seasons. However, more fine-tuning is needed to handle machine inaccuracies and minor inconsistencies in the gripper construction between the robots.

A reviewer of our PBVS approach [23] pointed out that our approach could be understood as a visual angles approach instead of triangulation and that our approach introduces an error in our workpiece detection accuracy. Therefore, we changed our computations to visual angles, fixed the issue and simplified the macenski2020marathon behind it.

In 2025, we freeze our advances of the neural-based vision to detect machines on the shopfloor as a practical integration of the markerless-detection chalange to focus our work on a full ROS 2 integration.

5 Behavior Engine and High-Level Reasoning

In the following we describe the reactive and deliberative layers of the behavior components. Currently, the Lua-based behavior engine provides a set of skills in the reactive layer. Those skills implement simple actions for the deliberative layer, which is realized by an agent based on the CLIPS Executive (CX) [13], a goal reasoning framework that supports multi-agent coordination. We are developing our behaviors in the behavior engine FlexBE [28,21] to move towards a full integration in ROS 2.

5.1 FlexBE-based Behavior Engine

In previous work we have developed the Lua-based Behavior Engine (BE) [11]. It served as the reactive layer to interface between the low- and high-level systems and was based on hybrid state machines (HSM). Currently, we are working on using FlexBE [28,21] as BE instead of our Lua-based BE.

FlexBE is a hierarchical state machine allowing to reuse states, which are mostly wrapper around ROS 2 actions clients executing actions like navigating to a goal, controlling the base to a given pose, and moving the gripper, and whole behaviors within behaviors. It provides a UI to easily build new behaviors by connecting state in- and outputs, changing parameters, and provides a large collection of helper and example states as well as example behaviors. It can be used as an action client which will be used to get called by the ventral agent to execute goals.

Our seperation between central agent and behaviour engine responsability will change to give more control to the behavior engine to only reason about the up-coming goals and handling errors instead of calling multiple behaviors in a row to fulfill a goal and handling all errors on central agent level. New behaviors include the whole process to fullfil a goal from driving to the machine, grasping, and workpiece checking. All execution errors will be handled in the behavior and if something cannot be resolved, the behavior is failed and the errors is further propagated to the central agent.

5.2 Reasoning and Planning with the CLIPS Executive

In previous seasons, we implemented a centralized agent based on the CLIPS Executive (CX) [13], which uses a goal reasoning model [1]. In our latest iteration, a goal represents a task that can be accomplished by one or multiple actors (either robots or software agents) using a sequence of actions. The program flow is determined by the *goal mode*, which describes the current progress of the goal. The mode transitions are determined by the goal lifecycle, which is depicted in

Figure 3. When a goal is created, it is first *formulated*, merely meaning that it may be relevant to consider. The goal reasoner may decide to *select* a goal, which is then *expanded* into one or multiple plans, either by using manually specified plans or automatic planners such as PDDL planners [12]. The reasoner then *commits* to one of those plans, which is *dispatched*, typically by executing a skill of the behavior engine. Eventually, the goal is *finished* and the outcome is *evaluated* to determine the success of the goal.

5.3 Centralised Goal Reasoning

Since 2022 [25] we switched from a distributed reasoning approach (i.e. one where each robot makes their own decisions based on a on a common knowledge base) to a centralized agent (i.e. one where all robots are controlled by a central instance).

We represent all steps that are required to build a product in one goal tree, with each sub-task being a goal consisting of multiple actions. The agent chooses to pursue new orders, based on a set of high-level heuristics considering the feasibility of the order, machine workloads, time constraints, and the current state of the shopfloor.

If an order is selected, all goals associated with the order are added to a new goal tree for this order. The agent then selects the goal from the order goal tree, that has the highest priority, and is executable according to the current world state. Executability is represented in terms of PDDL preconditions. Additionally, a set of support goals is created that provide the agent with the ability to perform order-agnostic tasks like buffering caps, performing payments at ring stations, or discarding unusable workpeices.

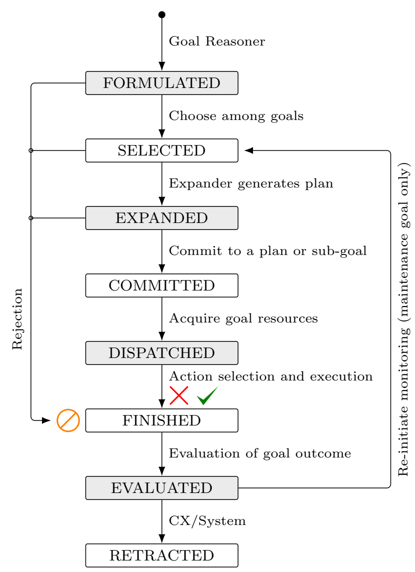

Fig. 3: The goal lifecycle with all possible goal modes [13].

An execution monitoring system is used to track the progress of goals and actions and handles unexpected outcomes (e.g., robots losing workpieces, robots disconnecting, or machine failures).

5.4 ROS2 CX Agent

In 2025, we are working on a new high-level agent for ROS2. This agent will be built on the ROS2 CLIPS Executive and use a new goal and action model.

The agent will be built based on a temporal planning concept, which allows us to actively consider the sequencing and time constraints present in the RCLL for the first time. We will use planning to check for feasibilty of the chosen orders given the current WM. We then de-serialise the plan into sets of equivalent actions.

In our new model, each goal is equal to pursuing an order, and actions are on a more macroscopic level, closely resembling what we previously viewed as a goal. This allows us to give more fine-grained control over the execution to the execution layer, while still allowing the agent to make high-level decisions.

6 Conclusion

In 2025, we focus on our ROS 2 integration and continue our work on the central agent, neural-based perception, and multi-agent navigation. We focus on modernizing our software stack by heavily leaning into the ROS 2 infrastructure and replacing legacy fawkes component by ROS-based counterparts.

The goal of this season is to switch all components that are still running in fawkes to ROS 2. Additionally, the redesign of the arm needs to be put into practice and its grasping robustness needs to increase.

The website of the Carologistics RoboCup Team with further information and media can be found at `https://www.carologistics.org`.

Acknowledgements. We gratefully acknowledge the financial support of RWTH Aachen University and FH Aachen University of Applied Sciences.
This work is partially supported by the German Research Foundation (DFG) under grants *GL-747/23-1*, *2236/1* This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 2236/1 and Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. 390621612.

References

1. Aha, D.W.: Goal Reasoning: Foundations, Emerging Applications, and Prospects. AI Magazine **39**(2) (Jul 2018)
2. Gat, E.: Three-layer architectures. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) Artificial Intelligence and Mobile Robots, pp. 195–210. MIT Press (1998)
3. Hofmann, T., Limpert, N., Mataré, V., Ferrein, A., Lakemeyer, G.: Winning the RoboCup Logistics League with Fast Navigation, Precise Manipulation, and Robust Goal Reasoning. In: RoboCup 2019: Robot World Cup XXIII. pp. 504–516. Springer International Publishing (2019)
4. Hofmann, T., Mataré, V., Neumann, T., Schönitz, S., Henke, C., Limpert, N., Niemueller, T., Ferrein, A., Jeschke, S., Lakemeyer, G.: Enhancing software and hardware reliability for a successful participation in the RoboCup Logistics League 2017. In: RoboCup 2017: Robot World Cup XXI. pp. 486–497. Springer International Publishing (2018)
5. Karras, U., Pensky, D., Rojas, O.: Mobile Robotics in Education and Research of Logistics. In: IROS 2011 – Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (2011)
6. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: RoboCup: The Robot World Cup Initiative. In: Proc. 1st Int. Conf. on Autonomous Agents (1997)

7. Macenski, S., Martín, F., White, R., Clavero, J.G.: The marathon 2: A navigation system. arXiv preprint arXiv:2003.00368 (2020)
8. Micheli, A., Bit-Monnot, A., Röger, G., Scala, E., Valentini, A., Framba, L., Rovetta, A., Trapasso, A., Bonassi, L., Gerevini, A.E., Iocchi, L., Ingrand, F., Köckemann, U., Patrizi, F., Saetti, A., Serina, I., Stock, S.: Unified planning: Modeling, manipulating and solving ai planning problems in python. SoftwareX **29**, 102012 (2025). https://doi.org/https://doi.org/10.1016/j.softx.2024.102012, `https://www.sciencedirect.com/science/article/pii/S2352711024003820`
9. Niemueller, T., Ewert, D., Reuter, S., Ferrein, A., Jeschke, S., Lakemeyer, G.: RoboCup Logistics League Sponsored by Festo: A Competitive Factory Automation Benchmark. In: RoboCup Symposium 2013 (2013)
10. Niemueller, T., Ferrein, A., Beck, D., Lakemeyer, G.: Design Principles of the Component-Based Robot Software Framework Fawkes. In: Int. Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR) (2010)
11. Niemueller, T., Ferrein, A., Lakemeyer, G.: A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao. In: RoboCup Symposium 2009 (2009)
12. Niemueller, T., Hofmann, T., Lakemeyer, G.: CLIPS-based execution for PDDL planners. In: ICAPS Workshop on Integrated Planning, Acting and Execution (IntEx) (2018)
13. Niemueller, T., Hofmann, T., Lakemeyer, G.: Goal reasoning in the CLIPS Executive for integrated planning and execution. In: Proceedings of the 29th International Conference on Planning and Scheduling (ICAPS) (2019)
14. Niemueller, T., Lakemeyer, G., Ferrein, A.: Incremental Task-level Reasoning in a Competitive Factory Automation Scenario. In: Proc. of AAAI Spring Symposium 2013 - Designing Intelligent Robots: Reintegrating AI (2013)
15. Niemueller, T., Neumann, T., Henke, C., Schönitz, S., Reuter, S., Ferrein, A., Jeschke, S., Lakemeyer, G.: Improvements for a Robust Production in the RoboCup Logistics League 2016. In: RoboCup Symposium – Champion Teams Track (2016)
16. Niemueller, T., Reuter, S., Ewert, D., Ferrein, A., Jeschke, S., Lakemeyer, G.: Decisive Factors for the Success of the Carologistics RoboCup Team in the RoboCup Logistics League 2014. In: RoboCup Symposium – Champion Teams Track (2014)
17. Niemueller, T., Reuter, S., Ewert, D., Ferrein, A., Jeschke, S., Lakemeyer, G.: The Carologistics Approach to Cope with the Increased Complexity and New Challenges of the RoboCup Logistics League 2015. In: RoboCup 2015: Robot World Cup XIX (2015)
18. Niemueller, T., Zug, S., Schneider, S., Karras, U.: Knowledge-Based Instrumentation and Control for Competitive Industry-Inspired Robotic Domains. KI - Künstliche Intelligenz **30**(3), 289–299 (2016)
19. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: ICRA Workshop on Open Source Software (2009)
20. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
21. Schillinger, P., Kohlbrecher, S., von Stryk, O.: Human-robot collaborative high-level control with application to rescue robotics. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 2796–2802 (2016). https://doi.org/10.1109/ICRA.2016.7487442
22. Swoboda, D., Viehmann, T., Tschesche, M., Ferrein, A., Lakemeyer, G.: The Carologistics RoboCup Logistics Team 2024. Tech. rep., RWTH Aachen University and FH Aachen University of Applied Sciences (2024)

23. Tschesche, M., Hofmann, T., Ferrein, A., Lakemeyer, G.: Using off-the-shelf deep neural networks for position-based visual servoing. In: RoboCup 2024: Robot World Cup XXVII (2024), `https://kbsg.rwth-aachen.de/papers/robocup2024-pbvs.pdf`, to appear
24. Viehmann, T., Eltester, S., Hofmann, T., Limpert, N., Mataré, V., Ferrein, A., Lakemeyer, G.: The Carologistics RoboCup Logistics Team 2021. Tech. rep., RWTH Aachen University and FH Aachen University of Applied Sciences (2021)
25. Viehmann, T., Limpert, N., Ferrein, A., Lakemeyer, G.: The Carologistics RoboCup Logistics Team 2022. Tech. rep., RWTH Aachen University and FH Aachen University of Applied Sciences (2022)
26. Viehmann, T., Limpert, N., Hofmann, T., Henning, M., Ferrein, A., Lakemeyer, G.: Winning the robocup logistics league with visual servoing and centralized goal reasoning. In: Eguchi, A., Lau, N., Paetzel-Prüsmann, M., Wanichanon, T. (eds.) RoboCup 2022. pp. 300–312. Springer International Publishing, Cham (2023)
27. Viehmann, T., Swoboda, D., Lakemeyer, G.: Towards an integration of clips-based reasoning in the ros 2 ecosystem. 1st German Robotics Conference (GRC) (2025)
28. Zutell, J.M., Conner, D.C., Schillinger, P.: Ros 2-based flexible behavior engine for flexible navigation. In: SoutheastCon 2022. pp. 674–681 (2022). https://doi.org/10.1109/SoutheastCon48659.2022.9764047