DOR: 20.1001.1.27170314.2022.11.2.3.1

Research Paper

Investigation of a Fully-automated Manufacturing Environment Realized through a Flexible Logistic System

Jan-Hinrich Kämper[1*], Arne Stasch[2]
[1]Department of Computing Science, Carl von Ossietzky University, Oldenburg, Germany
[2]OFFIS Institute for Information Technology Oldenburg, Oldenburg, Germany
[*]Email of Corresponding Author: jan-hinrich.kaemper@uni-oldenburg.de
Received: June 10, 2022; Accepted: August 1, 2022

Abstract
Modern industrial production and transportation systems require agile material handling systems. Changing requirements demand flexibility in transportation layout and control systems that are adaptable to the actual needs. Decentralized control systems promise complexity reduction and dependability while centralized approaches improve planning and allow general optimization. By using the use case of a cross-docking logistic system the paper describes how this approach can be combined and implemented by using an agent-based control system on a decentralized lightweight microcontroller infrastructure. The scenario applies to manufacturing environments with modularized assembly lines. These modules are connected by different transport units which merge into an intelligent infrastructure including AGVs, conveyors, and storage.

Keyword
MFS, Flexible Logistic System, AGV, In-house Transportation

1. Introduction

Highly dynamic markets in a globalized world and the increased demand for individualized products make flexible concepts for in-house logistics indispensable in modern manufacturing environments. Companies need to optimize their processes to sustain their position in very competitive markets with high volatility in supply and demand. Optimization in this context not only relates to costs but also and more importantly to “soft factors” such as reliability, agility, and transparency.

Traditional manufacturing systems with a strict sequential order of assembly lines and machines tend to impose a very inflexible structure onto the underlying logistical processes. Often manufacturing systems are designed and the planning of the corresponding material flow infrastructure including conveyors, cranes, belts, etc. is tailored to the needs of the manufacturing system. The big downside of this approach is that neither the manufacturing system itself nor the logistical system can be modified without modifying the other. This is not dramatic for minor changes, but as soon as essential parts of the manufacturing system are subject to updates or modifications the logistical system has to be adapted appropriately. Scalability is another crucial criterion that cannot be satisfied with the traditional approach, e.g. suppose that the throughput of a production site has to be increased by a certain factor then the entire logistical chain must be re-engineered as well. Also, traditional systems suffer from a lack of reactivity and efficiency. Sudden deviations in demand and supply cannot be accounted for on a short-term basis (time horizon of minutes or hours) by accelerating or reducing the performance of transport as it is hardly possible to turn off some of the conveyor belts or transport hubs without affecting the entire system due to connections and dependencies between the components. In the worst case, each unit of transport can be a single point of failure disrupting the whole network when it breaks down. Hence, the aim here is to rethink the idea of in-house logistics as an adequate service for the underlying manufacturing environment instead of being an inseparable part of the system. This approach shall support three high-level quality goals of agile manufacturing:

1. Flexibility to external influence factors on a short-term basis, such as demand and supply, energy prices, security threads, quality issues, etc.
2. Robustness and high availability by proactive behavior of decentralized organizational structures in very dynamic environments
3. Enabling high performance in mass-customization through a highly adaptable logistical setup

The approach described in this paper is based on decentralized control of all mobile and stationary transport entities in an agile manufacturing system (compare [1]). All these entities work autonomously to accomplish their primary goal, i.e. the completion of transportation jobs of the right good at the right time to the right place. They interact to organize the material flow between each other and communicate via a wireless network to this end. Properties of all entities can quite accurately be captured by the Agent metaphor [2]. Therefore all entities in the system are modeled as modularized agents that manage the material flow. The change of paradigm is thus to go from centralized planning of production processes and transport between production steps to decentralized behavior-driven system control. Typical optimization goals in this context concern energy saving or high overall throughput. To preserve goal-driven optimization a hybrid system that allows reactive balancing between the new decentralized kind of control and the traditional central algorithms is deployed. This is done by introducing dynamic hierarchies in agent societies.

This approach has been realized in a small use case consisting of four AGVs, four intermediate storage buffers, and three stationary conveyors as well as a coupled Multi-Agent Simulation Framework that enables scaling up the size of the system in a virtual environment. The system is a simple cross-docking storage scenario, but when thinking of the intermediate storage buffers as points of production, then the whole scenario represents a complete manufacturing environment.

2. State of the art

The basic principles that the approach proposed here is inspired by are not new. However, the combination of them and especially their application in an autonomous in-house logistics system that fits into arbitrary manufacturing chains with possible tight integration in a smart grid is innovative.

The idea of substituting fixed conveyor lines, manually driven vehicles, or forklifts with a fleet of mobile autonomous transport robots emerged in recent decades as a result of the ongoing automation of industrial systems. The company Kiva Systems develops and deploys such systems for warehouse logistics in e-commerce companies [4]. Their approach focuses on order fulfillment solutions with the self-organization of products and high flexibility of the material handling equipment used. Fraunhofer IML built up an experimental hall in the last years where 80 AGVs manage the transportation of goods from commissioning stations into a storage rack where they can move vertically as well [5]. Their research focus has been on the intelligent swarm behavior of the vehicle fleet. Karis is a system developed by the Karlsruhe Institute of Technology where AGVs can team up autonomously to reach a higher transport capacity [6].

Lewandowski et al. [22] developed a cyber-physical system equipped with RFID to realize a material handling infrastructure where items track and trace themselves through the environment based on modular software agents. However, they use a centralized repository to link the tracking data with the agent layer. Their approach tackles the issues posed by "Industrie 4.0" a German Universities Excellence Initiative that aims to improve automation technologies by integrating physical and IT components in complex manufacturing environments through deep pervasion of resources with software and their interconnection through the internet. The main methods used to create this kind of digital factory are self-optimization, interoperability, cognition, and intelligent support of workers in their increasingly complex work [7].

A multi-agent system (MAS) is an appropriate approach for distributed control. Agents can work independently in decentralized systems. The loss or gain of agents does not affect the other agents' functionality. MAS makes decentralized systems flexible and reliable. Requirements, like real-time operations, industrial standards, and small footprints sort out most of the existing MAS frameworks for real-time microcontroller-based applications. JADE [8] is a very common MAS framework but microcontrollers with little processing power do not run Java. Mobile C uses C/C++ as a programming language and C meets industrial standards [9]. However Mobile C needs a General Purpose Operating System (GPOS) and even the low memory embedded GPOS, e.g. Embedded Linux, needs a larger footprint than 259 KB Flash Memory and this does not account for RAM usage. This is not applicable for low power/processing power microcontrollers which can be used in any environment. One part of this research is to create an Agent Framework that can run on such electronic control units (ECU) and is even applicable to any control unit.

Having multiple agents collaborate in teams is a concept as old as the agent metaphor itself. Different variants exist that implement such a collaborative behavior. Anderson et al investigate the question of whether central or decentralized control is preferable in production systems and inspire their argumentation from nature. Ogston et al. describe a method to cluster agents in teams based on properties and goals [10]. Marcolino et al. discuss the trade-off between diversity and uniformity in designing MAS [11]. One way to mediate between central and decentralized control is holonic manufacturing systems (HMS). HMS was first mentioned by Koestler [12]. Fletcher et al. describe a holonic model for manufacturing environments aimed toward robustness [13]. Also, Gerber et al. used HMS in a logistic scenario [14].

3. Realization of the fully automated logistic system

A decentralized in-house transportation system is a proper use case to evaluate the approaches of distributed problem-solving. The aim is to create a modularized material flow system (MFS, [15]) that uses a MAS as a control method. Dividable parts of the MFS are modules that have their actuators and sensors to interact with the environment (Figure 1). Each module has its ECU and can act on its behalf. They communicate and create a cyber-physical system (CPS) in bulk which acts as an extensive MFS [15]. To preserve the option of applying global optimization methods the MAS is enhanced with the ability to dynamically reconfigure between central and decentralized control. The resulting system thus has a hybrid control structure which is described in detail in the following section.

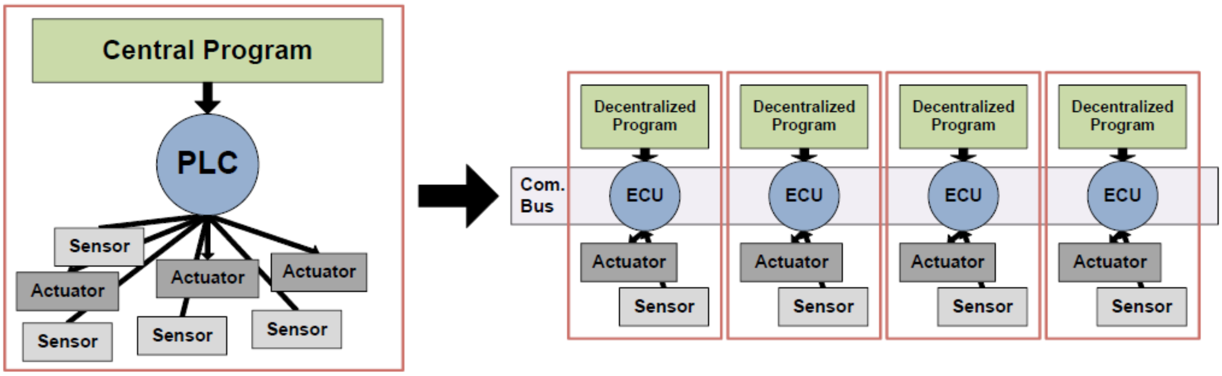

Figure 1. Change from centralized to distributed structure

3.1 Collaboration Strategies

In addition to the modeling of the agent architecture as described in the previous section, a collaboration layer is provided which enables all agents from different conveyors to form larger teams orientated from the principle of HMS [14]. Typically the agents being part of teams are only platform agents representing the entire conveyor that they belong to including the corresponding routing, order, and package agents. So for example a team could consist of two platform agents from AGVs and two platform agents from ramps who team up because they are located close to each other and complement one another in their skills. So they should work on the incoming jobs collaboratively and not each one on its own. Such a team acts in the agent society as an aggregated agent. Aggregated agents also have proactive and behavior-driven control just like single agents.

They also provide the same communication interface towards their environment as single agents.

The crucial difference between a team of agents and a single agent lies in internal control. Whilst a single agent only has to manage itself and consequently, there is no such issue of control, teams of agents must somehow organize their collaborative behavior.

3.1.1 Central vs decentralized control driven by team structure

Internal control in these teams is intentionally designed to be easily adjustable. Unlike in a true MAS, there always exists some kind of hierarchy in a team. Thus the single agents within a team are not as free as they would be without the team. To some degree, they obey the superior control of their team. The exact degree of freedom they have is defined on a dynamic basis. One extreme is a strict hierarchy where one designated managing agent assigns orders to all other agents that only perform actions to complete these assigned orders ("authority"). There is no interaction or proactive behavior at all among the other agents. The managing agent takes the role of a central instance leading the team, so that distributed control no longer exists in that team. The other extreme would be to have a team of entirely autonomous agents that only talk to one very weak central instance when interacting with the environment ("autonomy"). The central instance in that case only represents a communication interface towards the surrounding world.

Depending on the number and size of the teams and the configuration of their internal control the described approach enables a flexible switching between central and decentralized control in the operating mode (in the remainder called “aut-balancing”). This ability is a true asset since it allows for flexibility regarding different external influence factors. The purely decentralized control is preferable when efficient and robust transport processes are required and the demand remains far under the maximum bound of the systems transport capacity. However, when the throughput and speed have to be increased due to peaks in demand or failure of machines it becomes necessary to employ optimization methods that can only be executed with a central instance having a global view over parts of the system. Also, exceptional circumstances such as security threads, error recovery, or obliged audits require a central control to make the system behavior more transparent and be able to detect problems and risks.

3.1.2 Aut-Balancing in the simulation and issues

We realize this approach only in the virtual simulation environment since there is no additional value in doing this in the physical world where the number of entities is too small to investigate the effects of large swarm dynamic behavior. This is still a work in progress so evaluation results cannot yet be presented. There are several issues regarding the realization of aut-balancing which will not be discussed here but are topics for future work:

1. Selecting the managing agents
2. Choosing the internal hierarchy in a team
3. Deciding when to reconfigure the system
4. Supplying enough computing power for managing agents

4. Microcontroller infrastructure for MAS

In this section, the modularization of MFS is addressed. The networked ECUs control their modules and a MAS should be implemented as a decentralized control concept for the collaboration strategies. In previous work [18] the design for an Agent Framework for microcontrollers with low processing power was presented. The concept is refined and the adaptability of the concept will be proven through the use of different platforms. Inspired by the model of a modular SW architecture for ECUs from AUTOSAR [19] the SW should be adaptable to different microcontrollers. The focus of this part of the research is to design an Agent RTE which delivers in- and output abilities and flawless communication between agents. The layered architecture shown in Figure 2 is an example of a MICAz node [17] on a ramp module. Through unitary interfaces, Agent RTE can use different HW for the same functions. Even the RTOS in this example Contiki OS should be replaceable through another. The Agents are embedded in the same environment on every platform.

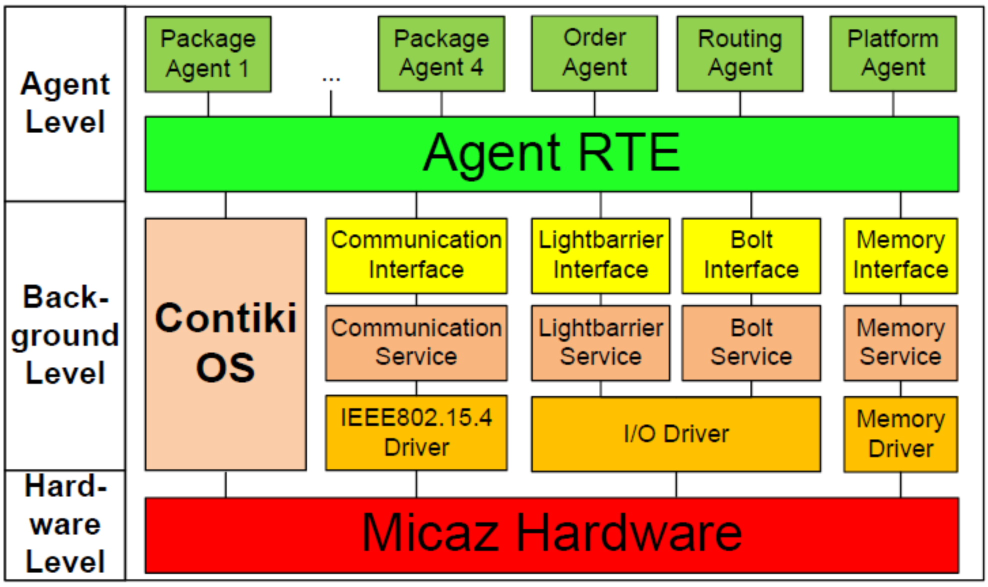

Figure 2. The modular architecture of a ramp

The difficult part is constructing a lightweight RTE. The possibility for an agent to communicate with every other Agent with a reduced set of FIPA messages [20] is a RAM-consuming task. Allowing all agents to use the same external communication channel per platform and the ability to communicate on the same platform, the use of message queues or token methods is inevitable.
Although the Agent RTE sets many boundaries to capabilities and the design of the agents, it makes it easier to add new agents or reconfigure, add or remove modules in appreciation of the MFS view. This enables "traveling" agents like the Package Agents who can "travel" with the items from module to module. This use case of MFS affiliates to real-time safety critical systems which are sensitive against transferred executable code and this is not the object of this research. The
Package Agents will be only transferred via parameters and the Agent RTE activates and deactivates the existing code of the agent.

5. Construction of the overall system

The aim was to create a highly scalable fully autonomous MFS that can benefit from the technologies introduced in the previous sections. This approach to an MFS (Figure 3) includes the transport units: AGVs, conveyors, and storage. The AGVs transport items individually from point to point. The conveyors take over fixed high-traffic routes. The storage is represented by autonomous ramps with FIFO logic. Due to scalability, costs, and workload, a realistically scaled plant has not been realized. On an area of 50 m^2 four AGVs, three short conveyors and four ramps are representing a realistic extract of the plant. The virtual simulation of the whole plant works in a hybrid mode to interact with the physical parts. It allows single-mode tests of new algorithms before they are used in the real environment. The systems design allows creating any layout combined with the modules to adapt to necessary structures. A modularized infrastructure consisting of different resources assures the cooperation of the modules and is described in the next section.

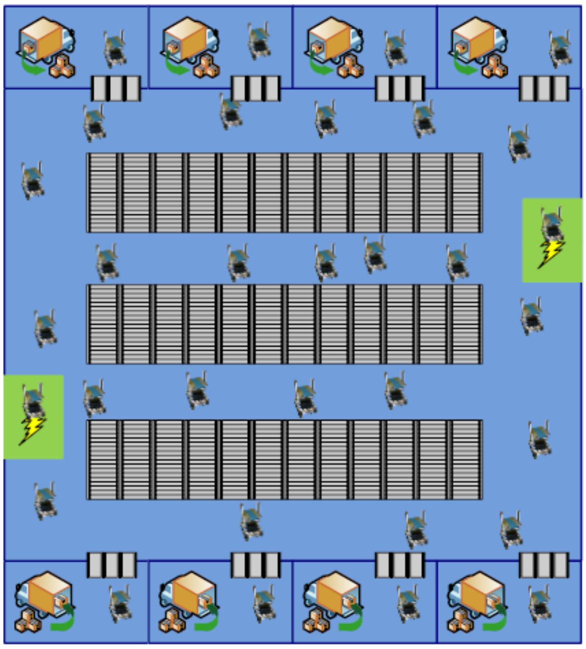

Figure 3. Draft of an entire MFS site

5.1 Infrastructure

The infrastructure of the realized parts of the system is designed in a modular setup [3]. The described module types (Figure 4) are physically very different but they all have the same vertical system layout [15]. Every module has its actuators, sensors, and an ECU that carries a multi-agent platform described in section 4.

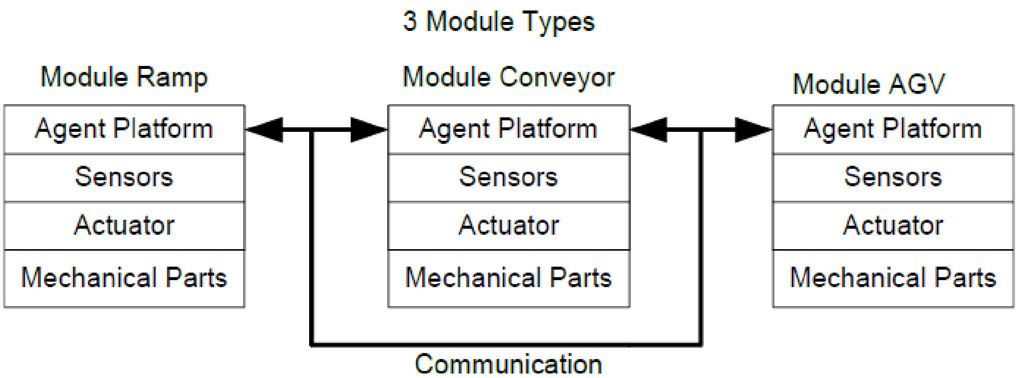

Figure 4. Module Types

The ECU controls the module and communicates with other modules (Figure 5). The AGVs and ramps are equipped with MICAz nodes [17] which are based on the ATMEL ATmega128 chip and use wireless IEEE 802.15.4 radio standard. They are often used as sensor nodes in distributed measurement applications. The "STASH" (Shortened and Transformed Agent System Handler) controllers are custom ECUs with Texas Instruments MSP430 low-power microcontrollers. They communicate over the Fieldbus standard Profibus. A Gateway between those two communication standards is necessary. The gateway is realized through a MICAz node with a Profibus adapter. In the remainder of this section, the three module types will be explained in more detail.

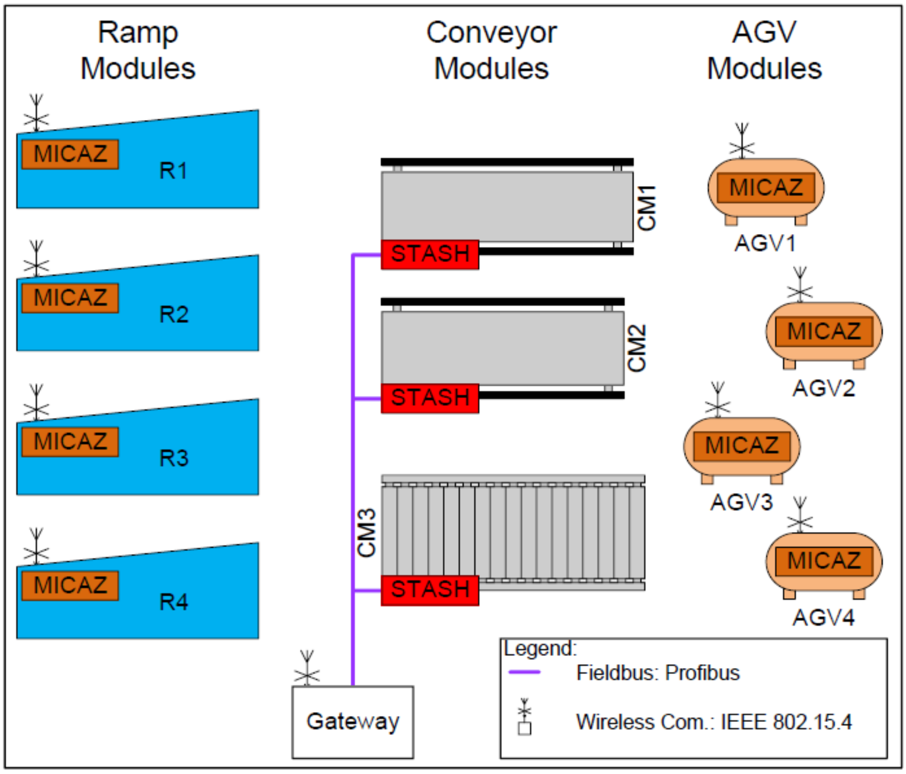

Figure 5. Infrastructure with ECUs and communication types

5.2 AGVs

The AGVs are VolksBots from the Fraunhofer IAIS with a custom lifting platform (Figure 6). The bot has three wheels on each side which are coupled with a belt. They allow a driving speed of about 2 m/s and turn on the spot by rotating the right and left sides in opposite directions. The lifting platform can take in and take out one item with a motorized belt. They can also lift the item to the height of the interacting module with a motor-driven arbor. Two light barriers are monitoring the positions of the items.

Figure 6. Volksbot with Lifting Platform

Laser scanners in front and back ensure positioning and coordination with other AGVs. Analyzing laser scanner data and path planning needs more processing power than a small low-power controller

could provide. Subnotebooks provide the power and are connected to MICAz nodes. The MICAz node assigns higher commands and the subnotebook acts in lower levels of execution.
Algorithms used for pathfinding are the A* search algorithm from Hart et al. [22] and Lee's algorithm [23] depending on the traffic situation and current goals of the AGVs (high speed, energy saving, etc.). The software architecture of the AGVs is layered in the following Order: On the lowest layer, the hardware (sensors and actuators) are controlled. The operational layer above contains all important algorithms for analyzing data and driving as well as decision-making (routing, collision detection, positioning, etc). The layer on top ensures the connection to the MFS infrastructure via a MICAz gateway. The infrastructure indicates the destinations and handles the MAS communication and representation in the module network.

5.3 Conveyors

The conveyors are short roll and belt conveyors. Through the historical use of these conveyors and the "STASH" controller, everything works on Profibus (Figure 7). The ECUs are Profibus Slaves and cannot communicate with other electronics in the Profibus network than a programmable logic controller (PLC) as Profibus master. The PLC remains in the system as a Profibus network switch to transfer messages (ECU to ECU), actuator commands (ECU to frequency converter), and light barrier state (light barrier to ECU). The higher logic and decision making is realized in the ECU or the ECUs can be substituted by the PLC to evaluate decentralized system control versus central control.

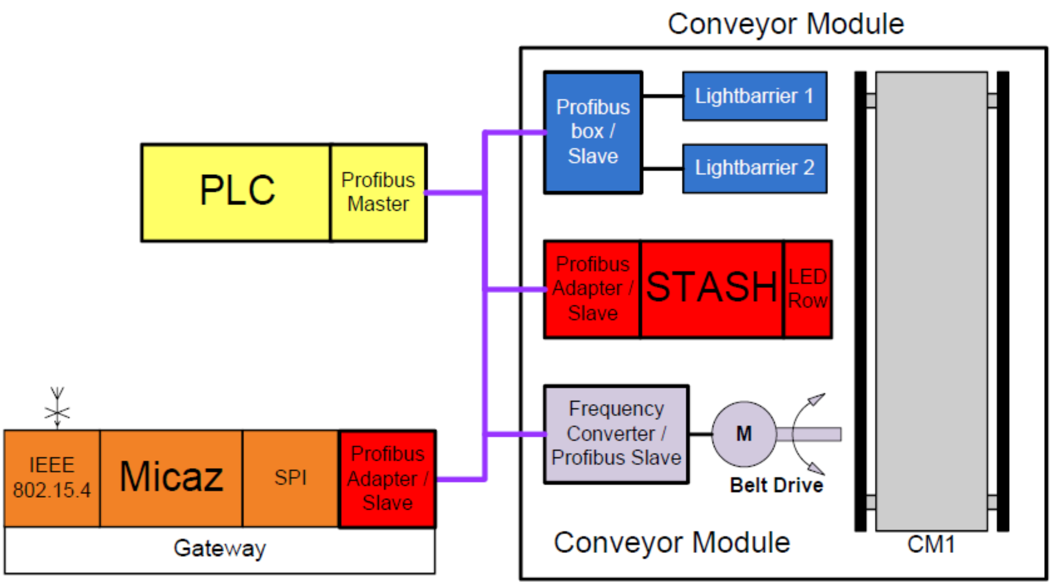

Figure 7. A conveyor module structure with necessary

5.4 Ramps

Storing items on a self-controlled shelf with only a minimum of electronics that allow easy docking to AGVs was the requirement for the storage. The ramps were custom-made to correlate with these requirements. The items which are loaded on the higher side of the ramp roll down till magnetic bolts stop them (Figure 8). One ramp can store four items at the same time and checks the slots with light barriers. They can come and go only in FIFO order. The magnetic bolts allow isolating the item on the low outgoing side. So the receiving AGV can take the item easily.

The reason for this storage design is to be able to store items with a minimum of required energy-consuming actuators. In this case, we only need a lifting platform on the AGV and bolts on the ramps. Most common storage systems employed in warehouses or production environments are less space-consuming but need a lot more mechanical parts and consume more energy to move items, e.g. forklifts, automated racks, cranes, etc.

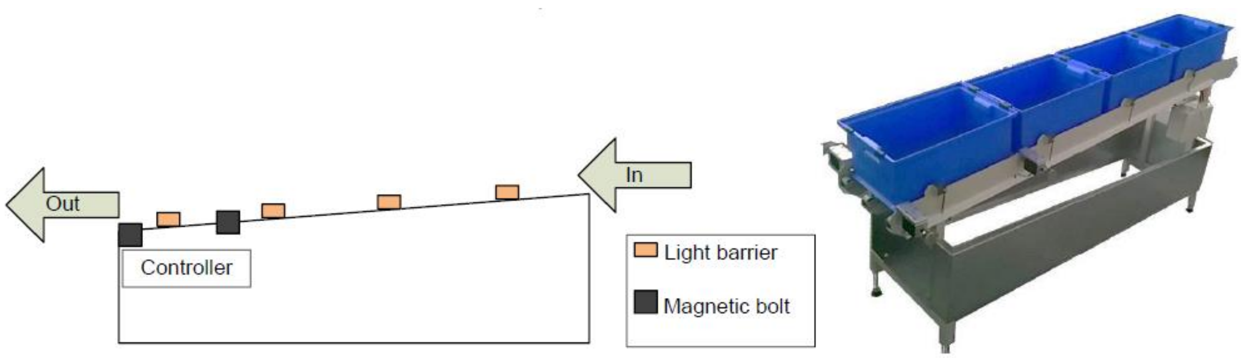

Figure 8. Draft of ramp setup (left) picture of a ramp (right)

5.5 Coupled Simulation

The eleven modules of this MFS setup are only a small part of a realistic plant environment. A simulation of an entire plant is developed to scale up the MFS to a size of a realistic scenario. In this scenario, the simulations work as a test bench for different strategies and algorithms for the swarm of entities, module positioning, and layout planning. The simulation enables tackling research questions, e.g. concerning the optimal amount of vehicles in a certain layout to achieve a predefined throughput or the best strategies regarding team formation. Besides the technical aspects and the roll-out of the architecture on a real physical infrastructure, the virtual simulation can go beyond and allows examining agent behavior in larger swarms. In a hybrid mode, the simulation works in real-time and synchronizes with the physical entities by sending and receiving messages from the physical infrastructure. This allows us to test the simulated results and to get a realistic evaluation of tested designs etc. The coupled simulation is implemented with the
JADE agent framework.

6. Conclusion and Future Work

The work described in this paper shows how a flexible in-house logistic system for agile manufacturing environments can be created using a lightweight multiagent framework for microcontrollers running on heterogeneous material flow components. A system designed in this way can easily be fitted to arbitrary manufacturing environments. The modularity of the agents enables fast transfer to different platforms and simplifies information exchange between the various components via a wireless network. Items define their route through the MFS and are thus able to react quickly to sudden changes. Failures of single entities can be compensated easily by substituting the affected hardware and creating a new agent. The same holds for adaptivity for the number of entities active in the system which can be increased or decreased with no additional overhead other than registration. This approach has been implemented in a physical system consisting of 11 entities and will be connected to a virtual simulation allowing macroscopic investigations of team behavior and different strategies in the future. The goal is to find solutions for an agile balancing between decentralized control driven by distributed single agent behavior and central control performed by clustered agents running optimization methods and exact algorithms for decision-making on a global level. The system is designed to reconfigure itself in this respect depending on external influence factors like availability of goods, demand situation, energy price volatility, security risks, and management decisions. The next steps in further developing the system will be to model these external factors in the coupled simulation and to finally realize the coupling of the agents running on the physical entities and the JADE agents from the simulation. Then it will be possible to evaluate the efficiency of the whole concept and compare it with the performance of traditional material handling systems using centralized control and fixed conveyor lines or even manually operated vehicles.

7. References

[1] Gunasekaran, A. 2001. Agile Manufacturing: The 21st Century Competitive Strategy ultiagent systems: a modern approach to distribute artificial intelligence. Elsevier.

[2] Weiss, G. 1999. Multiagent systems: a modern approach to distribute artificial intelligence. MIT press.

[3] Günthner, W. 2010. Internet der Dinge in der Intralogistik. Springer, Heidelberg.

[4] D'Andrea, R. 2012. Guest editorial: A revolution in the warehouse: A retrospective on KIVA systems and the grand challenges ahead. IEEE Transactions on Automation Science and Engineering. 9(4):638-639.

[5] Kamagaew, A. 2011. Concept of cellular transport systems in facility logistics. 5th International Conference on Automation, Robotics and Applications (ICARA), IEEE.

[6] Seibold, Z. 2013. Layout-optimized sorting of goods with decentralized controlled conveying modules. Systems Conference (SysCon), IEEE.

[7] Horbach, S., Ackermann, J. Müller, E. and Schütze, J. Building blocks for adaptable factory systems. Robotics and Computer-Integrated Manufacturing. 27(4): 735-740. 2011.

[8] Bellifemine, F., Poggi, A. and Rimassa, G. 1999. JADE–A FIPA-compliant agent framework. Proceedings of The Practical Application of intelligent Agents and Multi-Agent Technology. 99: 97-108.

[9] Chou, Y., Ko, D. and Cheng, H. 2010. An embeddable mobile agent platform supporting runtime code mobility, interaction and coordination of mobile agents and host systems. Information and Software Technology. 52(2):185-196.

[10] Ogston, E., Overeinder, B. Van Steen, M. and Brazier, F. 2003. A method for decentralized clustering in large multi-agent systems. Proceedings of the second international joint conference on Autonomous agents and multiagent systems.

[11] Marcolino, L. 2013. Diversity beats strength? Towards forming a powerful team. 15th International Workshop on Coordination, Organizations, Institutions and Norms (COIN).

[12] Koestler, A. 1972. The Rules of the Game: Beyond Atomism and Holism–The concept of the holon 1. Taylor & Francis.

[13] Fletcher, M., and Deen, M. 2001. Fault-tolerant holonic manufacturing systems. Concurrency and computation: practice and experience, 13(1):43-70.

[14] Rodriguez, S., Hilaire,V., Gaud, N., Galland, S. and Koukam, A. 2011. Holonic multi-agent systems. Self-organising Software. Springer, Heidelberg.

[15] Ten Hompel, M., Nettstraeter, A., Feldhorst, S. and Schier, A. 2011. Engineering of Modular Material Flow Systems in the Internet of Things. Automatisierungstechnik. 59(4):248-256.

[16] Lee, E. 2008. Cyber Physical Systems: Design Challenges. 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC).

[17] Hill, J.L. and Culler, D.E. Mica: 2002. A wireless platform for deeply embedded networks. IEEE Micro, 22(6): 12 - 24.

[18] Stasch, A. and Hahn, A. 2013. Towards a Multi-Agent Platform for Cyber-physical Systems based on Low-power Microcontroller for Automated Intralogistics. Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics.

[19] Heinecke, H. Schnelle, K., Fennel, H. and Bortolazzi, J. 2004. AUTomotive Open System ARchitecture-an industry-wide initiative to manage the complexity of emerging automotive E/E-architectures. Convergence International Congress & Exposition on Transportation Electronics, Detroit, Michigan, United States.

[20] Nicol, R.C. and O'Brien, P.D. 1998. FIPA-towards a standard for software agents. BT Technology Journal. 16:51–59.

[21] Lewandowski, M. 2013. Agent-based Control for Material Handling Systems in In-House Logistics. European Conference on Smart Objects, Systems and Technologies.

[22] Hart, P. E., Nilsson, N. J. and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics. 4(2): 100–107.

[23] Lee, C. Y. 1961. An Algorithm for Path Connections and Its Applications. IRE Transactions on Electronic Computers, EC-10 (2): 346–365.