Efficient messaging through cluster coordinators in decentralized controlled material flow systems

Christian Lieberoth-Leden[1], Daniel Regulin[2] and W. A. Günthner[1]

[1]Technical University of Munich, Institute for Materials Handling, Material Flow and Logistics, Munich, Germany
[2]Technical University of Munich, Institute of Automation and Information Systems, Munich, Germany

Abstract. The modularization of the hard- and software is one approach handling the demand for increasing flexibility and changeability of automated material flow systems. A control that is distributed across several different hardware controllers leads to a great demand for coordination between the modules while planning for example transports, especially if there is a mutual dependency between the modules on the executing tasks. Short-term changes in planning often initiate a rescheduling chain reaction, which causes a high communication load in the system. In the presented approach, module clusters with a centralized coordinator are automatically formed out of multiple modules and substitutional take over the surrounding communication for the modules. As a result, they minimize exchanged messages by focusing on the essential information.

1 Introduction

Present day material flow systems are mostly operated by an individual central control. Flexibility is only facilitated within predefined limits. This is for example the case when a designated alternative route is temporarily used for a higher throughput. New demands on an existing material flow system which require flexibility not originally considered (e.g., an extension of the system) require a manual adaption of the individual control. The manual adjustment of an individual material flow control is performed by skilled employees who know the system functionality and have competencies in programming. A material flow system that provides flexibility beyond predefined limits is characterized as being convertible. Convertible material flow systems can be realized through reusable Plug-and-Play modules that configure autonomously the material flow control. Plug-and-Play material flow systems are able to be modified instantaneously to match the current demands and reduce setup costs, particularly for systems with a high demand for flexibility [1].

In this paper, we define a module as an encapsulated unit that performs predefined functions, such as conveying, buffering or identifying a transport unit (TU). A module possesses all the necessary software functions to control its hardware, to communicate with other modules or superior systems and for the purpose of autonomous self-configuration. Standardized software interfaces and property descriptions enable collaboration between heterogeneous modules, for example the combination of a conveyor with a crane. Consequently, autonomous self-controlled modules enable convertible material flow systems that can be changed during runtime.

In a convertible system that is changed during runtime not every possible material flow situation can be simulated beforehand. Flaws in the material flow like deadlocks must be avoided by planning the execution of transportation in advance. Usually, not all the transport assignments for a specific time slot and module are known in advance, as TUs can, for example, be manually placed in the system. Consequently, TUs are added to the system without a specific sequence, however execution of their transportation is planned instantly. Therefor for each transport specific time slots are exclusively reserved at the modules of the TU path. Later, when new transports are added, they could cause an inefficient planning scenario for a module.

Modules that share resources like a rail, are examples that depend on each other. Only if all the module transports for a specific time slot are initially known, can the utilization of the common resources be optimally planned. Since this is not the case here, the planned schedule must be changed afterwards to achieve optimization. But if the module schedule has to be changed, the changes with all the other modules can potentially lead to a chain reaction with a significant increase in communication traffic.

The communication capacity and computational power of a modular material flow system are technically limited. Therefore, the number of modules in a system is limited, as well as the minimum latency. Regarding a material flow system consisting of self-controlled modules, we aim for a control that plans transports in advance by reserving modules for a designated time slot

---

in a communication efficient manner, even for modules which depend on each other.

To implement a communication efficient control, we can state the following requirements: in particular, we want to reduce the communication load for rearranging existing reservations; by means of frequent and easy rescheduling of firmed reservations we want to increase the throughput; additionally, the new control concept should not cause more traffic for regular reservations, even though they are never rescheduled; the existing individual module control for processing reservation requests should also not be changed to enable a simple implementation. In conclusion, we can only add control functions for more efficient messaging or change the path of communication between different modules.

A solution for the problems for the above-stated requirements is module clusters. The modules build up clusters with a centralized coordinator, which represents the module features to external modules. The coordinator takes over communication with the system and decreases the amount of outgoing messages by focusing on the essential information with respect to the actual planning status.

After the introduction, we provide an overview of agent systems, agent communication concepts and decentralized routing strategies. Based on the routing strategies, we describe in the following section a decentralized material flow control for convertible systems that avoids deadlocks. Afterwards, we introduce module clusters and describe how they affect the messaging between modules for regular reservations and rescheduling by using the transversal carriage system as an example. Subsequently, we evaluate the cluster concept by calculating the message load analytically and conducting a parameter study. In the last section, we summarize and discuss the utilization of cluster coordinators for decentralized controlled material flow systems.

2 State of the art

Autonomous entities that are able to execute predefined tasks such as the modules mentioned in this paper, can be realized by means of agents that control a module and communicate with other modules or systems. Vogel-Heuser stated that the utilization of such agents enables Cyber-Physical Production Systems for Industrie 4.0 applications [2]. In this paper, we use agents to realize a decentralized controlled material flow system. Therefore, we implement a specific routing and messaging strategy for the module agents.

Systems with several cooperating agents are called multi-agent systems. A multi-agent system might involve tasks that can not be performed by a single agent. In this case, multiple agents form a group in order to solve the task together; this process is called coalition forming. Another reason for a coalition is more efficient task handling, since in a group the agents aim for the absolute optimum [3]. In this paper, agents group together to facilitate task handling but we focus on their messaging strategy.

Hellenschmidt et al. have already stated, with respect to an agent system with distributed service providers, that all relevant information has to be communicated to one entity in order to make an optimal decision. Therefore, they developed a model to build a temporary communication group. The group consists of several entities, in which an entity can gather information and broadcast decisions [4]. The focus of this work is not to reduce the communication load, but to broadcast information to all relevant entities.

The focus of the present paper is the planning of transportation in an automatic material flow system. The process is divided into routing and reservation. For the routing, a path from a start to a destination is determined. Various criteria can be considered such as the transport time or the traffic frequency [5]. Then, the reservation process blocks the selected route for other TUs, so that they do not interfere with the transport. Modules only have local knowledge about the actual position of the TUs in the system. Consequently, the modules require knowledge, when to pass on a TU. This knowledge is generated through the reservation process, where for each TU the entire route is reserved in advance for the physical transport [6]. The reservation of a route leads to a lower system throughput. Other TUs are not allowed to use a reserved path till the designated transport has been completed. Other transports have to wait even though they want to use the same path in the same direction and do not interfere with the reserved transport. In this case, a time slot based reservation algorithm increases the throughput. It is not the entire route which is reserved for the whole transportation period, but only parts of the path. In our case module, for example, it refers to the actual required time for transportation [5]. For decentralized controlled modules the reservation process with its communication has already been described. The difficulty with the communication load for rescheduling module reservations that depend on each other has not yet been considered.

If modules depend on each other, the reservation process has to be coordinated with other dependent modules in order to decide, for example, on the utilization of a common resource. Chisu et al. examined different decentralized control strategies based on their throughput for a system with several transversal carriages (TC) travelling on a single rail. The best results are achieved by reserving for each transport the entire path on the rail for the expected process time [7]. Transport rescheduling is not considered, since the control only reserves the rail for actual TUs waiting to be transported and not for ones in the future.

3 Planning transports in decentralized controlled material flow systems

In this section, we focus on the functionality of a time slot based reservation algorithm in a decentralized controlled material flow system. First, we describe the process of finding the shortest path to a destination called the routing. If multiple TUs are in a system at the same time, the transports can interfere with each other causing

deadlocks. Therefore, we discuss different strategies to handle deadlocks. Determined by the routing, the path time slots at the corresponding modules are reserved in order to avoid deadlocks.

Prior to the physical transport of a TU, the route has to be determined. Since routing is not a core theme in this paper, we assume that TUs always want to travel on the shortest path in terms of time. Only the modules possess knowledge about their workload at any given time, so their data is needed to determine the shortest path. Sending the total knowledge of each module in the system to a TU with a route request leads to a greater communication load. More efficient communication is possible when only the relevant modules for a route request send messages to each other. For executing transportation, a module only needs to know which neighbouring module a TU has to be transferred to. Therefore, each module possesses a table, structured as a matrix, where, for every possible destination module in the system and time, exactly one neighbouring module is listed that provides the shortest transportation time in this situation. If a module's routing table changes, the updated table is sent to all neighbours. The neighbouring modules examine for each destination module whether a shorter path is available and update their routing table with the new neighbouring module and time for a specific relation [6, 8].

When several TUs are transported at the same time in a system, they can block each others' paths, resulting in a deadlock. Strategies for handling deadlocks are prevention, detection and avoidance [9]. In material flow systems deadlocks are prevented if no deadlocks can occur due to the layout. Since the material flow system is convertible, a layout that prevents deadlocks can not be warranted. The detection and recovery of deadlocks requires a lot of communication between the modules to gather all the necessary information. Since we aim for efficient communication, this strategy is not considered any further. In conclusion, we use an avoidance strategy, where we reserve exclusive time slots for a specific TU at the modules, as explained in the following section.

The reservation process is based on the works of Mors and Mayer [5, 6]. Once a TU transport assignment has been received by a module, the reservation process starts. Each module passes the reservation request to the neighbouring module, which is listed in the module's routing table for the shortest path to the destination module. If the next module is the destination module, the reservation process is successful and a confirmation is returned. The confirmation travels the reversed route and the modules confirm the reservation.

When a module receives a reservation request, the entrance and exit time has to be determined by calculating the process time for the TU at the module. For a continuous conveyor, the processing time is the travelled length divided by the continuous speed. The present paper focuses on a TC system. A TC system is an example of several modules that depend on each other. TCs travel on a rail and can reach several entrance and exit modules. The time required to travel to the entrance prior to the transport has to be considered as well for the process time. If more than one TC uses the same rail, the reservations for the rail from the other TCs have to be considered with respect to the process time. The reason is that those reservations may require a waiting time for an open reservation slot. The earliest possible and latest allowed exit time is then passed on with the reservation request.

4 Application areas and communication of module clusters

Module clusters especially qualify for modules, which depend on each other. Here, they can simplify and reduce communication load, particularly if changes in the reservation planning process are required. In the following, we use a TC system to describe module - clusters as shown in Fig. 1. Every TC can reach every entrance and exit. For every possible destination module in the system, the TC with the path with the shortest time is listed in the routing table as neighbouring module for an entrance.
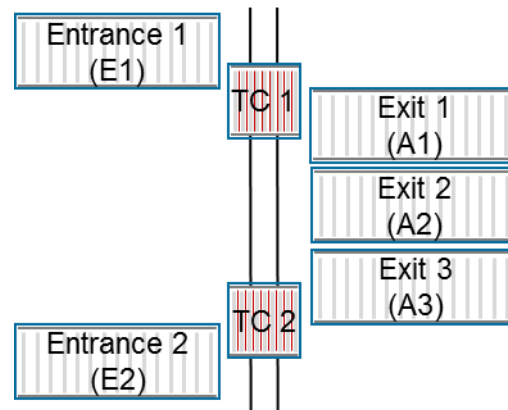

Figure 1. Example of a TC System.

The communication in a decentralized controlled TC system using a time slot reservation algorithm is shown in Fig. 2. When a reservation request is received by an entrance module it is processed and forwarded to the TC as listed in the routing table. The TC also processes and forwards the request to the corresponding exit module as stated in the TC's routing table. After the TC receives the confirmation of the reservation, the required rail segment is firmly reserved for the time slot and the confirmation is passed along to the entrance module.

Each time a TC rail segment reservation for a time slot is confirmed, this information has to be communicated to all other TCs that use the same rail. In conclusion, every TC is aware of the current reservation schedule of the rail. TCs offer transport times to the entrances for every possible entrance exit relation. When a TC offers transportation to an entrance, it calculates the travel time to the entrance, the handling time for loading a TU and the travel time to the exit module. For all those calculations the reservation schedule of the rail has to be taken into account. Each time the reservation schedule is changed, every TC has to recalculate all of their entrance-exit relations to check whether the change affects their offered transports. If transport times for entrance-exit relations change, the new transport times are

communicated to the entrance of this relation. The entrance then decides whether the TC still offers the timely shortest path or has to be replaced by another TC.

Module clusters consist of several modules which correlate in one way or another with each other. In the case of limited common resources, the modules need to follow a conflict management strategy to share the resources. In order to collaborate, the modules need a cooperation strategy to coordinate each module for offering enhanced functions.
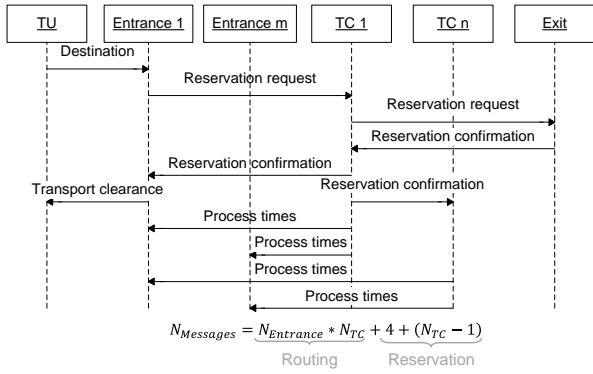

$$N_{Messages} = N_{Entrance} * N_{TC} + \underbrace{4 + (N_{TC} - 1)}$$
$$\quad\quad\quad\quad\text{Routing} \quad\quad\quad\text{Reservation}$$

Figure 2. Communication process for a regular reservation without module clusters.

In this paper, we discuss a cluster whose modules depend on each other. We aim to simplify the communication for routing and reservations between the internal modules of a cluster and with respect to other external modules in the material flow system. A cluster consists of one central coordinator, which is responsible for the internal and external communication. The coordinator represents the cluster in the system and behaves like a regular module. The properties are derived from the sum of the represented modules in the cluster, but the coordinator brings its own control agent along. The control agent, or any other properties of the modules in the cluster, are not modified with the exception that all communications now go through the coordinator. Thus, we fulfil the requirement that no modification in a module's control is required. The communication with a coordinator is shown in Fig. 3. For each entrance module, the coordinator is now listed as neighbouring module for all destination modules.
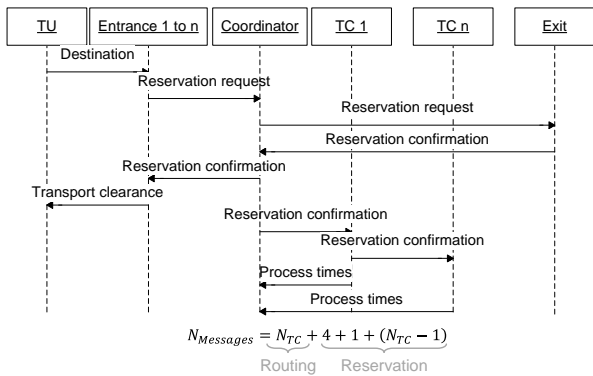

$$N_{Messages} = N_{TC} + \underbrace{4 + 1 + (N_{TC} - 1)}$$
$$\quad\quad\quad\quad\text{Routing} \quad\quad\text{Reservation}$$

Figure 3. Communication process for a regular reservation with a cluster coordinator.

Hence, all the reservation requests for the TCs are processed by the coordinator. The coordinator can determine the transport times that are based on the expected loads and at this point does not have to assign and reserve a specific TC to a transport. The actual assignment of a transport to a specific TC is done as late as possible to minimize the risks of delays that would cause rescheduling the reservations. Shortly before arrival, the coordinator selects the TC from its routing table with the shortest transport time and forwards a reservation request to the corresponding TC. After a confirmed rail segment reservation, every TC updates its transport times for all entrance exit relations and transmits them to the coordinator, which again updates the routing table.

In the following we describe the rescheduling process with the TC system introduced in Fig. 1. Fig. 4 shows the reservation schedule for the rail over time for both TCs with respect to confirmed transport assignments. The shaded area represents a new reservation request that was added at short notice. Unfortunately, the request slightly overlaps a confirmed reservation from TC 1, so it has to be shifted to the next free time slot in the future. However, in terms of system capacity, a more efficient solution is to slightly shift the predecessor transports to an earlier time slot.
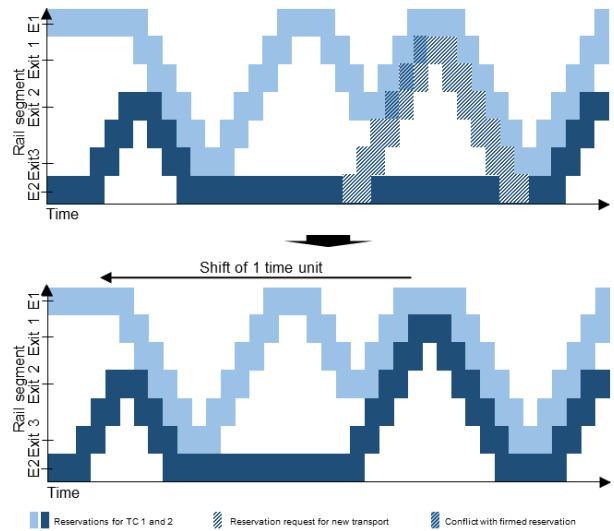

Figure 4. Schedule of confirmed rail segment reservations over time for a TC system and a new reservation request.

In the case of rescheduling, a request has to be sent to all affected modules, as shown in Fig. 5. The notification procedure works as follows: TC 2 requests TC 1 to reschedule one transport. TC 1 detects that, in order to shift the requested transport, two different TC 1 transports and one TC 2 transport have to be moved as well. If the TC reschedules a transport, the corresponding entrance has to be queried whether the transport can start earlier or later. Subsequently a rescheduling request is sent to the affected entrances of TC 2 and TC 1 which then also sends a request to the entrance.

The entrances determine the available buffer for the time slots and inform the requesting modules. The requesting modules also determine the buffer and pass the information along to the module which first sent a

request. In the present case, TC 2 decides whether rescheduling is possible and then sends a rescheduling confirmation to all affected modules. In conclusion, rescheduling of firmed reservations can easily trigger a chain reaction of changes that cause a large amount of message traffic in the system.
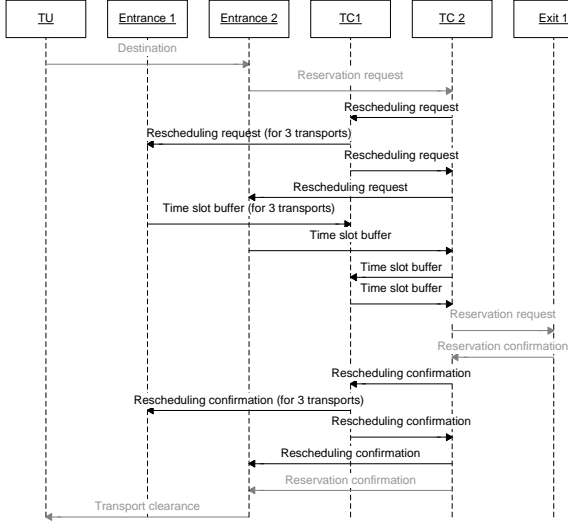

Figure 5. Communication process for rescheduling reservations in the TC system.

An informed coordinator that plans rescheduling and subsequently provides only the affected modules with necessary information is able to reduce the communication load. Since the cluster coordinator assigns transports to a TC just before the arrival of the TU, it only needs to know whether sufficient transport capacity is available. On this basis, new transports are planned and rescheduled with no need to communicate with TCs at this point. The cluster coordinator fulfils the requirement to cut the communication load for rescheduling. Since the coordinator does not use confirmed reservations, but rather estimates, the planning accuracy decreases. For our system, we always expect a high demand for rescheduling, since the modules randomly receive transport assignments. Furthermore, we can now efficiently utilize optimization potential, since transports can easily be rearranged to minimize empty running of TCs. For this purpose, we accept a reduced planning accuracy, since the probability that transportation is conducted as initially planned is minor. However, we still use confirmed reservations for the physical transport to avoid collisions, but we reserve the arrival of the TU at the TC system just beforehand.

5 Analytical evaluation of the communication load of module clusters

The communication load for a reservation and rescheduling process in a TC system with or without a cluster can be analytically calculated, which is shown in the following sections. Disturbances such as an incorrect broadcast of a message are not considered.

The number of broadcasted messages can be analytically calculated in correlation to the number of entrances and TCs. The formulas shown below in Fig. 2 and 3 apply to a successful routing and reservation process. We use both equations to derive another equation that we can use to calculate the difference in the communication load if a cluster coordinator is present, as shown in (1).

$$\Delta N = N_{TC} * (N_{Entrances} - 1) - 1 \qquad (1)$$

By using only one entrance, the coordinator has no effect, and the number of messages stays the same since it acts exactly like a single entrance. With an increasing number of entrances, the coordinator reduces the communication load, because it represents communication for more and more modules. Consequently, the cluster coordinator fulfils the requirement not to cause any more communication, even in the case there is no need.

The number of broadcasted messages can be calculated with the formula shown in Fig. 6. In the first line of the formula, the number of messages that are sent by the TC are calculated, which itself originally sent a change request. In the second line, the number of messages broadcasted by other TCs will be calculated. Those TCs have to reschedule their transports, because of a change request from another TC.

Figure 6. Formula for the calculation of the communication load with respect to rescheduling in the TC System.

We simplify the complex formula with the assumption, that each TC only sends a change request for a single transport. Furthermore, we evaluate the module cluster with two representative scenarios, for which we can simplify the formula further. In both scenarios, a TC sends a rescheduling request to all other TCs.

- In scenario 1, the required new schedule of each TC does not interfere with the transports of other TCs.
- In scenario 2 the required new schedule of each TC interferes with the transports of other TCs. Therefore, each TC additionally sends a rescheduling request to all other TCs.

Finally, all of the TCs send a change request to the corresponding entrances of their affected transports. Starting from the entrances, a message with the time slot buffer is passed back to the module that originally requested rescheduling. After deciding on a rescheduling option, the confirmation is finally distributed to the affected modules in the end. For scenario 1, we can simplify the formula and replace the formula shown in Fig. 6 with 2, and for scenario 2, we can use 3 accordingly.

In a parameter study, we can vary the parameter number of entrances and TCs to observe the effects on

the number of broadcasted messages for a change request. The results are shown in Fig. 7. If we double the number of entrances and TCs, the number of messages nearly quadruples for scenario 1 and increases eightfold for scenario 2. In the case of a TC system with a cluster coordinator, the number of messages stays the same. In conclusion, the cluster communicator fulfills the requirement to cut down on the communication for modules that depend on each other.

$$M_{Messages} = 3 * N_{Entrances} + 3 * (N_{TC} - 1) * (1 + N_{Entrances}) \quad (2)$$

$$M_{Messages} = 3 * N_{Entrances} * (1 - N_{TC}) - 3 * N_{TC} + 3 * N^2_{TC} * (1 + N_{Entrances}) \quad (3)$$
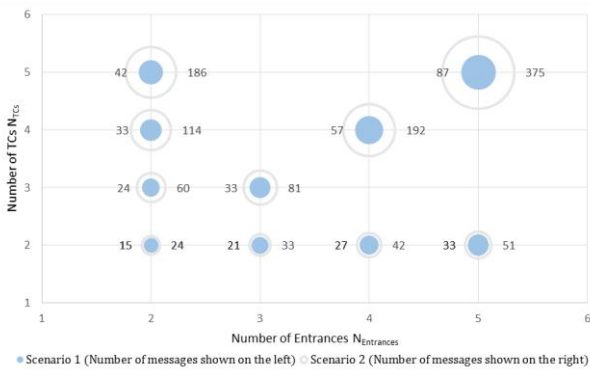

Figure 7. Results of the parameter study to observe the correlation between the number of entrances and TCs on the communication load.

6 Conclusion

In this paper we introduced convertible heterogeneous material flow systems that are realized by making use of autonomous software and hardware modules. We discussed different options to ensure deadlock free transportation in a decentralized controlled system. A routing concept with a time slot based reservation algorithm, as used in previous works for similar systems, appeared to be the best option to prevent deadlocks. In heterogeneous material flow systems, modules might depend on each other, because they use the same resources. Delays, early arrivals and new transport assignments lead to a demand for rescheduling confirmed reservations. We discovered that, in this case, a time-slot-based reservation may cause a very high communication load, especially for modules that depend on each other.

For reducing the communication load we introduced the module cluster with a centralized coordinator. The coordinator takes over the communication from the modules in the cluster. We compared the conventional communication process for each regular reservation and rescheduling process to the process with a cluster coordinator. We used a transversal carriage system as an example system, which is generally representative for modules that depend on each other. We analytically evaluated the number of broadcasted messages in a transversal carriage system. We then stated formulae to calculate the number of messages for each a regular reservation process and rescheduling, each with or

without a cluster. In a parameter study, we showed the effect of module clusters depending on the complexity of the cluster. We assessed that module clusters have the greatest effect on cutting message traffic in the case of rescheduling firmed reservations.

Module clusters represent a hybrid solution to an either completely decentralized or centralized material flow control. The concept of a decentralized control for convertible material flow systems has been combined with selective advantages of a central control and provides a greater knowledgebase and reduced demand on coordination. Further studies will analyse automatic formations and additional application areas of module clusters.

References

1. K. Furmans, F. Schönung, K. R. Gue, International Material Handling Research Colloquium, *Plug-and-Work Material Handling Systems* (2010)
2. B. Vogel-Heuser, J. Lee, P. Leitão, at - Automatisierungstechnik **63**, *Agents enabling cyber-physical production systems*, 777-789 (2015)
3. O. Shehory, S. Kraus, Artificial Intelligence **101**, *Methods for task allocation via agent coalition formation*, 165–200 (1998)
4. M. Hellenschmidt, R. Wichert, *Self-organization: Things with autonomous cooperation – a system analysis - Selbstorganisation: Dinge in eigenverantwortlicher Kooperation - eine Systemanalyse* (Internet der Dinge, Berlin, 2007)
5. A. W. t. Mors, *The world according to MARP - Multi-Agent Route Planning* (Technische Universiteit Delft, 2010)
6. S. H. Mayer, *Development of a completely decentralized control system for modular continuous conveyor systems* (Karlsruher Institut für Technologie, 2009)
7. R. Chisu, F. Kuzmany, W. A. Günthner, Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik **5**, *Decentral coordination and cooperation in the internet of things - Dezentrale Koordination und Kooperation im Internet der Dinge*, 35-37 (2009)
8. W. A. Günthner, *Internet of things in the intralogistics - Internet der Dinge in der Intralogistik.* (Heidelberg, 2010)
9. E. G. Coffman, M. Elphick, A. Shoshani, ACM Computing Surveys **3**, *System Deadlocks*, 67-78 (1971)

Acknowledgment