Implementation, Self-Configuration and Coordination of Logistical Functions for Autonomous Logistics Modules in Flexible Automated Material Flow Systems

Christian Lieberoth-Leden, Juliane Fischer, Birgit Vogel-Heuser, Johannes Fottner
Technical University of Munich, Munich, Germany
Email: lieberoth-leden@fml.mw.tum.de, juliane.fischer@tum.de, vogel-heuser@tum.de, kontakt@fml.mw.tum.de

***Abstract*—The modularization of hard- and software is one approach to handle the demand for increasing flexibility and changeability of automated material flow systems that are, for example, utilized in flexible production systems. Depending on the current system configuration, the position and number of entrances and exits of a module may vary. Subsequently, the feasible tasks and the internal execution order of operations within a module are affected. During design time, the later system configuration is unknown, therefore, a concept is proposed to generally describe a module's internal logistical operations. After a system reconfiguration, the module's internal function control automatically determines the execution order for active entrances, exits and tasks. Additionally, the function control is able to efficiently coordinate the execution of parallel transports on the same module to increase the throughput.**

***Index Terms*—convertible and flexible automated material flow systems, cyber-physical systems, decentralized control, execution coordination, execution schedule**

I. Introduction

Present day automated material flow systems (aMFS) are mostly operated by an individual central control. Developing the specialized control software demands a manual effort, and flexibility is only facilitated within predefined limits. This is the case, for example, when a designated alternative route is temporarily used for a higher throughput. New demands on an existing aMFS that require flexibility not originally considered (e.g., an extension of the system) cannot be realized. In this case, an aMFS is not considered or supplemented by manual processes [1], [2]. An aMFS that provides flexibility beyond predefined limits is characterized as being convertible. Convertible aMFS are a response to reconfigurable manufacturing systems that increase the manufacturing responsiveness for a faster adaption to changing market conditions [3].

A convertible aMFS can be realized by means of function-oriented modularization of the hardware and software [4]. By adding, removing or changing modules, the layout of an aMFS is adapted to the new demands. New demands arise from changing manufacturing or logistic processes, which are caused by new products requiring different operations, a fluctuating production volume, a modification of the layout in the production due to new machinery or other reasons. The control of convertible aMFS can be realized through local autonomous modules that cooperate with each other. Advantages are a reduced software complexity and eased re-configurability [5].

In this paper, a module is defined as an encapsulated unit that performs predefined tasks, such as conveying, buffering or identifying a transport unit (TU). A module possesses all the necessary information and software to control its hardware, to communicate with other modules or superior systems and to perform an autonomous self-configuration. Standardized software interfaces and property descriptions enable collaboration between heterogeneous modules [6], for example, the combination of a conveyor with a crane. Consequently, autonomous self-controlled modules allow for convertible aMFS that can be changed during runtime [7].

This paper distinguishes between an external material flow control and an internal function control. The material flow control plans in which order the TUs need to pass through the modules and the tasks to be performed on the TUs by the modules in cooperation with the other modules. The function control independently executes the operations within a module, with the exception of a handshake with neighboring modules for a common transfer of a TU. For the execution of a transport, the function control has to derive the necessary operations and the execution sequence of these operations from the assigned TU order and tasks on the module.

The execution sequence depends on the number of active entrances, exits, and tasks of a module as well as the position of the neighboring modules. Subsequently, predefined operation sequences cannot be created during design time but must be generated and parametrized dependent on the current system configuration. For design time, an approach is required to implement generalized operations that enable the performance of a wide variety of different operation sequences for different system

Manuscript received October 16, 2018; revised May 2, 2019.

doi: 10.18178/ijmerr.8.4.498-505

configurations. The approach should focus on general logistical operations while at the same time allowing for encapsulated complex module-specific operations (e.g. with requirements on precise timing) that can also be combined automatically with the general logistical operations. During run time, the function control should automatically derive a valid execution sequence for a given entrance, exit and task. Furthermore, the function control must be able to coordinate the execution sequences of different TUs on the same module in order to avoid blockades and optimize the throughput of the module.

This paper focuses on the internal control of a module and not on the strategic planning processes for the material flow control. After an overview of the state of the art in controlling convertible aMFS and production systems consisting of separate modules, the general functionality of a material flow control and the benefits of an additional individual function control for each module is described. Following, the engineering process for a flexible function control is brought into focus. Also, the proposed concept is evaluated on a representative logistical module. Subsequently, the quantitative evaluation for the concept is shown with a simulation study. The last section concludes the proposed concept and gives an outlook on the planned future work.

II. 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 agents in order to accomplish a task. *Vogel-Heuser et al.* stated that the utilization of such agents allows for Cyber-Physical Production Systems for Industry 4.0 applications [8]. *Mayer* developed the hardware for autonomous material flow conveyor modules and the software for a decentralized routing and reservation process to avoid deadlocks [9]. Modules are reserved exclusively for one TU as long as the transport is not completed. In order to achieve a better utilization of a resource's capacity, *Mors* introduces time slots [10]. *Lieberoth-Leden et al.* propose a communication ontology for a time-slot-based reservation process with absolute time in aMFS [11] and *Seibold* proposes a reservation and routing process using relative (logical) time, which is more robust concerning delays in the transport process [12].

Several approaches introduced standardized modules to build reconfigurable aMFS controlled by multiagent systems [13], decentralized controls [9] or a centralized control [14], [15]. All approaches have in common that TUs enter and exit modules through predefined and during design time known interfaces. A specific internal module control is developed for each module. Modules with multiple components and a varying number of entrances, exits or positions are not considered. *Libert* presents a procedure to develop an individual material flow control for aMFS realized with software agents. As an agent platform, his uses the PC-based JADE-Framework and not an implementation on industrial real time control hardware [16].

Beyer et al. present a multiagent system approach for designing and choosing a suitable layout for a MFS. The development engineer enters a set of requirements for the multiagent system to be designed in an assistance system, which generates alternative design solutions and rates them according to the specified requirements [17]. This approach can be applied when developing new MFS as well as to reconfigure or optimize existing MFSs.

For decentralized controls, the control function of the system is split into several less complex control functions and distributed on various modules that communicate with each other. In most systems this leads to increased communication traffic compared to a central control [18].

Another approach to increase the flexibility of aMFSs is to substitute the commonly used stationary conveyor modules with a fleet of identical, automated guided vehicles (AGVs). For path planning and coordination of such AGVs in a warehouse, *Digani et al.* propose a two layer, hierarchical control architecture to reduce the complexity and simplify the control problem [19]. For control purposes, the global area of the warehouse is divided into smaller sectors. While the global path planning for the AGVs through the sectors is done in a centralized manner on top layer, the route planning and coordination inside a sector are performed in a decentralized manner by the AGVs on lower level. However stationary conveyor modules achieve a greater TU throughput and cannot always be substituted with AGVs.

Armentia et al. present a concept to generate software for service-oriented systems and focus on the composition and dynamic reconfiguration of service-orientated applications [20]. In service-oriented applications, all software functionalities are defined as independent services that interact with each other and allow for a flexible adaptation of these applications to changing requirements. However, the approach from *Armentia's et al.* is not focused on logistic functionalities where the services (functionalities) for different TUs have to be executed in a predefined sequence. *Dorofeev et al.* propose a device adapter to wrap the functions of a device and offer them as a service to the system and execute the internal device operations [21]. Only validated executions are allowed that are predefined manually.

Aicher et al. propose a meta model to describe encapsulated modules in aMFS with standardized interfaces for code generation [6] and version compatibility [22]. The engineering process for logistical operations and the generation of execution sequences is not considered.

Keddis et al. describe a capability-based approach and defines different functions for modules in adaptable manufacturing systems. The functions consist of one or more primitive operations. *Keddis et al.* describe that several primitive operations can be combined to allow for more complex operations such as combining different handling operations to an assembling operation. However, the process of combining primitive operations is not further explained. For each product, a production plan is generated manually. The production plan lists the requested functions and determines the order of execution. The production plan is then mapped to suitable modules that can execute the requested function. Depending on the layout and the number of modules that can perform a function in the manufacturing system, there are several different paths. Valid paths are determined with the branch-and-bound and backtracking algorithm [23]. Later works focus on selecting a path that optimizes i.e. energy consumption, utilization, delivery time, etc. in correlation with other products [24]. The implementation process of module functions and the execution of assigned functions are not regarded. Furthermore, the approaches lack a concept for the coordination of the parallel execution of functions from different TUs.

III. Software Interfaces of Modularized aMFS

The material flow control determines the predecessor module, the successor module, the tasks to be performed for the TU and the TU sequence at the exit for each TU and module. The function control provides the module's properties to the material flow control and executes the material flow.

A. *Material Flow Control*

The material flow control receives transport orders from superior systems i.e. an enterprise resource planning system (ERP) or a warehouse management system (WMS), or it generates its own orders during the transport process "Fig. 1". Transport orders comprise information about the properties of the TU, source, destination, tasks (i.e. wrapping with foil) and material flow properties (i.e. required TU sequence at the destination). The material flow control conducts resource planning and schedules the transport orders.

Independent from a centralized or decentralized implemented material flow control and the structure of the database, such as centralized or distributed, the outcome of the scheduling process is identical. The material flow control arranges a module sequence (transport from source to destination), a TU sequence at each module (satisfy the material flow properties) and assigns tasks to selected modules (fulfill the required tasks). In order to schedule transport orders, the material flow control requires knowledge about the linkage of modules and the feasible tasks of a module.

B. *Module Structure and Function Control*

Decentralized controlled logistical modules are made up of one or more components and the hardware control unit. A component consists of at least one sensor or actuator like a belt conveyor with a drive and a light barrier. A logistical component can perform an atomic operation (cf. primitive operations in manufacturing systems [23]) i.e. conveying. The general structure, operations, properties, etc. of a module can be described in a meta model [6]. Atomic operations describe general abilities but lack logistical relevance for the material flow control. In order to conduct a material flow through a module, several atomic operations must often be linked together.

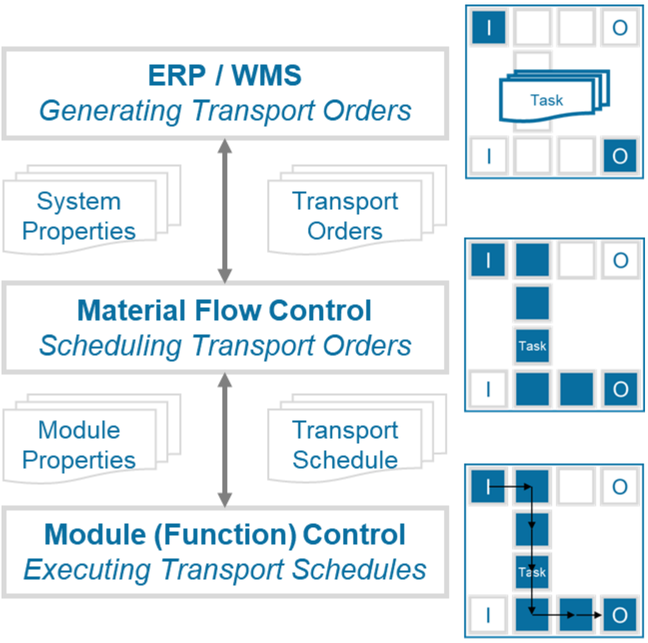

Figure 1. Process of generating, scheduling and executing transport orders in an aMFS.

Once a module is repositioned or added to an aMFS, it performs an autonomous self-configuration. After the identification of connected neighboring modules, feasible logistical offers (LO) are derived from the operations such as conveying a TU from neighbor A to neighbor B. LOs implicate through which neighbor a module can be entered and exited and the feasible tasks performable between entry and exit point. The accessible neighboring modules and the feasible tasks depend on the current system configuration and the entry point of a TU. For example, the balance of a unidirectional conveyor can only be utilized if the entrance is positioned ahead of the balance because reverse conveying is not allowed. Subsequently after reconfiguring an aMFS, the LOs of the affected modules have to be repeatedly determined. The material flow control selects a LO for a TU. When the TU arrives at the module, the predefined LO is executed by the locally assigned function control.

C. *Application of the Flexible Logistical Function Control*

In an aMFS containing solely modules with one operation, only a simple function control is needed that matches the connected neighboring modules to the operation in order to generate the module offer. The material flow control links the operations, or modules, respectively, together in the context of the scheduling process.

Incorporating several components in one module and describing the capabilities with atomic operations leads to the following advantages:

- The number of decentralized control units is reduced, which leads to less traffic in the communication system.
- Modules can be connected independently from predefined hardware interfaces and allow for more flexible system configurations and layouts.
- Implementation of an individual execution strategy for transport tasks within a module independent from the general material flow strategy of the system.
- More sophisticated operations can be realized efficiently utilizing several components in one module and combining operations.

More sophisticated operations have, for example, constraints on timing and the interleaved collaboration of several components. For example, the discharging of a TU from a belt conveyor on the fly through a pusher requires precise timing and a defined setup such as a light barrier at an exact and known position, as shown in “Fig. 2”. Identifying the required interfaces, verifying the capabilities and executing the function with several separate modules is more complex and error-prone. Encapsulating a sophisticated operation and connecting it with defined interfaces to basic operations within one module enables neighboring modules to utilize the basic operations (e. g. conveying with constant speed) to configure a common material flow interface. Alternatively, a detailed description of valid system configurations for the operator is needed. Subsequently, modules with several components encapsulating sophisticated operations allow for more intuitive system configurations for the operator but also enable flexible system configurations with no predefined hardware interfaces by providing basic operations to establish a material flow interface between two modules.

Defining only the operation *discharging* for the module implies that the hardware interfaces of the connected modules are set since the operation has a fixed starting and ending position (A to B and A to C). Describing the capabilities of the module with several atomic operations enables one to connect neighboring modules at points not originally designated (A_{New}). The function control verifies if TUs from A_{New} qualify for the operation *discharging* (TUs need to enter before the second light barrier from the left).

For complex modules with many components and different transport options, one’s own material flow strategy might be more efficient than the general strategy of the material flow control. A module’s function control facilitates the utilization of internal material flow strategies for a more efficient material flow in the aMFS.

IV. Automatic Generation of Logistical Function Schedules

Modules are developed during design time with no specific knowledge about their later application in an aMFS. Subsequently, the number, the kind and the position of neighboring modules are unknown. But the information is needed to coordinate the internal material flow within a module. In order to cover as many variants of different applications as possible, the operations must be implemented generically. Furthermore, the execution of the operations must be adaptable to the current application. The coordination of the material flow within a module must satisfy a given TU sequence at the exits and avoid deadlocks independent from the current position or number of neighbors.

The following subsections distinguish between the engineering process of self-configuring modules during design time, the automated configuration process after layout changes and the execution of operations during run time in order to accomplish a material flow. In this paper, a pusher module shown in “Fig. 2” is used to explain the concept and conduct a conceptual evaluation.

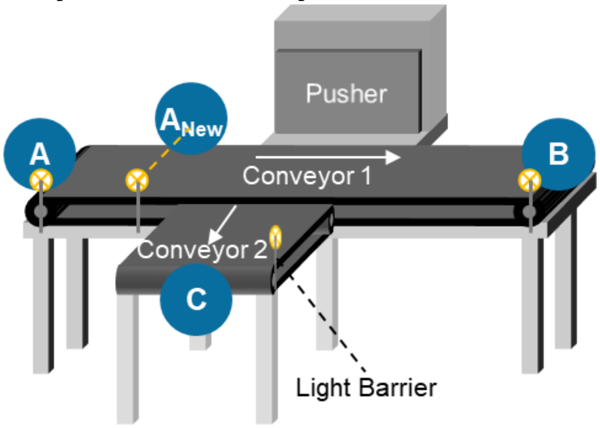

Figure 2. Example of a logistical module consisting of two conveyors and a pusher for discharging TU from one conveyor to another.

A. Engineering Functions

Each component of a module is able to execute a set of atomic operations with specific properties. In this paper, an area in which an operation is performed is called operating space (OS). Each operation is assigned to only one OS with a specified capacity for TUs. OSs with a capacity greater than one are able to simultaneously execute their operation on multiple TUs (e. g. to enable bulk conveying). “Fig. 3” shows the layout and coordinate system of the logistical module from “Fig. 2”. Three atomic operations and, subsequently, three OSs can be identified for the module: OS 1 for the operation *conveying* on conveyor 1 with the capacity three, OS 2 for the operation *conveying* on conveyor 2 with the capacity one and OS 3 for the operation *pushing* simultaneously allocating conveyor 1, conveyor 2 and the pusher with the capacity one.

During design time, logistical functions (LF) are derived from one or more atomic operations. LFs provide the specific capabilities needed to realize a material flow in aMFS. The execution of a dynamic pusher function demands a conveying operation and a timed push operation in order to discharge a TU from conveyor 1. The operations are condensed and implemented in the LF *inTD* (internal transfer dynamic discharging). The LFs are constructed manually for each module during design time and implemented in a function list that is described in the following and shown in “Fig. 4”. A unique name identifies an LF within a module. Operations required to perform a function are listed with their corresponding OS. The entrances and exits to an LF are described by value ranges and refer to the overall dimensions of the TU depending on the direction of movement. The LF *C1* (Conveying 1) can be entered at each position between the coordinates x ∈{0-18} but only centered in the middle of the conveyor y∈{7} and on the conveyor z∈{0}. Whereas the LF *inTD* can only be entered at x∈{6} because it has to pass the light barrier to perform a pusher operation. Thus, the LF *inTD* can be performed on a TU entering the module anywhere before or at x={6}.

Besides transporting a TU from an entrance to an exit, an LF can perform a handling task such as weighing or identifying a TU and is stated in the column *Task*. The transport and the task can be executed with different parameters, which are also stated in the function list.

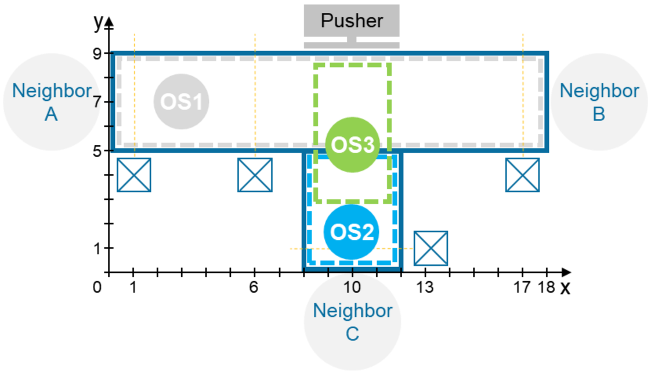

Figure 3. Diagram showing the OSs and coordinates of the pusher module.

Name	Description	OS		Entrance	Exit	Parameter	Task
C1	Conveying 1	1	x y z	{0-18} {7} {0}	{x-18} {7} {0}	Speed {0.1;0.2}	
C2	Conveying 2	2	x y z	{10} {5} {0}	{10} {0} {0}	Speed {0.1;0.2}	
inTD	Internal Transfer Dynamic Discharging	1,2,3	x y z	{6} {7} {0}	{10} {1} {0}		
exTA	External Transfer A	1	x y z	{0} {7} {0}	{1} {7} {0}	Speed {0.1}	
exTB	External Transfer B	1	x y z	{17} {7} {0}	{18} {7} {0}	Speed {0.1}	
exTC	External Transfer C	2	x y z	{10} {1} {0}	{10} {0} {0}	Speed {0.1}	

Figure 4. Function list of the pusher module.

LF can be performed in only one OS, in several OSs of one module for an internal transfer or in several OS of two modules for an external transfer. For reconfigurable aMFS, the external transfers are generated automatically and added to the function list during the configuration phase (cf. "Fig. 4" last three entries in the function list). The external transfer functions are generated for defined logistical transfers between modules in an aMFS and regarded in previous works [7]. LFs performed in several OSs link OSs together. The linkage is represented in the OS routing matrix shown in "Fig. 5".

Each OS and each connected neighboring module is stated in the first row and column of the matrix. The OS routing matrix is the basis for the automated configuration of different workflows through the module. The LFs are added manually to the matrix during design time. For the pusher module, the OS routing matrix states that from OS1 a transport to OS3 executed by the LF *C1* and from OS3 a transport to OS2 executed by the LF *inTD* is available. The coordinates for the entrance and exit are given in the function list.

B. Configuration of Logistical Offers

The external material flow control selects a module for the transport sequence if the destination is accessible through the module and the requested tasks can be fulfilled. Further optimization like traffic balancing is considered after the key requirements are satisfied. Subsequently for each entrance, the reachable exits and feasible tasks are of interest for the external material flow control.

		From					
	OS	1	2	3	A	B	C
To	1	C1			exTA		
	2		C2	inTD			
	3	C1					
	A						
	B	exTB					
	C		exTC				

Generated automatically during configuration phase

Figure 5. OS routing matrix of the pusher module.

At configuration phase, neighboring modules are identified through a comparison of the module's position coordinates within the system and the dimension of the module. After the position and type of a module interface (entrance or exit) are set, the external transfer functions are generated. The OSs are determined from the exit coordinates of the external transfer functions. The function control adds the generated external transfer functions to the LF list and OS routing matrix. Following the completion of the LF list and OS routing matrix for the current configuration, function control calculates executable logistical offers shown in "Fig. 6". LOs consist of an entrance, exit, optional task, LF sequence and properties (i.e. process time). The LF sequence states the type of LF, order and properties (entrance and exit coordinates, speed, etc.) for a specific LO.

First, the function control considers every theoretically possible combination of entrances, exits and tasks for a LO. Next, for each theoretical LO, an LF sequence is generated. A depth-first search is applied to the OS routing matrix to calculate every valid path from the entering to the exiting neighboring module. LO with no valid paths are not further considered in the calculation. The remaining valid paths are parametrized and validated. The entrance value range of an LF must match the exit value range of the predecessor function and the parameters (i.e. conveying speed) must be suitable. For

LOs with more than one valid path, the path with the most convenient properties is selected by the function control. Conclusively, the LOs with valid LF sequences are added to the LO list shown in "Fig. 6". The function control is able to determine LOs and the corresponding LF sequence independent from predefined hardware interfaces for neighboring modules. For each entrance, the accessible exits and feasible tasks are assigned individually for the current configuration of the aMFS. Usually a logistical module connects two to four neighbors, performs one task and possesses a limited number of components. During the configuration phase, calculating time is not a critical shortage. Therefore, a heuristic procedure is applied to calculate valid LOs.

C. Execution

During scheduling, the material flow control selects a LO for a TU and informs the function control. When a TU arrives at a module, the function control prepares the execution of the assigned LO with the LF sequence.

Entrance	Exit	Task	OS	LF Sequence
A	B		A,1,B	exTA({0,7,0};{1,7,0};0.1), C1({1,7,0};{17,7,0};0.1), exTB({17,7,0};{18,7,0};0.1)
A	C		A,1,3,2,C	exTA({0,7,0};{1,7,0};0.1), C1({1,7,0};{6,7,0};0.1), inTD({6,7,0};{10,1,0}), C2({10,1,0};{10,1,0};0.1), exTC({10,1,0};{10,0,0};0.1)

Figure 6. List of LOs of the pusher module.

The parallel execution of multiple LOs for different TUs requires the coordination of common OS to avoid collisions or deadlocks. For this purpose, an LF schedule is established with a column for each OS and rows representing every single execution step. The LF of all arriving TUs is entered in the LF schedule according to the LF sequence. The order of the LF sequence of different TUs is scheduled according to the planned order of the TUs arriving at the module. LFs allocated to several OSs are assigned to several OSs' columns but in the same row, since the OSs are required for the same execution step (parallel execution).

The execution schedule shown in "Fig. 7" gives an overview about how the LF of a TU depends on the execution of the LF of another TU or which LFs can be performed independently in parallel execution. Adjoining LFs for different TUs in the same OS (same OS equals same underlying operation) can be checked for simultaneous execution. For example, on a long conveyor, several TUs can be transported at the same time rather than waiting till the transport of the successor TU exits the conveyor.

For further internal optimization, a module-specific OS management system can be implemented. For example, an H-crossing (see "Fig. 8") can be optimized by pooling TU, travelling the same direction in the mid section, for one consecutive transport. The constraints are that TUs arrive at the assigned exit in sequence. For a grid, the OS management system arranges the LF sequences in such a manner that interferences are reduced to a minimum to increase the throughput of the module.

V. Evaluation

The evaluation of the engineering process is shown in the example of the pusher module in the previous section. In this section, the advantages of the concept in terms of TU throughput of a module are evaluated. The concept was implemented in a simulation model to prove that the concept can be implemented in code and to show the advantages in terms of throughput.

Step	OS 1	OS 2	OS 3	OS 4
1	exTA			
2	C1			
3	inTD	inTD	inTD	
4		C2		
5		exTC		
6	exTA			
7	C1			
8	exTB			

Possible parallel execution   Possible simultaneous execution

Figure 7. Exemplary execution schedule for the pusher module for two different TUs (marked with different colors).

For the simulation model, the H-crossing ("Fig. 8" left) was chosen because the module consists of several components, allows for the parallel processing of different TUs, and a strategy can be applied to optimize the module. Three scenarios are distinguished for the evaluation of the throughput:

1) Scenario 1: Logistical offers are implemented manually for the module and can only be executed serially to avoid conflicts such as deadlocks.

2) Scenario 2: Logistical functions are manually implemented and different OSs are defined, such as those proposed in this paper. The function control coordinates parallel execution depending on the situation and avoids deadlocks.

3) Scenario 3: In addition to scenario 2, the function control applies a module strategy. The mid section prioritizes successor TUs traveling in the same direction.

The results of the simulation study are shown in "Fig. 9". Between scenario 1 and 2, a major difference of the throughput can be observed. Also, the maximum throughput can be increased because of the optimized utilization of the module through parallel execution. Scenario 3 differs from scenario 2 only in the maximum throughput. The sequence of the TU is mainly determined by the material flow control; therefore, the optimization with a local module strategy has only little influence on the throughput.

VI. Conclusion and Outlook

Flexible aMFSs consist of different module types with specific logistical functionalities. Modules with predefined entrances, exits and logistical functions contradict the concept of flexible aMFS. Depending on the number and position of the entrances and exits, a module can offer different logistical functions and tasks to the system. In this paper, a concept is proposed that enables an engineer to parametrize general logistical functions for modules during design time. Value ranges describe valid entrance and exit points for the flexible combination of different logistical functions. Additionally, the concept enables the developer to implement sophisticated functions within a module that consists of different components.

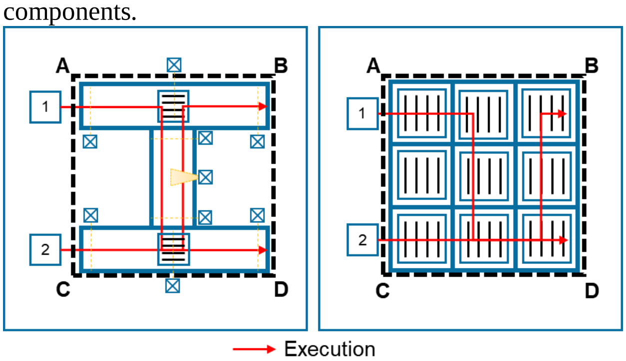

Figure 8. Different logistical modules that request different OS management systems for an efficient exceution.

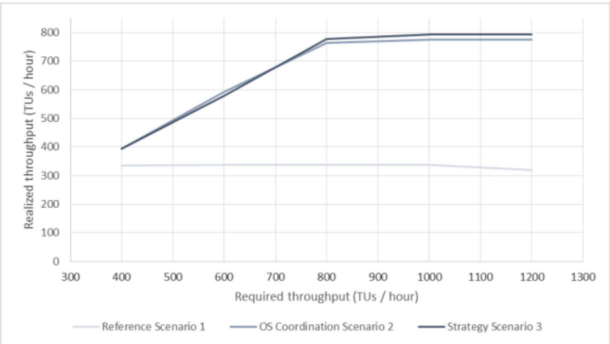

Figure 9. Comparison of the throughput for different scenarios for the H-crossing's function control.

Depending on the actual system configuration, a module's function control automatically identifies logistical functions to offer to the material flow control during the configuration phase. For each entrance, the function control automatically derives execution sequences to all reachable exits and tasks of a module. Furthermore, a module is able to coordinate the execution of functions and optimize the throughput through the parallel or simultaneous execution of functions and the avoidance of blockades. The concept was implemented in a simulation model to proof the concept and examine the advantages in terms of throughput.

The concept proposed in this paper is suitable for modules with several components and several options for placing entrances and exits that perform a material flow. For modules with only one component or fixed entrances and exits, predefined control code is sufficient. Also, for modules with a large number of components, the concept may lead to long configuration times because heuristic algorithms are applied.

The local module strategy can be in contradiction to the material flow strategy. Therefore, only simple module strategies and modules with few components should be employed in the self-configuring aMFS. In contrast to the parallel execution of LFs, the local module strategies have only little influence on the throughput. Therefore, the local module strategies should be automatically adopted by the material flow control. In the future, suitable interactions between the function control and the material flow control to jointly optimize the system and module throughput will be investigated.

Acknowledgment

We thank the German Research Foundation (DFG) for funding this project (GU 427/25-1, VO 937/24-1).

References

[1] X. L. Hoang, A. Fay, P. Marks, and M. Weyrich, "Systematization approach for the adaptation of manufacturing machines," in *Proc. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–4.

[2] K. Furmans, F. Schönung, and K. Gue, "Plug-and-work material handling systems," in *Proc. International Material Handling Research Colloquium (IMHRC)*, Milwaukee, Wisconsin, 2010.

[3] Y. Koren et al., "Reconfigurable manufacturing systems," CIRP Annals - Manufacturing Technology, vol. 48, no. 2, pp. 527–540, 1999.

[4] M. Wilke, "Control concept for autonomous changeable material flow systems," *Logistics Journal*, 2008.

[5] D. M. Dilts, N. P. Boyd, and H. H. Whorms, "The evolution of control architectures for automated manufacturing systems," *Journal of Manufacturing Systems*, vol. 10, no. 1, pp. 79–93, 1991.

[6] T. Aicher et al., "Increasing flexibility of modular automated material flow systems: A meta model architecture," in *Proc. 8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM), Troyes*, France, 2016, pp. 1543–1548.

[7] D. Regulin, D. Schutz, T. Aicher, and B. Vogel-Heuser, "Model based design of knowledge bases in multi agent systems for enabling automatic reconfiguration capabilities of material flow modules," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 133–140.

[8] B. Vogel-Heuser, J. Lee, and P. Leitão, "Agents enabling cyber-physical production systems," at - Automatisierungstechnik, vol. 63, no. 10, 2015.

[9] S. H. Mayer, "Development of a completely decentralized control system for modular continuous conveyor systems," Dissertation, Institut für Fördertechnik und Logistiksysteme, Karlsruher Institut für Technologie, Karlsruhe, 2009.

[10] A. W. t. Mors, "The world according to MARP: Multi-Agent Route Planning," Dissertation, Technische Universiteit Delft, Delft, 2010.

[11] C. Lieberoth-Leden, D. Regulin, W. A. Günthner, M. Figueira, and Z. Guo, "Efficient Messaging through Cluster Coordinators in Decentralized Controlled Material Flow Systems," *MATEC Web Conf.*, vol. 81, p. 6005, 2016.

[12] Z. Seibold, "Logical time for decentralized control of material handling systems," Dissertation, Institut für Fördertechnik und Logistiksysteme, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2016.

[13] G. Black and V. Vyatkin, "Intelligent component-based automation of baggage handling systems With IEC 61499," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 337–351, 2010.

[14] S. Haneyah, Generic Control of Material Handling Systems. Enschede: University of Twente, 2013.

[15] A. Brusaferri, A. Ballarino, F. A. Cavadini, D. Manzocchi, and M. Mazzolini, "CPS-based hierarchical and self-similar automation architecture for the control and verification of reconfigurable manufacturing systems," in *Proc. the 2014 IEEE Emerging*

Technology and Factory Automation (ETFA), Barcelona, Spain, 2014, pp. 1–8.

[16] S. Libert, "Contribution for designing agent based material flow controls," "Beitrag zur agentenbasierten Gestaltung von Materialflusssteuerungen," Dissertation, Lehrstuhl für Förder- und Lagerwesen, Technischen Universität Dortmund, Dortmund, 2011.

[17] T. Beyer, P. Gohner, R. Yousefifar, and K.-H. Wehking, "Agent-based dimensioning to support the planning of Intra-Logistics systems," in *Proc. 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–4.

[18] R. Jünemann and A. Beyer, "Control of material flow and logistics systems," Steuerung von Materialfluß- und Logistiksystemen: Informations- und Steuerungssysteme," Automatisierungstechnik. Berlin, Heidelberg: Springer, 1998.

[19] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Towards decentralized coordination of multi robot systems in industrial environments: A hierarchical traffic control strategy," in *2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2013, pp. 209–215.

[20] A. Armentia et al., "Achieving reconfigurable service oriented applications using model driven engineering," in ETFA 2011, Toulouse, France, 2011, pp. 1–4.

[21] K. Dorofeev et al., "Device adapter concept towards enabling plug&produce production environments," in *Proc. the IEEE International Conference on Emerging Technologies And Factory Automation (ETFA)*, Limassol, Cyprus, 2017.

[22] T. Aicher, D. Schütz, M. Spindler, and B. V. H. W. Günthner, "Automatic analysis and adaption of the interface of automated material flow systems to improve backwards compatibility," in *20th IFAC World Congress (IFAC 2017)*, 2017.

[23] N. Keddis, G. Kainz, and A. Zoitl, "Capability-based planning and scheduling for adaptable manufacturing systems," in *Proc. IEEE International Conference on Emerging Techologies and Factory Automation (ETFA)*, Barcelona, Spain, 2014, pp. 1–8.

[24] N. Keddis, B. Javed, G. Igna, and A. Zoitl, "Optimizing schedules for adaptable manufacturing systems," in *Proc. IEEE International Conference on Emerging Techologies and Factory Automation (ETFA)*, Luxembourg, 2015, pp. 1–8.

Christian Lieberoth-Leden graduated in mechanical engineering from the Technical University of Munich in 2014. Since 2015 he is a research assistant at the chair of Materials Handling, Material Flow, Logistics with Professor Fottner at the Technical University of Munich. His main research interest is the design of modular control software for flexible automated material flow systems in automated production and logistics systems.